



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO INGENIERÍA MECÁNICA



Implementación de un método de medición y estimación del ángulo de ataque en una aeronave no tripulado de ala fija.

POR

Geovanni Ignacio León Rojas

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Aeroespacial

Profesor Guía:
Dr. Bernardo Andrés Hernández Vicente

Febrero 2024
Concepción (Chile)

© 2024 Geovanni Ignacio León Rojas

© 2024 Geovanni Ignacio León Rojas

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

Resumen

El ángulo de ataque (AOA) es el ángulo formado entre la dirección del viento relativo y la cuerda media aerodinámica. Este es uno de los parámetros con mayor importancia en el vuelo de una aeronave, debido a que se relaciona con las fuerzas aerodinámicas, la entrada en pérdida y la eficiencia aerodinámica. En el presente trabajo se diseña, implementa y valida un método de medición y estimación del AOA en un aeronave no tripulada de ala fija.

El método de medición diseñado se basa en correlacionar una diferencia de presión normalizada con el AOA medido por la veleta pivotante, esta diferencia de presión se obtiene mediante la medición de presión en el eje longitudinal de la aeronave y una medición a 45° respecto a este eje, para luego normalizar dicha diferencia por la presión P_{45} . Para lograr la medición de las presiones se diseña una sonda que cumpla los requerimientos de medición y se utilizan los sensores Ms4525do, los cuales son sensores de presión diferencial digitales que tiene una resolución de $0,84 [Pa]$.

Se diseña un método de estimación del AOA basado en la fusión de sensores y algoritmos. En primer lugar, se obtiene el ángulo de ataque y el ángulo de deslizamiento lateral mediante un método algebraico. Luego se implementa un filtro de Kalman extendido (EKF) para estimar las velocidades lineales en el cuerpo de la aeronave y con estas calcular el AOA. El EKF utiliza el resultado del AOA obtenido del método algebraico como pseudomedición para ayudar a la convergencia. Para el EKF el vector de estado se compone de las velocidades en el cuerpo, el vector de medición se compone de la velocidad aerodinámica, velocidades inerciales y el AOA obtenido algebraicamente y el vector de entradas compuesto por las aceleraciones, velocidades angulares y ángulos de actitud. Para lograr las mediciones se utiliza un controlador de vuelo Pixhawk 6C el cual cuenta con unidades de medida inercial.

Para implementar los métodos, se habilita la aeronave Kadet LT-40, junto con las veletas pivotantes. Se instaló el hardware necesario para una operación básica, y posteriormente se llevó a cabo un vuelo de prueba, durante el cual se determinó que la autonomía de las baterías es de 6 minutos. Se implementa el Hardware necesario para el método de medición y estimación, la sonda diseñada se instala bajo el ala derecha y los sensores de presión dentro del fuselaje. El piloto automático se instala dentro del fuselaje cercano al centro de gravedad de la aeronave.

Para la validación de los métodos se realizan una serie de vuelos para la obtención de datos. En método de medición se realiza la correlación entre el AOA real y la diferencias de presiones, obteniendo una curva de relación polinomial de grado tres, la cual se aplica para obtener el AOA dando como resultado un AOA con una desviación estándar de $0,84^\circ$ y un error cuadrático medio de $0,718$. El método de estimación no logro la convergencia, esto debido a que las ecuaciones que relacionan las velocidades inerciales con las velocidades en el cuerpo no fueron apropiadas para utilizarlas en el paso de actualización, por otro lado, en las ecuaciones de propagación falto considerar una entrada de control que se relacione directamente con el AOA, como lo es la deflexión del elevador.

Abstract

The angle of attack (AOA) is the angle formed between the relative wind direction and the aerodynamic mean chord. This is one of the most important parameters in the flight of an aircraft, because it is related to aerodynamic forces, stall and aerodynamic efficiency. In the present work, a method for measuring and estimating the AOA in an unmanned fixed-wing aircraft is designed, implemented and validated.

The measurement method designed is based on correlating a normalized pressure difference with the AOA measured by the pivoting vane, this pressure difference is obtained by measuring the pressure in the longitudinal axis of the aircraft and a measurement at 45° with respect to this axis, and then normalizing this difference by the pressure P_{45} . To achieve the pressure measurement, a probe is designed to meet the measurement requirements and the Ms4525do sensors are used, which are digital differential pressure sensors with a resolution of 0.84 [Pa].

An AOA estimation method based on sensor and algorithm fusion is designed. First, the angle of attack and sideslip angle are obtained using an algebraic method. Then an extended Kalman filter (EKF) is implemented to estimate the linear velocities on the aircraft body and with these calculate the AOA. The EKF uses the AOA result obtained from the algebraic method as a pseudo-measurement to aid convergence. For the EKF the state vector is composed of the velocities on the body, the measurement vector is composed of the aerodynamic velocity, inertial velocities and the AOA obtained algebraically and the input vector is composed of the accelerations, angular velocities and attitude angles. A Pixhawk 6C flight controller with inertial measurement units is used to achieve the measurements.

To implement the methods, the Kadet LT-40 aircraft is enabled, along with the pivoting wind vanes. The necessary hardware for a basic operation was installed, and later a test flight was carried out, during which it was determined that the autonomy of the batteries is 6 minutes. The necessary hardware for the measurement and estimation method is implemented, the designed probe is installed under the right wing and the pressure sensors inside the fuselage. The autopilot is installed inside the fuselage near the center of gravity of the aircraft.

For the validation of the methods, a series of flights are made to obtain data. In the measurement method, the correlation between the real AOA and the pressure differences is made, obtaining a polynomial relation curve of degree three, which is applied to obtain the AOA, resulting in an AOA with a standard deviation of 0.84° and a mean square error of 0.718. The estimation method did not achieve convergence, this because the equations that relate the inertial velocities with the velocities in the body were not appropriate for use in the update step, and the propagation equations did not consider a control input that is directly related to the AOA, such as the elevator deflection.

Tabla de Contenidos

Resumen	i
Abstract	ii
Tabla de Contenidos	iii
Lista de Figuras	v
Lista de Tablas	vii
Glosario	viii
1 CAPÍTULO 1: Introducción.....	1
1.1 Contexto.....	1
1.2 Objetivo general	3
1.3 Objetivos específicos.....	3
1.4 Condiciones de diseño	3
1.5 Metodología.....	4
1.6 Carta Gantt.....	5
2 CAPÍTULO 2: Estado del arte	6
2.1 Métodos de medición.....	6
2.2 Métodos de estimación.	7
3 CAPITULO 3: Marco teórico	9
3.1 Ecuaciones del movimiento.....	9
3.2 Filtro de Kalman	14
4 CAPITULO 4: Validación del algoritmo filtro de Kalman.....	19
5 CAPITULO 5: Habilitación de la aeronave	22
5.1 Descripción de la aeronave KADET LT-40	22
5.2 Hardware primario.....	24
5.3 Conexión de Hardware primario	28
5.4 Veletas pivotantes.....	29
6 CAPITULO 6: Diseño de método de estimación y medición	31
6.1 Método de medición	31
6.2 Método de estimación.....	35
6.3 Conexión final	40
6.4 Presupuesto de la implementación	42
7 Capítulo 7: Vuelos “Adquisición y procesamientos de datos”	43
7.1 Metodología para realizar los vuelos.....	43

7.2 Vuelos realizados.....	43
8 Capítulo 8: Resultados de métodos diseñados y análisis de resultados.	47
8.1 Método de medición	47
8.2 Método de estimación.....	52
9 Conclusiones y trabajos futuros	56
9.1 Conclusiones.....	56
9.2 Trabajos futuros	58
Referencias	59
Anexo A: Carta Gantt.....	61
Anexo B: Sistema de medición de empuje.....	62
Anexo C: Datos de los vuelos realizados.....	63
Anexo D: Códigos.....	68
Anexo E: Lista Pre-vuelo	84
Anexo F: Otros:	86

Lista de Figuras

Figura 1.1: Representación del ángulo de ataque y diagrama polar ([5, 6]).	1
Figura 3.1: Marcos de referencia (Elaboración propia).	10
Figura 3.2: Rotación al marco de referencia inercial.	11
Figura 3.3: Diagrama de bloques del algoritmo filtro de Kalman.	17
Figura 4.1: Sistema mecánico básico-MRA.	19
Figura 4.2: Simulación mediante KF.	20
Figura 4.3: Simulación mediante EKF.	20
Figura 4.4: Resultados de EKF ante un sistema no lineal.	21
Figura 5.1: Aeronave KADET LT-40.	22
Figura 5.2: Planos de fuselaje.	23
Figura 5.3: Estabilizador Horizontal y vertical.	24
Figura 5.4: Motor SK4250-650.	25
Figura 5.5: Hélice seleccionada.	25
Figura 5.6: Variador de velocidad.	26
Figura 5.7: Servomotor FP-S148.	27
Figura 5.8: Receptor X8R.	27
Figura 5.9: Radio Master TX12.	28
Figura 5.10: Diagrama de conexión.	28
Figura 5.11: Veletas pivotantes.	29
Figura 5.12: Resultado de calibración de la veleta AOA.	30
Figura 5.13: Resultado de calibración de la veleta AOS.	30
Figura 6.1: Sonda diseñada.	32
Figura 6.2: Sonda fabricada e instalada en la aeronave.	32
Figura 6.3: Sensores Ms4525do.	34
Figura 6.4: Arduino UNO R3.	34
Figura 6.5: Conexión método de medición.	35
Figura 6.6: Pixhawk 6c.	38
Figura 6.7: Módulo PM07.	39
Figura 6.8: Esquema de conexión con Pixhawk y el módulo de potencia.	39
Figura 6.9: Entorno del software Mission Planner.	40
Figura 6.10: Conexión de todo el hardware.	41
Figura 7.1: Aceleraciones en vuelo N°2.	44
Figura 7.2: Actitud de la aeronave en vuelo N°2.	44
Figura 7.3: Velocidad angular en vuelo N°2.	45
Figura 7.4: Velocidad inercial en vuelo N°2.	45
Figura 7.5: Velocidad aerodinámica del vuelo N°2.	46
Figura 7.6: Voltajes en veletas pivotantes del vuelo N°2.	46
Figura 8.1: AOA obtenido de la veleta pivotante, vuelo N°2 10-01-2024.	47
Figura 8.2: Presiones medidas en el vuelo N°2.	48
Figura 8.3: Relación de presión normalizada para vuelo N°2.	48

Figura 8.4: Curva obtenida de la regresión polinomial.	49
Figura 8.5: Angulo de ataque obtenido del método de medición, para vuelo N°2.....	49
Figura 8.6:Relación de presión normalizada en vuelo N°3.	51
Figura 8.7: Comparación de presión dinámica en vuelo N°2.....	51
Figura 8.8: Resultado del ángulo de ataque por método algebraico para el vuelo N°2.....	52
Figura 8.9: Resultados del EKF para vuelo N°2.	53
Figura 8.10: Velocidad estimada usando EKF sin las velocidades inerciales.	54
Figura 8.11: AOA estimado utilizando EKF sin las velocidades inerciales.	54

Lista de Tablas

Tabla 5-1: Características del fuselaje.	22
Tabla 5-2: Características del ala	23
Tabla 5-3: Características geométricas del empenaje.	24
Tabla 5-4: Características del motor usado.	25
Tabla 5-5: Canales del modo de vuelo.	28
Tabla 6-1: características del sensor Ms4525do	33
Tabla 6-2: Presupuesto.	42
Tabla 8-1: Error medio, desviación estándar y varianza para resultado de método de medición en vuelo N°2.....	50

Glosario

RPAs	Sistema de aeronave pilotada a distancia.
AOA	Angulo de ataque.
UAV	Aeronave no tripulado.
LTA	Laboratorio de técnicas aeroespaciales.
UdeC	Universidad de Concepción.
cr	Critico
USD	Dolar Estadounidense
AOS	Angulo de deslizamiento lateral.
CG	Centro de gravedad
p, q, r	Velocidades angulares en el cuerpo
u, v, w	Velocidades lineales en el cuerpo
$v_x v_y v_z$	Velocidades en el marco de referencia inercial
b	Marco de referencia del cuerpo
a	Aceleración
I	Inercia
E	Marco de referencia Tierra
R	Matriz de rotación Eueler
\bar{H}	Momento angular
T	Empuje
L	Sustentación
D	Arrastre
KF	Filtro de Kalman
EKF	Filtro de Kalman extendido
x	Vector de estados
y	Vector de salidas
A	Matriz de estados
B	Matriz de entrada
u	Vector de entrada
C	Matriz de salida
\bar{D}	Matriz de transición directa
$k - 1$	Paso de tiempo anterior
k	Paso de tiempo actual
F	Matriz de transición de estados
P	Matriz de error de covarianza
Q	Matriz de covarianza del proceso
y_k	Residual
H	Matriz de observación
K	Ganancia de Kalman
\bar{R}	Ruido de la medición
z_k	Medición

Letras griegas

α	Angulo de ataque.
β	Angulo de deslizamiento lateral.
θ, ϕ, ψ	Angulo pitch, roll, yaw
\emptyset	Angulo de empuje

CAPÍTULO 1: Introducción

1.1 Contexto

En la actualidad y en un futuro se proyecta que los sistemas de vehículos aéreos no tripulados controlados de forma remota (RPAs) sean altamente usados para aplicaciones académicas, civiles y militares. Esto debido a su facilidad de despliegue, costos de mantención y operación reducidos, además por su alta movilidad y capacidad de vuelo [1, 2]. Según el estudio realizado en [3], el valor de mercado global emergente de servicios relacionados con UAV está valorado en más de 127,000 millones de USD. Por otra parte, en [2] se ha reportado que los UAV de ala fija tienen un amplio espectro de aplicaciones debido a la capacidad de volar largas distancias a altas altitudes, como también la capacidad de transportar diferentes tipos de carga útil, donde destacan: cámaras, sensores, radares, Lidar, equipos de comunicación, entre otras. Las aplicaciones de este tipo de vehículos van desde la comunicación, entrega de bienes, percepción remota a usos más específicos como inspecciones en líneas de transmisión, aplicaciones militares y aplicaciones agrícolas.

Para lograr utilizar la máxima capacidad de vuelo de estas aeronaves en una misión, en ocasiones es necesario realizar maniobras poco convencionales como vuelo en espacios limitados, vuelo a altos ángulos de ataques, vuelos a alta velocidad, maniobras militares. Un parámetro de vuelo crítico para dichas maniobras es el ángulo de ataque α (AOA) [4], ya que en aeronaves de ala fija se relaciona directamente con las fuerzas aerodinámicas. El ángulo de ataque se define como el ángulo formado entre la línea de cuerda del perfil alar y el viento relativo local, como se puede apreciar en la Figura 1.1 [5].

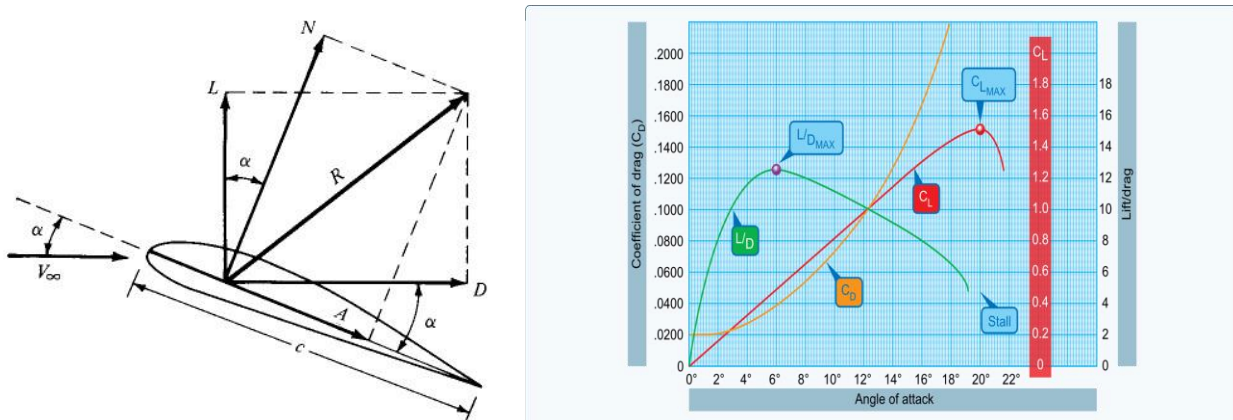


Figura 1.1: Representación del ángulo de ataque y diagrama polar ([5, 6]).

La fuerza de sustentación y arrastre son en función del AOA. La relación entre estos se realiza mediante los coeficientes aerodinámicos, que son valores adimensionales y normalizados por variables específicas de la aeronave. En la Figura 1.1 se presentan un diagrama polar tipo, que relaciona el AOA con los coeficientes mencionados. Se puede observar que hay valores característicos, como el coeficiente de sustentación máximo ($C_{L_{MAX}}$), que es la máxima sustentación

que puede generar un avión antes de entrar en pérdida “*stall*”, este fenómeno es el resultado de una rápida disminución de la sustentación provocada por la separación del flujo de aire en la superficie alar [5], y ocurre en un ángulo de ataque específico denominado ángulo de ataque crítico (α_{cr}).

Debido a las variadas aplicaciones, ya sean autónomas o controladas remotamente, se espera que este tipo de vehículos requiera sistemas (hardware y software) más sofisticados con un uso de herramientas de navegación y sistemas de control diseñados con consideraciones más cuidadosas, donde conocer el AOA es necesario para el diseño de tales sistemas de control. Por ejemplo, en [7] se diseña un controlador para un UAV de ala fija, con una serie de simplificaciones de forma tal que las fuerzas aerodinámicas queden independientes del AOA, ya que no poseen un método para realizar la medición de dicho parámetro. En [8] diseñan e implementan un control para el seguimiento del terreno de una aeronave de ala fija, donde la dinámica está en función del ángulo de ataque. En [9] diseñan un sistema para la medición del ángulo de ataque en una pequeña aeronave no tripulada de ala fija, para posteriormente diseñar un controlador de actitud longitudinal, que tiene como retroalimentación el AOA.

Considerando la importancia de conocer el valor del ángulo de ataque, es necesario que este tipo de aeronaves tengan métodos implementados para la medición y/o estimación de dicho parámetro. En la actualidad existen diversos métodos de medición del AOA, donde se destaca la veleta pivotante que es un cuerpo aerodinámico que es capaz de alinearse con la dirección del viento, el tubo de presión diferencial que se basa en la medición de presión en al menos dos puntos determinados y luego se realiza una correlación con el AOA y por último el tubo de búsqueda nula, basado en la medición de presión en dos puntos, donde tiene un sistema de control que es encargado de anular la diferencia de presión y correlacionar el AOA [10]. Con respecto a los métodos de estimación, existen variados algoritmos que permiten la obtención del AOA, que requieren diferentes parámetros de entrada como aceleraciones, velocidades angulares, deflexión de las superficies de control, etc. Sin embargo, se destacan los algoritmos que se basan en la dinámica longitudinal de la aeronave y que utilizan el filtro de Kalman extendido para la estimación [11].

Actualmente el laboratorio de técnicas aeroespaciales (LTA) de la Universidad de Concepción posee solo la veleta pivotante como método para la medición del ángulo de ataque, sin embargo, este mecanismo de medición no está habilitado en ninguna aeronave y además este método tiene diferentes factores que causan errores en la medición, lo que significa que este método no es el más apropiado. Como no se tienen métodos implementados para la obtención del AOA no es posible el diseño ni la validación de diferentes controladores para estos vehículos, no se pueden analizar datos de vuelo reales en función del AOA y por último no se realizan misiones autónomas debido a la falta de información del ángulo de ataque como dato de retroalimentación.

De lo anterior, surge la necesidad de implementar y validar un método de medición y estimación del ángulo de ataque, los cuales fueron seleccionados y analizados previamente en el proyecto de ingeniería aeroespacial [12]. La implementación se realizará en la aeronave no tripulada de ala fija KADET LT-40 disponible en el LTA. El método de medición a implementar se basa en utilizar un sensor de presión diferencial, donde se correlaciona la diferencia de presión con el AOA. Por otro

lado, el método de estimación a implementar es una fusión de un algoritmo de filtro de Kalman extendido (EKF) con un método algebraico, donde las ecuaciones son rescatadas de la dinámica de una aeronave de ala fija de seis grados de libertad.

1.2 Objetivo general

El objetivo principal de este proyecto es implementar un método de medición y estimación del ángulo de ataque en un aeronave no tripulado de ala fija, para luego comparar el performance entre estos.

1.3 Objetivos específicos

1. **OE1:** Concebir y diseñar el hardware y software necesario para la implementación de un método de medición y estimación del AOA para una aeronave no tripulada de ala fija.
2. **OE2:** Habilitar la aeronave KADET LT-40 junto con la veleta pivotante disponible en el LTA, para cumplir con los requerimientos que imponen los métodos diseñados, además realizar el presupuesto de hardware necesario para dicha habilitación.
3. **OE3:** Implementar el método medición y estimación del ángulo de ataque en la aeronave habilitada.
4. **OE4:** Validar los métodos implementados, realizando al menos dos vuelos para la adquisición de datos y posteriormente analizando el AOA obtenido por los diferentes métodos para así evaluar el performance de estos.

1.4 Condiciones de diseño

Como se ha mencionado, este proyecto tiene como objetivo implementar un método de medición y estimación del ángulo de ataque en un aeronave de ala fija disponible en el LTA, con la finalidad de evaluar el performance de los diferentes métodos y validarlos mediante pruebas de vuelos. En específico se utilizará el aeronave no tripulado KADET LT-40 de la universidad de concepción.

El presente proyecto tiene diferentes limitaciones la cuales se presentan a continuación:

- La aeronave disponible para la implementación del proyecto tiene limitaciones de operación, donde se destaca la velocidad máxima de 25 metros por segundo la cual afecta directamente a las presiones ejercidas en el avión. Por otro lado, la masa de los sistemas a implementar debe ser reducida para no exceder el peso de despegue y maximizar la duración de los vuelos.
- Una limitación importante es que no existe ningún método operativo en el LTA para la obtención del AOA, por lo que se habilitará la veleta y se asumirá que el AOA obtenido por esta es el real.
- Los vuelos que se realizaran están limitado a la meteorología, el espacio disponible en el club de aerodelismo de Concepción, la disponibilidad del piloto y la disponibilidad y estado de las baterías dispuestas en el LTA.
- El método de medición que se implementará estará sujeto a limitaciones con respecto a las mediciones de los sensores a utilizar, debió al error inherente de estos, lo que conllevara a errores en la obtención del AOA.
- El controlador de vuelo a utilizar trae integrado unidades de medida inercial, la cual tienen una resolución y exactitud limitada, por lo que al usar estas mediciones en el método de estimación conllevara a ciertos errores.

- El hardware que se utilizara está limitado económicamente por los recursos que dispone el departamento de ingeniería mecánica de la Universidad de Concepción para este proyecto.

1.5 Metodología

Para el desarrollo de este proyecto se utilizará el modelo Concebir-Diseñar-Implementar-Operar (CDIO), el cual es un ciclo vital completo de un producto, se basa en un análisis sistemático de la problemática, logrando integrar lo teórico con lo práctico para la formación en la ingeniería. A continuación, se detalla la metodología a utilizar para este proyecto, donde se destacan diferentes hitos a cumplir para lograr los objetivos específicos y consigo el objetivo general.

Para lograr el primer objetivo específico se tomará como base lo investigado previamente en el proyecto de ingeniería [12], donde se selecciona el método de medición y estimación del AOA a diseñar en aeronaves no tripulada de ala fija, teniendo énfasis en la tecnología utilizada, en el performance y en la validación de cada método. Posteriormente se diseñará un método de medición y estimación del ángulo de ataque. Para el método de medición que se basa en la diferencia de presión, se analizara analíticamente las presiones a las que estará sometido el sensor y se diseñara la estructura de la sonda para la toma de presión, donde se realizaran diferentes iteraciones, además se diseñara la electrónica para el correcto funcionamiento. Con respecto al método de estimación, se diseña una arquitectura de algoritmo de EKF y método algebraico, que toman en consideración la dinámica longitudinal de una aeronave de ala fija de seis grados de libertad.

Para lograr la habilitación del avión KADET LT-40 disponible en el LTA, primeramente, se recopila información de trabajos anteriores que utilizaron dicho aeronave, con la finalidad de entender el hardware instalado en el avión, además se inspeccionara y reparara la estructura de la aeronave en caso de ser necesario. Posteriormente se analizará el funcionamiento del subsistema de propulsión compuesto por el motor, hélice y variador de velocidad, donde se medirá y estudiará el empuje que genera este sistema. Se realizará una revisión del manual del controlador de vuelo Pixhawk 6c para su correcta instalación en la aeronave y operación (adquisición de datos), además se instalará el receptor que se comunicará con el radio control. Finalmente se integrarán todos los componentes y se realizarán diferentes pruebas de vuelo en el club de aerodelismo de Concepción, con el fin de evaluar el funcionamiento del RPAs. Con respecto a la veleta pivotante, se realizará la calibración correspondiente con ayuda de un controlador tipo Arduino, para encontrar la curva que relaciona el AOA con la posición de la veleta.

Con relación a la implementación del método de medición se imprimirá la sonda diseñada en 3D, se realizará la calibración del mecanismo sensor de presión diferencial más sonda con los datos del primer vuelo a realizar, con el fin de obtener la curva que correlaciona el AOA con la diferencia de presión. Para el método de estimación se programará previamente en el software Matlab el algoritmo de EKF con la finalidad de validar su funcionamiento con ecuaciones conocidas (Masa-Resorte-Amortiguador). Luego se programará el EKF para el modelo diseñado, donde se realizarán una serie de pruebas con datos que se rescatarán del simulador de vuelo Xplane-11, con el fin de corroborar el funcionamiento de dicho modelo.

Por último, para la validación de los métodos implementados se realizarán una serie de vuelos en el club de aerodelismo de concepción, donde en cada vuelo se guardarán los datos obtenidos por el método de medición implementado, la medición de la veleta pivotante y datos de vuelo del controlador necesarios para el método de estimación. Se analizarán y trabajarán los datos obtenidos por el método de medición, los resultados de la estimación y los datos medidos de la veleta para posteriormente realizar una comparación entre estos y así evaluar el performance, considerando el AOA medido por la veleta como el AOA real. Cabe mencionar que los vuelos a realizar incluirán todas las etapas de vuelo con el fin de realizar un análisis de los métodos a implementar más completo, además es importante destacar que el tratamiento de los datos no es en tiempo real, es decir se realiza posterior al vuelo (offline).

1.6 Carta Gantt

Para realizar este proyecto se elaboró una carta Gantt con el fin de planificar el cumplimiento de diferentes hitos, dicha carta se presenta en anexos.

CAPÍTULO 2: Estado del arte

En este capítulo se presenta el análisis del estado del arte de los métodos comúnmente utilizados en la actualidad, para la medición y estimación del ángulo de ataque en aeronaves no tripuladas de ala fija. La información se extrae de lo realizado en [12].

2.1 Métodos de medición

Existen distintos métodos para la medición del ángulo de ataque en aeronaves de ala fija, donde se destaca que la mayoría de los métodos concentran su esfuerzo en la medición de presión en distintos puntos, para luego correlacionar dichas mediciones con el AOA. Sin embargo, existen métodos donde se mide el ángulo a través de sistemas mecánicos.

Uno de los métodos más utilizados para la medición del AOA en aeronaves de ala fija es la veleta pivotante, en este método la medición se realiza mediante cuerpos aerodinámicos que son capaces de alinearse con la dirección del flujo de aire. Las veletas pivotantes traen integrado un potenciómetro, el cual cambia su resistencia eléctrica al cambiar la posición angular de la veleta, así la medición del AOA se logra al correlacionar la caída de voltaje en el potenciómetro con el AOA. Esta medición es influenciada por una serie de factores, donde se destaca la calibración realizada, la distorsión del flujo de aire y la deflexión del mecanismo por fuerzas aerodinámicas [13, 14].

Otro de los métodos utilizados para la medición del AOA, es mediante la medición de presión diferencial, el cual se basa principalmente en la medición de presión en al menos dos puntos determinados para luego correlacionar la diferencia de presión en dichos puntos con el AOA [15]. Dentro de este ámbito existen diferentes formas para la medición de presión, donde el más común es el tubo de presión diferencial, que consta de dos orificios orientados en ángulos iguales con respecto al eje longitudinal para la toma de presión. La sensibilidad y formas de las curvas de calibración depende de la forma geométrica de la nariz del tubo de presión y del lugar en donde se instala el sistema de medición [14].

En [16] se desarrolla un sistema de medición del ángulo de ataque en un aeronave no tripulado de ala fija, donde se utilizan sensores de presión electrónicos de bajo costo y un microcontrolador Arduino. En dicho trabajo se destaca el método de calibración para obtener la curva que relaciona el AOA con la diferencia de presión, ya que se calibra tomando cuatro velocidades características de una aeronave. Además, se realiza una normalización de la diferencia de presión para obtener una curva de correlación que sea independiente de la velocidad de vuelo [17].

En [14, 13] se presentan diferentes métodos para la medición del ángulo de ataque, donde se destaca que se utiliza la medición de presión como base. Se hace mención del método de medición por dispositivos de búsqueda de presión diferencial nula, el cual tiene un mecanismo giratorio que sea capaz de hacer la diferencia de presión entre dos puntos nula, así se relaciona la rotación realizada con el AOA. Otros métodos son la medición de diferencia de presión usando extensómetros, donde se relaciona la deformación de un diafragma con la diferencia de presión y la diferencia de presión con el AOA, por otro lado, está la medición del AOA mediante capacitancia variable, en el cual se correlaciona el cambio de capacitancia con la diferencia de presión y esta diferencia de presión con el

AOA, en este método existen diferentes tipos de sensores capacitivos donde se destaca los sensores basados en placa de circuitos impresos, debido a su capacidad de adaptarse a la forma del ala.

2.2 Métodos de estimación.

Todos los métodos de estimación investigados utilizan en su arquitectura al menos un algoritmo de filtro de Kalman extendido, donde los métodos se diferencian entre sí por el vector de estado, medición y control utilizados, además de las ecuaciones que modelan el movimiento de la aeronave.

En [18] se realiza una simulación de vuelo de una aeronave, donde se propone un método de estimación del ángulo de ataque mediante la resolución de ecuaciones algebraicas, la formulación del problema se basa en las ecuaciones de traslación y rotación de la aeronave. En primer lugar, realizan una estimación mediante un algoritmo de filtro de Kalman extendido (EKF) para la actitud de la aeronave (ángulos de pitch, roll y yaw) usando velocidades angulares del cuerpo. Luego se propone el método de estimación de AOA y el ángulo de deslizamiento lateral (AOS), el cual consiste en resolver dos ecuaciones no lineales en forma simultánea, las cuales tienen una serie de simplificaciones dinámicas y solo utilizan mediciones de aceleraciones lineales en el cuerpo y velocidades angulares. Sin embargo, las simplificaciones realizadas eliminan parte de la cinemática rotacional del problema, por lo que realizan una corrección de los resultados obtenidos mediante ecuaciones discretizadas y ponderadas por parámetros encontrados experimentalmente.

Por otro lado, en [19] proponen un algoritmo de EKF para la estimación del ángulo de ataque, donde el primer paso es la estimación de las velocidades en el cuerpo usando las mediciones de velocidades en el sistema inercial, para luego calcular el AOA usando la siguiente ecuación:

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \quad (2.1)$$

Posterior realizan un análisis de los resultados, donde se obtiene que el algoritmo de EKF utilizado es sensible a los ruidos del sensor, por lo que los resultados de la estimación presentan errores considerables.

En [20] se realiza una fusión de los métodos anteriores, donde utilizan los resultados de la estimación del AOA mediante la solución del sistema de ecuaciones no lineales como una medición para el algoritmo de Kalman extendido, es decir se realiza una estimación de los ángulos aerodinámicos mediante un algoritmo de EKF, donde en la etapa de actualización se introducen los resultados del AOA calculados algebraicamente con la finalidad de reducir el error en la estimación final.

Otros método de estimación se presenta en [21] donde aparte de realizar un algoritmo de EKF se propone un filtro complementario para la estimación del ángulo de ataque, el cual se trabaja en el dominio frecuencia y se compone de dos partes, un filtro de paso bajo y un filtro de paso alto, en cada parte las ecuaciones dependen de una serie de factores propios del filtro como lo es la frecuencia de ruptura y relación de amortiguamiento. Las mediciones que se necesitan para este método son obtenidas de la unidad de medida inercial. En la misma investigación proponen un método de minimización de errores el cual se basa netamente en las ecuaciones cinemáticas que describen la aeronave, donde es necesario realizar una serie de integraciones numéricas e iteraciones para estimar el AOA.

Por último, un método de estimación más complejo se realiza en [22], donde se realiza una fusión de mediciones de aceleraciones, velocidades angulares, ángulos de Euler y velocidad, las cuales son obtenidas de la IMU. Este método es un algoritmo de filtro de Kalman extendido, donde las ecuaciones de la etapa de predicción se obtienen diferenciando las ecuaciones de AOA, AOS y posteriormente reemplazando en las ecuaciones que describen la traslación, por otro lado, las ecuaciones de actualización se derivan de las ecuaciones de sustentación y fuerza lateral, las cuales dependen de una serie de coeficientes aerodinámicos y la deflexión de las superficies de control.

CAPITULO 3: Marco teórico

3.1 Ecuaciones del movimiento

Para realizar una estimación del ángulo de ataque, es necesario conocer el modelo matemático que describe la dinámica de vuelo de una aeronave, por lo que en esta sección se presentara la derivación de las ecuaciones del movimiento para una aeronave de ala fija con seis grados de libertad. Es importante mencionar que las ecuaciones a presentar se obtienen realizando una serie simplificaciones. La información presentada en este apartado se obtiene de [23].

3.1.1 Marcos de referencia

Para desarrollar las ecuaciones del movimiento, es necesario dominar diferentes sistemas de referencias debido a la conveniencia de definir diferentes vectores en distintos marcos como las fuerzas aerodinámicas, empuje, etc. Estos sistemas de referencia son tres en particular, el sistema Tierra, sistema del cuerpo y el sistema de estabilidad.

1. Marco de referencia Tierra

Este sistema se caracteriza por estar fijo a la tierra, en este problema se asume que será el sistema inercial lo que será útil para aplicar la segunda ley de Newton. Este sistema se denota con el subíndice "E". El eje Z_E siempre apunta al centro de la tierra, los ejes X_E e Y_E son ortogonales entre sí y perpendiculares al eje Z_E , la dirección de estos ejes comúnmente se ajusta a conveniencia del problema, sin embargo, es común que el eje X_E apunta hacia el norte y el Y_E apunte hacia el Este. Por otro lado, en este trabajo se asume que el origen de este sistema de referencia coincide con el CG de la aeronave.

2. Marco de referencia del cuerpo

Este marco de referencia esta fijo al cuerpo de la aeronave, donde su origen está en el centro de gravedad del avión. A lo largo de este trabajo se denotará con el subíndice "b", el eje X_b coincide con el eje longitudinal del avión y es positivo en la dirección de avance, el eje Y_b es perpendicular al eje X_b y apunta al ala derecha de la aeronave, por último, el eje Z_b es ortogonal a los ejes anteriores y apunta hacia abajo de la aeronave.

3. Marco de referencia de estabilidad

El sistema de estabilidad es un marco adicional que se utiliza para simplificar la definición de las fuerzas aerodinámicas que actúan en la aeronave, se representa por el subíndice "S". La principal característica es que el eje X_S coincide con la dirección del viento relativo, lo que implica que el eje X_S este separado angularmente por el ángulo de ataque con respecto al eje X_b .

A continuación, se presenta una imagen que resume la disposición de los marcos de referencia.

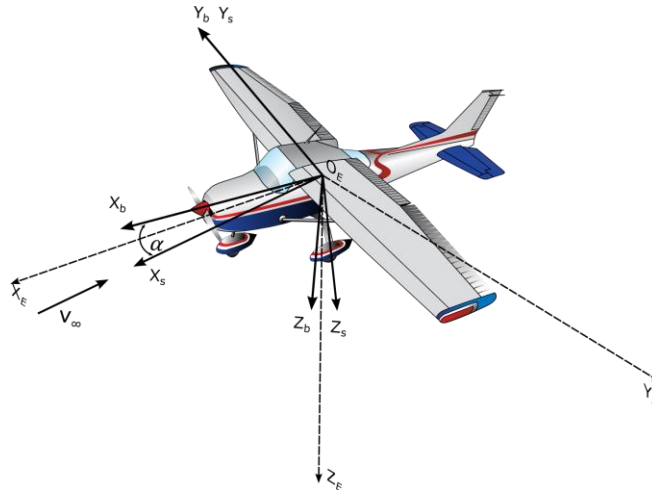


Figura 3.1: Marcos de referencia (Elaboración propia).

3.1.2 Ángulos de Euler y transformación de marcos de referencia.

Como se mencionó anteriormente en ocasiones resulta útil trabajar en diferentes marcos de referencias con el fin de simplificar las derivaciones matemáticas, sin embargo, para aplicar las ecuaciones de Newton se requiere que todas las fuerzas, momentos y variables físicas sean representados solo en un marco de referencia. Es por esto que es necesario realizar una breve explicación de los ángulos de Euler y sus matrices de rotación.

Los ángulos de Euler son utilizados para describir y controlar la orientación de un cuerpo en el espacio con respecto al sistema inercial, estos ángulos son llamados yaw (ψ), pitch (θ) y roll (ϕ). El ángulo de yaw es el ángulo formado en el plano horizontal entre el eje X_b y el eje X_E , este ángulo es el que da la dirección a la aeronave. El ángulo de pitch es ángulo que refleja el movimiento de cabeceo de una aeronave y está definido verticalmente entre el eje X_b y el eje X_E . Por último, el ángulo de roll es el que describe la rotación alrededor del eje X_b , es el ángulo formado entre el eje Y_b y la horizontal en el plano $(Y, Z)_E$.

Para realizar la transformación de una magnitud física de un sistema de referencia a otro es necesario realizar una secuencia de rotaciones en un orden específico, donde los valores de estas rotaciones son los ángulos de Euler. Si se requiere rotar un vector desde un marco de referencia distinta al inercial al inercial, se debe rotar en torno al yaw, pitch y finalmente en torno al roll, esto debido a que las rotaciones vienen dadas matemáticamente por multiplicaciones entre matrices y la multiplicación de matrices no es conmutativa. En la Figura 3.2 se presenta la descomposición de los vectores en los diferentes ejes de referencia, cuando una magnitud se rota a cierto grado yaw, pitch y roll.

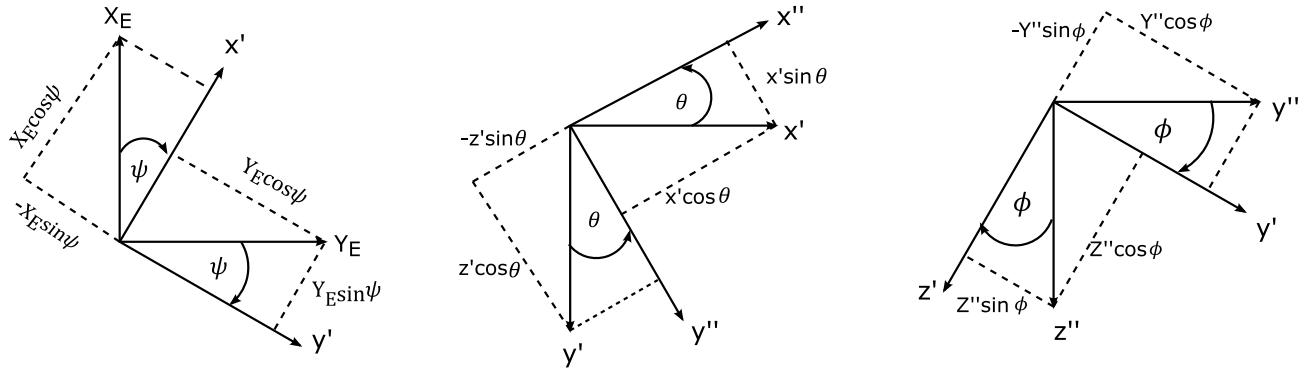


Figura 3.2: Rotación al marco de referencia inercial.

Matemáticamente la rotación está dada por una multiplicación sucesiva de tres matrices, la matriz resultante se denomina matriz de rotación de Euler y denota con la letra R. Es importante mencionar que la matriz de rotación para pasar del marco de referencia inercial a un marco de referencia no inercial es diferente, debido a que la multiplicación de matrices se realiza de orden inverso.

Para ejemplificar lo anterior se rotará el vector de la fuerza peso F_{weight_E} del marco de referencia inercial al marco de referencia del cuerpo.

$$F_{weight_b} = R \cdot F_{weight_E} = R \cdot \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_I \quad (3.1)$$

Donde R es la matriz de rotación que se determina de la siguiente manera.

$$R = R_1(\phi)R_2(\theta)R_3(\psi)$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Así el vector fuerza representado en el marco de referencia del cuerpo es,

$$F_{weight_b} = R \cdot \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_{Inercial} = \begin{bmatrix} -mg \cdot \sin(\theta) \\ mg \cdot \sin(\phi)\cos(\theta) \\ mg \cdot \cos(\phi)\cos(\theta) \end{bmatrix}_b \quad (3.2)$$

3.1.3 Ecuaciones de traslación

Las ecuaciones de traslación se derivan a partir de la segunda ley de Newton, estas ecuaciones están relacionadas con las fuerzas que actúan en la aeronave. Dicha ley establece que la tasa de cambio en el tiempo del momento lineal es igual a la suma de las fuerzas que actúan en la aeronave. Para aplicar esta ley se realizan tres simplificaciones principales: La aeronave es un cuerpo rígido, la masa es constante en el tiempo y las fuerzas actúan en el centro de gravedad del avión. La ecuación de la segunda ley de Newton es la siguiente

$$\left[\frac{d(m \cdot \vec{V})}{dt} \right]_{Inercial} = m \bar{a}_{inercial} = \bar{F} \quad (3.3)$$

La ecuación (3.3) es válida aplicarla solo para sistemas inerciales, sin embargo, es posible representarla en el marco de referencia fija al avión teniendo en cuenta la contribución de las velocidades en la aceleración inercial, esto debido a que el sistema está girando entorno a un sistema inercial. Así la aceleración inercial representada en el cuerpo queda de la siguiente manera,

$$(\bar{a}_{inercial})_b = \vec{V}_b + \vec{w}_b \times \vec{V}_b \quad (3.4)$$

En la ecuación (3.4) $\vec{V}_b = [u \ v \ w]_b$ son las velocidades lineales del centro de gravedad en los ejes del cuerpo y $\vec{w}_b = [p \ q \ r]_b$ son las velocidades angulares del avión. Así remplazando esto en la aceleración inercial y luego en (3.3) se obtiene lo siguiente.

$$m \cdot \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}_b = F_b \quad (3.5)$$

Donde F_b representa la sumatoria de fuerzas que actúan en el avión, dichas fuerzas se descomponen en la fuerza peso, fuerzas aerodinámicas y fuerza de empuje.

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix}_b = \begin{bmatrix} F_{weight_X} + F_{A_X} + F_{T_X} \\ F_{weight_Y} + F_{A_Y} + F_{T_Y} \\ F_{weight_Z} + F_{A_Z} + F_{T_Z} \end{bmatrix}_b \quad (3.6)$$

En la ecuación (3.2) ya se realizó la rotación de la fuerza peso al marco de referencia del cuerpo, por lo que falta realizar lo mismo con las fuerzas aerodinámicas y el empuje. Anteriormente se mencionó que el marco de referencia de estabilidad es útil para definir las fuerzas aerodinámicas, debido a que la fuerza de sustentación coincide con el eje Z_S y la fuerza de arrastre con el eje X_S , así la rotación al sistema del cuerpo se consigue mediante la aplicación de la matriz R_2 en función del ángulo de ataque.

$$\begin{bmatrix} F_{A_X} \\ F_{A_Y} \\ F_{A_Z} \end{bmatrix}_b = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} -D \\ F_{A_y} \\ -L \end{bmatrix}_S = \begin{bmatrix} -D \cos(\alpha) + L \sin(\alpha) \\ F_{A_{yS}} \\ -D \sin(\alpha) - L \cos(\alpha) \end{bmatrix} \quad (3.7)$$

Por otro lado, el empuje se asume que se separa angularmente ϕ_T con respecto al eje X_b , por lo que no es necesario aplicar la matriz de rotación. Tomando esto en consideración y remplazando (3.7) y (3.2) en (3.6) se obtiene las tres ecuaciones que describen la traslación de la aeronave.

$$m \cdot \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}_b = \begin{bmatrix} -mg \cdot \sin(\theta) + (-D \cos(\alpha) + L \sin(\alpha)) + T \cos(\phi_T) \\ mg \cdot \sin(\phi) \cos(\theta) + F_{A_Y} + F_{T_Y} \\ mg \cdot \cos(\phi) \cos(\theta) + (-D \sin(\alpha) - L \cos(\alpha)) - T \sin(\phi_T) \end{bmatrix} \quad (3.8)$$

Es importante mencionar que en estas ecuaciones las fuerzas aerodinámicas y de empujes están escritas de forma general, pues estas son afectadas por una serie de factores como lo son los coeficientes aerodinámicos, razón de cambio del AOA, deflexiones de las superficies de control, etc.

3.1.4 Ecuaciones de rotación

Al igual que en las ecuaciones de traslación, las ecuaciones de rotación son derivadas de la segunda ley de Newton, que establece que la tasa de cambio del momento angular de un cuerpo es igual a la sumatoria de los momentos aplicados.

$$\left[\frac{d\vec{H}}{dt} \right]_{Inercial} = \sum \vec{M} \quad (3.9)$$

Al aplicar la ecuación (3.9) en un marco de referencia inercial y luego rotarlo a un marco de referencia no inercial se debe tener en cuenta las velocidades relativas entre el sistema del cuerpo y el sistema inercial, lo que se logra con (3.10).

$$\left[\frac{d\vec{H}}{dt} \right]_{Inercial_b} = \left[\frac{d\vec{H}}{dt} \right]_b + \vec{\omega}_b \times \vec{H}_b \quad (3.10)$$

Considerando una distribución de masas en el cuerpo uniforme se tiene que el momento angular es el producto entre el tensor de inercias y la velocidad angular del cuerpo.

$$\vec{H}_b = \vec{I} \cdot \vec{\omega}_b \quad (3.11)$$

Desarrollando (3.11) y considerando que la aeronave es simétrica con respecto a los planos X_b, Y_b y Y_b, Z_b lo que se traduce que los productos de inercia entre estos planos son nulos se obtiene lo siguiente.

$$\vec{H}_b = \begin{bmatrix} pI_{xx} - rI_{xz} \\ qI_{yy} \\ rI_{zz} - pI_{xz} \end{bmatrix} \quad (3.12)$$

Derivando (3.12) con respecto al tiempo y reemplazando en (3.10) se logra obtener la tasa de cambio de momento angular representado en el marco del cuerpo.

$$\left[\frac{d\vec{H}}{dt} \right]_{Inercial_b} = \begin{bmatrix} \dot{p}I_{xx} + qr(I_{zz} - I_{yy}) - (\dot{r} + pq)I_{xz} \\ \dot{q}I_{yy} - pr(I_{zz} - I_{xx}) + (p^2 - r^2)I_{xz} \\ \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (qr - \dot{p})I_{xz} \end{bmatrix} \quad (3.13)$$

Para completar la ecuación (3.9) solo falta definir los momentos que actúan en la aeronave, los cuales se dejaron expresados de forma genérica y se realiza la distinción entre momentos que resultan de las fuerzas aerodinámicas y momentos resultantes del empuje.

$$\begin{bmatrix} \dot{p}I_{xx} + qr(I_{zz} - I_{yy}) - (\dot{r} + pq)I_{xz} \\ \dot{q}I_{yy} - pr(I_{zz} - I_{xx}) + (p^2 - r^2)I_{xz} \\ \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (qr - \dot{p})I_{xz} \end{bmatrix} = \begin{bmatrix} L_A + L_T \\ M_A + M_T \\ N_A + N_T \end{bmatrix} \quad (3.14)$$

En (3.14) se presentan las ecuaciones de rotación de un avión, considerando que la masa del avión es constante, que es un cuerpo rígido, que la distribución de la masa es constante y que las fuerzas aerodinámicas actúan en el CG. Los términos $L_A, M_A,$ y N_A corresponden al momento de balanceo, cabeceo y guiñada respectivamente.

3.1.5 Resumen de ecuaciones del movimiento

Juntando las ecuaciones de traslación (3.8) y las de rotación (3.14) se obtiene un sistema de ecuaciones que describen la dinámica del aeronave, sin embargo, a estas seis ecuaciones se deben agregar otras tres ecuaciones que relacionan las velocidades angulares del cuerpo con los ángulos y razón de cambio de los ángulos de Euler. Así el set de ecuaciones completas se presenta a continuación.

$$\begin{aligned}
m \cdot (\dot{u} + qw - rv) &= -mg \cdot \sin(\theta) + (-D\cos(\alpha) + L\sin(\alpha)) + T\cos(\Phi_T) \\
m \cdot (\dot{v} + ru - pw) &= mg \cdot \sin(\phi)\cos(\theta) + F_{A_Y} + F_{T_Y} \\
m \cdot (\dot{w} + pv - qu) &= mg \cdot \cos(\phi)\cos(\theta) + (-D\sin(\alpha) - L\cos(\alpha)) - T\sin(\Phi_T) \\
\dot{p}I_{xx} + qr(I_{zz} - I_{yy}) - (\dot{r} + pq)I_{xz} &= L_A + L_T \\
\dot{q}I_{yy} - pe(I_{zz} - I_{xx}) + (p^2 - r^2)I_{xz} &= M_A + M_T \\
\dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (qr - \dot{p})I_{xz} &= N_A + N_T \\
p &= \dot{\phi} - \dot{\psi} \cdot \sin(\theta) \\
q &= \dot{\theta} \cdot \cos(\phi) + \dot{\psi} \cdot \cos(\theta) \cdot \sin(\phi) \\
r &= \dot{\psi} \cdot \cos(\phi) \cdot \cos(\theta) - \dot{\theta} \cdot \sin(\phi)
\end{aligned} \tag{3.15}$$

3.1.6 Ecuaciones del movimiento longitudinal

Las ecuaciones del movimientos presentadas anteriormente pueden dividirse en dos conjuntos, cada uno con tres ecuaciones. Un conjunto modela el movimiento longitudinal mientras que el otro aborda la dinámica lateral y direccional. Para este trabajo es importante presentar las ecuaciones que modelan el comportamiento longitudinal, ya que es donde el ángulo de ataque tiene un mayor impacto.

$$\begin{aligned}
m(\dot{u} + qw - rv) &= -mg \cdot \sin(\theta) + (-D\cos(\alpha) + L\sin(\alpha)) + T\cos(\Phi_T) \\
\dot{q}I_{yy} - pe(I_{zz} - I_{xx}) + (p^2 - r^2)I_{xz} &= M_A + M_T \\
m(\dot{w} + pv - qu) &= mg \cdot \cos(\phi)\cos(\theta) + (-D\sin(\alpha) - L\cos(\alpha)) - T\sin(\Phi_T)
\end{aligned} \tag{3.16}$$

Al usar este set de ecuaciones se debe tener en consideración que solo representa los movimientos de traslación en el plano XZ_b y el movimiento de cabeceo.

3.2 Filtro de Kalman

El filtro de Kalman es una herramienta esencial en la teoría de control y estimación de sistemas dinámicos. Este algoritmo recursivo está basado netamente en la probabilidad y estadística bayesianas, esto permite que en su formulación incluya ruido y posibles incertidumbres que afectan al sistema como también a los sensores. El filtro de Kalman es utilizado para la fusión de sensores y para la estimación de variables ocultas de un sistema, su funcionamiento se basa al combinar de forma ponderada la información de dos fuentes principales, las mediciones actuales de una variable y la predicción del estado que viene dada por el modelo matemático que describe el problema.

Para comprender el algoritmo del filtro es necesario revisar algunos conceptos esenciales que vienen dados por la teoría de control y por las estadísticas bayesianas.

3.2.1 Espacio estado

Este es un concepto que se utiliza para representar y analizar sistemas dinámicos. Es una formulación matemática que describe la evolución del sistema en el tiempo mediante una relación entre sus entradas y salidas, basadas en las ecuaciones diferenciales que lo rigen. La representación en el espacio estados reemplaza una ecuación diferencial de orden n con n ecuaciones de primer orden. El espacio estado queda definido con las siguientes ecuaciones

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + w(t) \quad (3.17)$$

$$y(t) = C(t)x(t) + \bar{D}(t)u(t) + v(t), \quad (3.18)$$

donde $x(t)$ es el vector de estados del sistema, es decir contiene las variables de interés del sistema dinámico. A es la matriz de estado, esta contiene los parámetros que gobiernan la dinámica del vector de estados. B se denomina matriz de entrada y u es el vector de entradas, estos dos últimos determinan como las variables de control o entradas (fuerzas, perturbaciones) que afectan a la evolución del sistema (vector estados). En la ecuación (3.18) $y(t)$ se denomina vector de salida del sistema y representan las variables que se pueden medir u observar directamente, $C(t)$ es la matriz de salida y contiene los parámetros que relacionan las variables de estado con el vector de salida y $\bar{D}(t)$ se denomina matriz de transición directa y relaciona las entradas con las salidas. Por último, $w(t)$ representa el ruido de proceso, que captura las incertidumbres que afectan al sistema y que no son capturadas por el resto de las matrices y $v(t)$ el ruido de la medición [24, 25].

3.2.2 Algoritmo de filtro de Kalman lineal

Como se mencionó anteriormente este algoritmo es recursivo y se mueve entre un paso de predicción y un paso de corrección de la estimación. A continuación, se presentarán las ecuaciones matemáticas que describen dichos pasos.

Predicción del estado:

La ecuación para la predicción es la siguiente,

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + B_k u_k \quad (3.19)$$

Esta ecuación predice los estados actuales a partir de las estimaciones anteriores, donde F es la matriz de transición de estados, $x_{k|k-1}$ es la predicción en el paso de tiempo actual basa en la información hasta el paso de tiempo anterior y $\hat{x}_{k-1|k-1}$ es la estimación del paso de tiempo anterior. Es necesario considerar la incertidumbre o dispersión asociada a cada variable, esto se incluye en una matriz llamada matriz de covarianza de estados P , los elementos de la diagonal de esta matriz representan la varianza de la estimación de los estados y el resto de los elementos son la covarianza entre los estados.

$$\hat{P}_{k|k-1} = F \hat{P}_{k-1|k-1} F^T + Q_k \quad (3.20)$$

$P_{k|k-1}$ es la predicción de la matriz de covarianza en el paso de tiempo actual con la información del paso de tiempo anterior, $\hat{P}_{k-1|k-1}$ es la matriz de covarianza de estado estimada en el paso de tiempo anterior y Q es la matriz de covarianza de ruido del proceso, este último término se suma para tener

en cuenta todo el ruido no relacionado en el modelo a lo largo del tiempo, este se selecciona habitualmente realizando tuning y es propio de cada sistema.

Paso de corrección:

Este paso también se denomina paso de actualización y utiliza las medidas como un canal de retroalimentación para la predicción anteriormente realizada. Las ecuaciones para aplicar en cada paso de tiempo son las siguientes.

$$\begin{aligned}
 y_k &= z_k - H\hat{x}_{k|k-1} \\
 K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \\
 \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k y_k \\
 \hat{P}_{k|k} &= (I - K_k H)P_{k|k-1}
 \end{aligned} \tag{3.21}$$

El paso de actualización comienza calculando el residual y_k entre la medición z_k y la predicción $\hat{x}_{k|k-1}$, la matriz de observación " H ", transforma el espacio estados en espacio de medidas con el fin de poder establecer una relación entre estas. A continuación, se calcula la ganancia de Kalman K_k , el cual es un factor de peso para corregir el estado, establece que porcentaje se le da a la medición y que porcentaje se le da a la predicción, estos pesos están relacionados con la matriz de covarianza del proceso y la matriz de covarianza de la medición R , que dependerá de los sensores utilizados. Por último, con la ganancia de Kalman, la predicción y el residual es posible calcular la estimación del estado $\hat{x}_{k|k}$. Por otro lado, al ser un algoritmo recursivo la matriz de covarianza del proceso igual se debe actualizar en cada paso de tiempo.

Es importante destacar que las ecuaciones asociadas al algoritmo del filtro de Kalman son formuladas de manera discreta en el tiempo, esto se debe a la naturaleza discreta de las mediciones. Esta formulación permite observar la transición de las variables desde un paso de tiempo anterior " $k - 1$ " hacia el paso de tiempo actual " k ".

Los valores estimados por este algoritmo siempre deben estar entre la predicción y la medición, la ganancia de Kalman es la encargada de distribuir los pesos entre estos valores, además esta ganancia a lo largo del tiempo tiende a converger a un valor, ya que la estimación en cada paso va mejorando. Hay que destacar que el algoritmo presentado solo sirve para sistemas lineales, sin embargo, es la base para extensiones de este filtro.

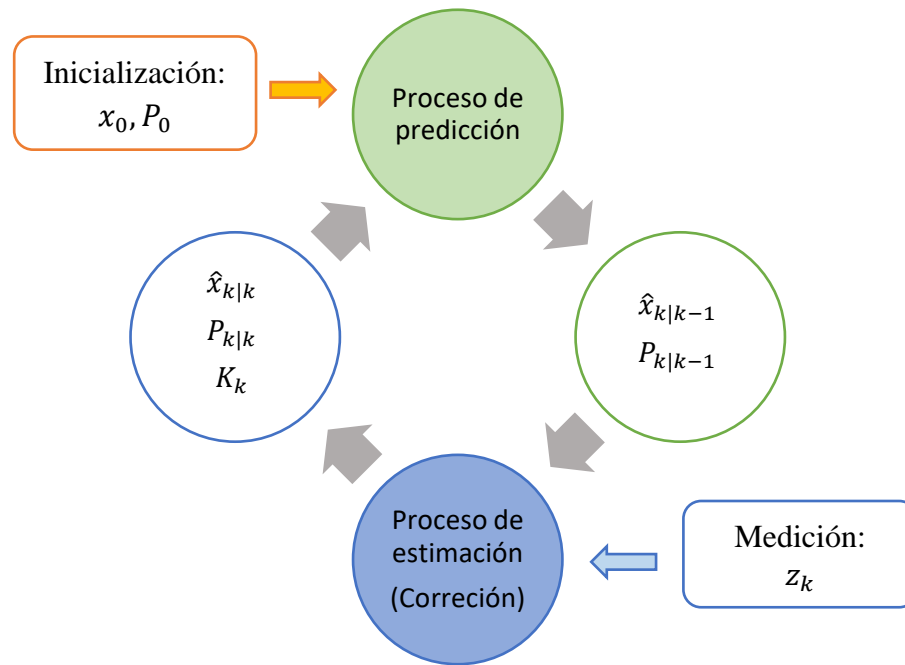


Figura 3.3: Diagrama de bloques del algoritmo filtro de Kalman.

3.2.3 Algoritmo de filtro de Kalman extendido

El interés de este proyecto es la estimación del ángulo de ataque en un UAV de ala fija, donde las ecuaciones que gobiernan este sistema son no lineales. El algoritmo presentado en 3.2.2 funciona correctamente solo para sistemas lineales, sin embargo, existe una modificación denominada filtro de Kalman extendido (EKF) que estima los estados de sistemas no lineales. El EKF logra manejar la no linealidad linealizando el sistema en un punto determinado por cada paso de tiempo. El sistema no lineal se puede formular de la siguiente manera.

$$\begin{aligned}\hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h(\hat{x}_{k|k-1}, v_k)\end{aligned}\quad (3.22)$$

Donde f es la función no lineal del sistema y h es la función de observación del sistema que igualmente puede ser no lineal. El resto de los términos siguen siendo los mismos que en el filtro de Kalman lineal. Para convertir el sistema no lineal a uno lineal el EKF calcula los jacobianos de las funciones no lineales con respecto a los estados o al ruido del proceso. Así las ecuaciones quedan de la siguiente manera.

Paso de predicción:

$$\begin{aligned}F &= \frac{df(x, u)}{dx} \\ \hat{x}_{k|k-1} &= f(\hat{x}_{k-1|k-1}, u) \\ P_{k|k-1} &= F\hat{P}_{k-1|k-1}F^T + Q_k\end{aligned}\quad (3.23)$$

Paso de corrección:

$$\begin{aligned}
 H &= \frac{dh(x_k)}{dx} \\
 y_k &= z_k - h\hat{x}_{k|k-1} \\
 K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \\
 \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k y_k \\
 \hat{P}_{k|k} &= (I - K_k H)P_{k|k-1}
 \end{aligned} \tag{3.24}$$

Es importante mencionar que el EKF es subóptimo, ya que si el comportamiento del sistema es considerablemente no lineal el rendimiento se degrada. Lo presentado en este capítulo es la parte básica del algoritmo de filtro de Kalman y la información presentada es rescatada de [24, 26].

CAPITULO 4: Validación del algoritmo filtro de Kalman.

Antes de aplicar el algoritmo EKF para la estimación de los ángulos aerodinámicos es necesario comprobar el rendimiento del algoritmo ante sistemas dinámicos conocidos, esto para tener en consideración en cómo afectan los diferentes parámetros del algoritmo en la estimación y tener claridad de posibles divergencias de los resultados. En la Figura 4.1 se presenta un sistema masa-resorte-amortiguador (MRA) con el cual se validará el algoritmo de Kalman.

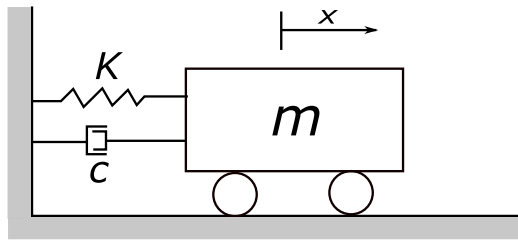


Figura 4.1: Sistema mecánico básico-MRA.

Si no se consideran perturbaciones o fuerzas externas la ecuación diferencial que describe la dinámica del sistema queda de la siguiente manera,

$$\ddot{x} = \frac{kx}{m} + \frac{c\dot{x}}{m} \quad (4.1)$$

Donde, m es la masa del sistema, k es la constante de rigidez del resorte y c es la constante de amortiguamiento. Se sabe que la ecuación (4.1) es una ecuación lineal y su comportamiento depende de las relaciones entre los parámetros.

En primer lugar, se aplicará el algoritmo de filtro de Kalman lineal, donde se asume que solo se tiene un sensor que medirá la posición del sistema y la velocidad se asumirá como un estado oculto que se estimará mediante el KF. En la Figura 4.2 se presenta la posición y velocidad reales y estimadas.

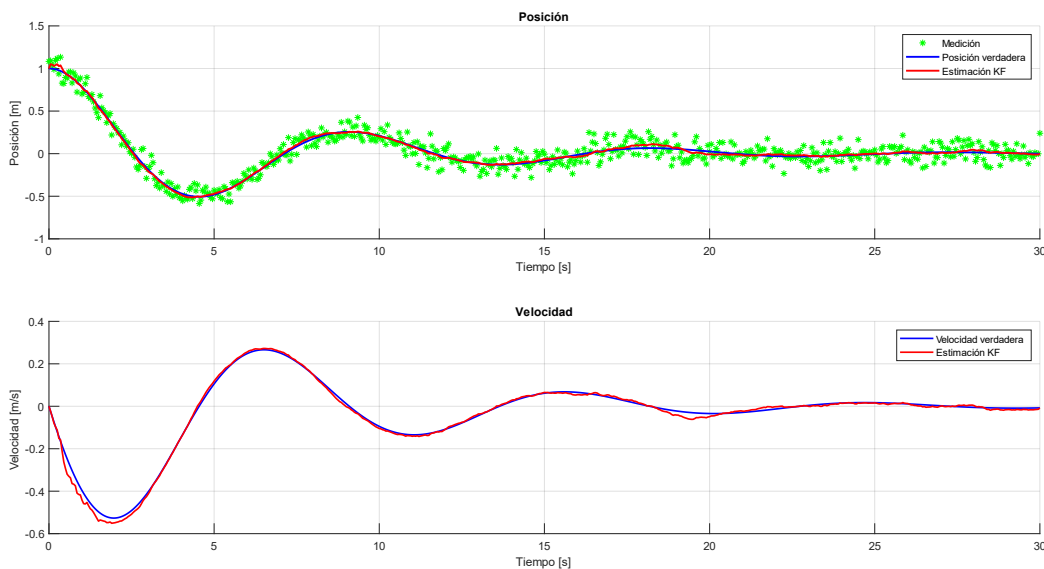


Figura 4.2: Simulación mediante KF.

Los resultados de la Figura 4.2 son para un sistema con masa de $10 [kg]$, una constante de amortiguamiento de $3 \left[\frac{Ns}{m} \right]$ y una constante de rigidez de $5 [Nm]$. Además, la medición tiene un ruido agregado que simula la característica del sensor con una covarianza de $0,0142 [m]$. Se puede observar que las estimaciones realizadas se ajustan de maneras consistente a las curvas reales, lo que indica que el algoritmo programado de KF funciona y tiene un buen desempeño. Con un error cuadrático medio de raíz (RMSE) para la posición de $0,0209 [m]$ y para la velocidad de $0,0112 \left[\frac{m}{s} \right]$.

Ahora se aplicará el algoritmo de filtro de Kalman extendido al mismo sistema dinámico, con los mismos valores de las constantes y ruido de la medición. Los resultados obtenidos se presentan en la Figura 4.3.

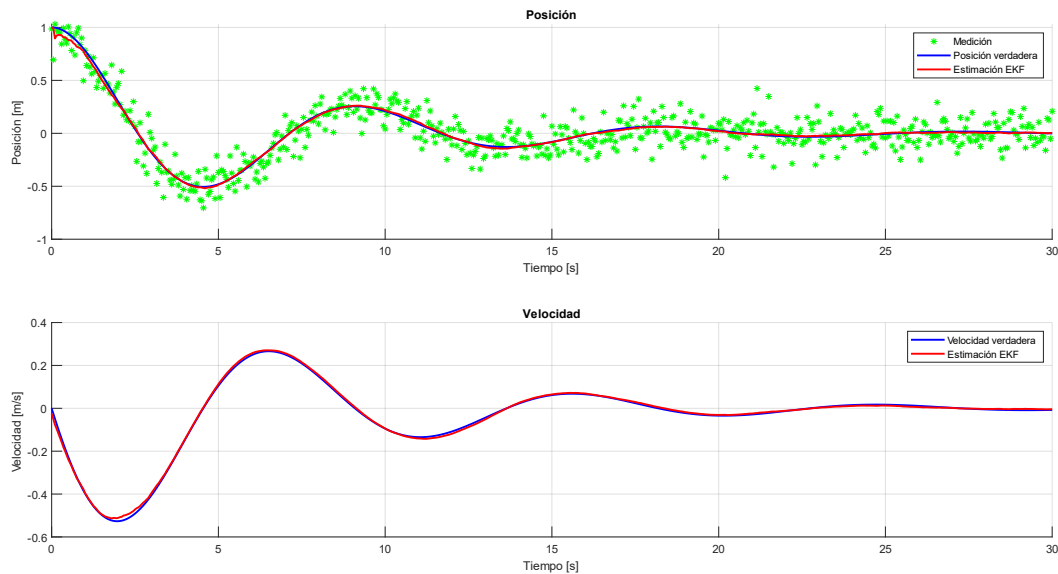


Figura 4.3: Simulación mediante EKF.

Se puede observar que al aplicar el algoritmo EKF al mismo sistema dinámico los resultados se ajustan de manera óptima a la posición y velocidad real. Se obtienen valores de RMSE de $0,0142 [m]$ y $0,0065 \left[\frac{m}{s} \right]$ correspondiente a la posición y velocidad.

Para validar el algoritmo de EKF frente a sistemas no lineales se elige la ecuación de Duffing. Esta ecuación se utiliza para modelar osciladores amortiguados y excitados en el cual la elasticidad no es lineal.

$$\ddot{x} + \gamma\dot{x} + \omega^2x + \beta x^3 = \text{acos}(\rho t) \quad (4.2)$$

Los valores de las constantes para realizar la simulación son los siguientes:

- Constante de amortiguamiento, $c = 0,3 \left[\frac{Ns}{m} \right]$

- Frecuencia natural, $w = 1 \left[\frac{rad}{s} \right]$
- Parámetro cubico no lineal $\beta = 1$
- Amplitud de la fuerza externa, $a = 0,5 [N]$
- Frecuencia de excitación $\rho = 1,5 \left[\frac{rad}{s} \right]$

Además, al igual que los casos anteriores se asume que se tiene un sensor para medir la posición, el cual tiene un error de medición con covarianza de $0,026 [m]$. Los resultados de la simulación se presentan en la Figura 4.4.

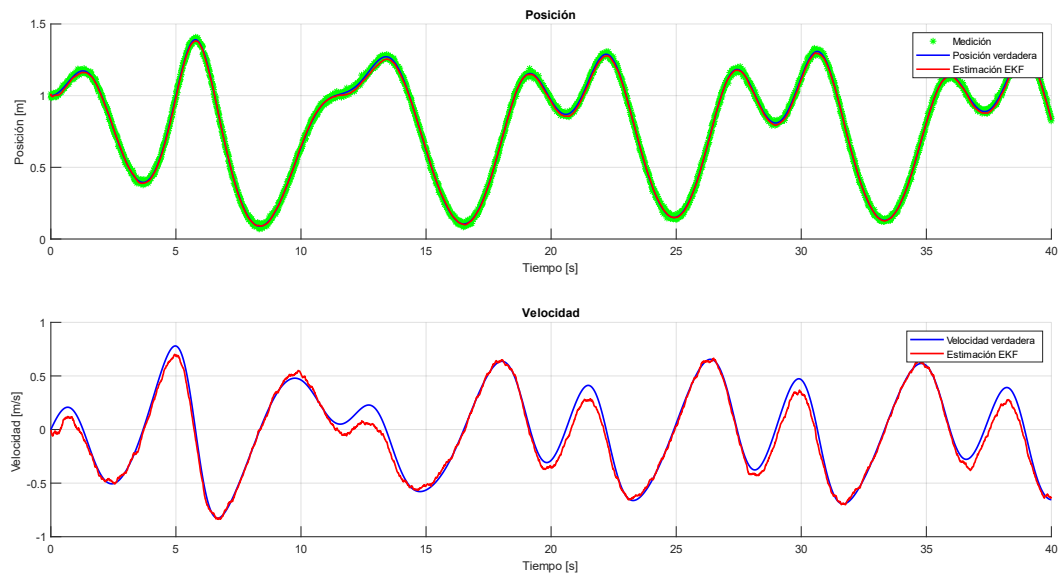


Figura 4.4: Resultados de EKF ante un sistema no lineal.

Se observa que las estimaciones de posición y velocidad se ajustan a los estados reales, con un RMSE para la posición de $0,0271 [m]$ y para la velocidad de $0,109 \left[\frac{m}{s} \right]$.

Con lo presentado en este capítulo queda demostrado que el algoritmo de Kalman funciona de forma eficiente para problemas lineales y con ruidos considerables, por otro lado, el programa realizado para el EKF se valida mediante una ecuación altamente no lineal, donde las estimaciones se ajustan a las curvas reales.

CAPITULO 5: Habilitación de la aeronave

En este capítulo se realizará una descripción de la aeronave a utilizar, se presentarán los componentes necesarios para la habilitación de la aeronave. Por último, se presentará la calibración de las veletas como también los diagrama de conexión entre los componentes.

5.1 Descripción de la aeronave KADET LT-40

El avión que se utilizara en este proyecto es el KADET LT-40 fabricado por la compañía SIG, este avión se caracteriza por ser utilizado en el aeromodelismo debido a su alta estabilidad y fácil maniobrabilidad. En la Figura 5.1 se presenta la aeronave.



Figura 5.1: Aeronave KADET LT-40.

La aeronave presentada tiene como superficie de control alerones que se encargan de realizar el roll, elevador encargado del movimiento de pitch y por último el timón que da la dirección al avión. A continuación, se presentan las características geométricas más importante de la aeronave. Lo presentado en este apartado se considera como el subsistema estructural de la aeronave.

5.1.1 Fuselaje

El fuselaje del avión en su mayoría está construido de madera balsa, material que tiene la características de tener una alta relación peso resistencia y un bajo costo económico. En la Tabla 5-1 se resumen las características más importantes del fuselaje.

Tabla 5-1: Características del fuselaje.

Característica	valor
Largo	1295 [mm]
Ancho (sección de payload)	85 [mm]
Alto (sección de payload)	150 [mm]
Volumen (sección de payload)	0,003825 [m^3]

Se da énfasis a las dimensiones geométricas de la sección payload, ya que el hardware a instalar posteriormente para los métodos de estimación y medición debe estar dentro de ese rango de valores.

En la Figura 5.2 se representa el fuselaje en una vista lateral y en planta (las medidas están en milímetros).

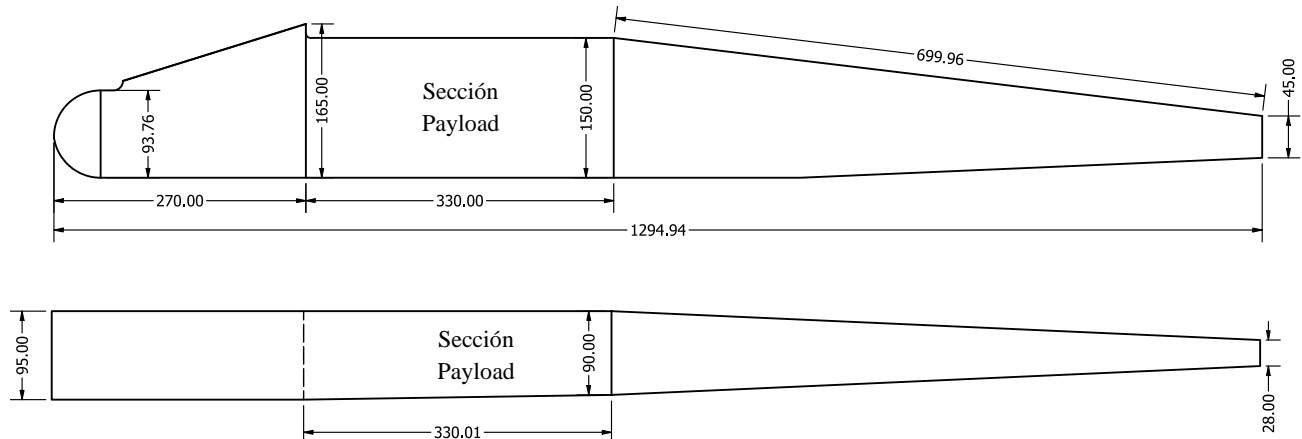


Figura 5.2: Planos de fuselaje.

5.1.2 Ala

Las características del ala se presentan en la Tabla 5-2, es importante mencionar que el ala a presentar tiene un perfil aerodinámico real que genera la sustentación requerida por el avión.

Tabla 5-2: Características del ala

Característica	Valor
Envergadura	1,77 [m]
Superficie Alar	0,58 [m^2]
Perfil aerodinámico	Clark-Y
Ángulo de diedro	4°
Razón de aspecto	5,4
Cuerda media aerodinámica	0,33 [m]
Posición del CG*	0,09 [m]
Carga alar ⁺	42 [$\frac{N}{m^2}$]
Superficie de alerones	0,0596 [m^2]

*Si bien el centro de gravedad no es una característica geométrica del ala, se proporciona este dato en la tabla ya que el CG se encuentra dentro de la cuerda media aerodinámica. En este caso se encuentra 0,09 metros desde el borde de ataque.

+Dato aproximado proporcionado por fabricante.

5.1.3 Empenaje

El empenaje de la aeronave es encargado de aportar estabilidad y control del avión en vuelo. Se compone del estabilizador vertical que se relaciona con la estabilidad lateral y el estabilizador horizontal relacionado con la estabilidad longitudinal. Las características del empenaje se presentan en la Tabla 5-3.

Tabla 5-3: Características geométricas del empenaje.

Estabilizador	Característica	Valor
Horizontal	envergadura	0,68 [m]
	Perfil	Placa delgada
	Superficie alar	0,1357 [m ²]
	Angulo de diedro	0°
	Razón de aspecto	3,4
	Superficie del elevador	0,033 [m ²]
Vertical	envergadura	0,214 [m]
	Perfil	Placa delgada
	Superficie alar	0,0486 [m ²]
	Angulo de diedro	0°
	Razón de aspecto	0,94
	Superficie del timón	0,01442 [m ²]

Es importante mencionar que la superficie del elevador y timón están dentro de las superficies de los estabilizadores.

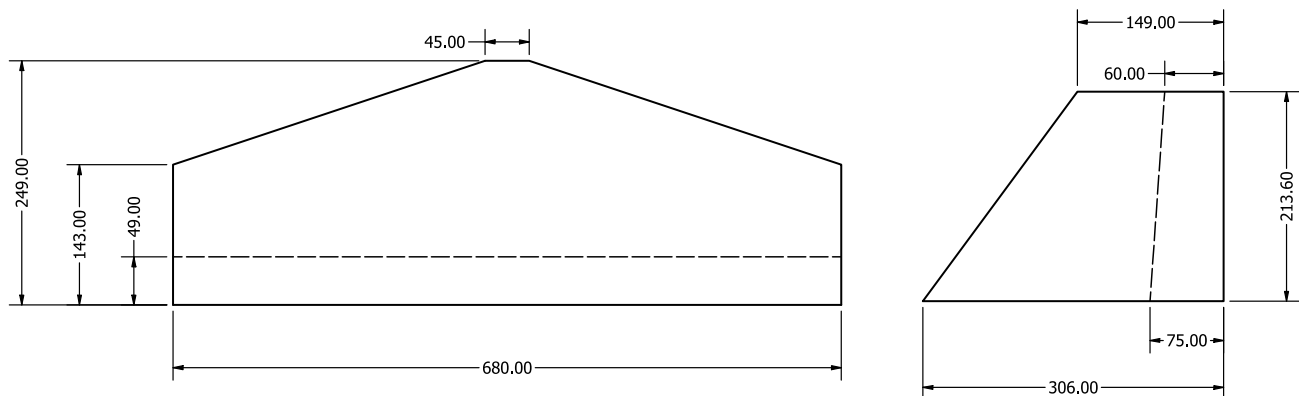


Figura 5.3: Estabilizador Horizontal y vertical.

5.2 Hardware primario

Se denominará Hardware primario a los componentes básicos que necesita la aeronave para realizar un vuelo controlado remotamente (modo en que se realizará el vuelo de prueba). Se presentará una breve descripción de cada componente dando énfasis en su función y características principales.

5.2.1 Subsistema de propulsión

Se conocerá como subsistema de propulsión al conjunto motor y hélice. El motor es el encargado de aportar la potencia necesaria para mover las hélices y así impulsar la aeronave a través del aire. Para la habilitación del KADET LT-40 se decide utilizar un motor eléctrico sin escobillas (brushless) modelo Turnigy SK4250-650 presentado en la Figura 5.4. Las características principales del motor se presentan en la siguiente tabla.

Tabla 5-4: Características del motor usado.

Característica	Valor
Constante de velocidad (kv)	650 $\left[\frac{rpm}{v}\right]$
Potencia	1150 [w]
Peso	230 [g]
Amperaje del variador recomendado	+60 [A]
Batería recomendada	Lipo 4s – 7s
Hélices recomendada	13x8"; 14x7"; 12x10"



Figura 5.4: Motor SK4250-650

La hélice es un elemento mecánico que es capaz de generar empuje debido a su forma aerodinámica. Se implementa una hélice de madera de diámetro 13 [in] y un paso de 8 [in], es decir teóricamente la hélice avanza 8 pulgadas por cada revolución. Se decide utilizar esta hélice debido a que es compatible con el motor anteriormente descrito y por restricciones de operación (que al despegar o aterrizar la hélice no choque con el piso).



Figura 5.5: Hélice seleccionada.

Cabe mencionar que el motor y la hélice presentada fue el resultado de un análisis de diferentes configuraciones (motor-hélice), a las cuales se le realizó la medición del empuje generado mediante una celda de carga. La configuración seleccionada alcanzó un empuje máximo de 29,5 [N] y fue la

que presento menos vibraciones. En Anexo A se presenta el sistema de medición de empuje y una tabla con los resultados de las diferentes configuraciones.

5.2.2 Controlador electrónico de velocidad (ESC)

El ESC actúa como intermediario entre la batería y el motor. Su función principal es regular la señal del motor mediante señales eléctricas temporizadas. Utiliza la corriente continua proveniente de la batería y la transforma internamente mediante una serie de transistores a corriente trifásica alterna. El ESC a utilizar se presenta en la Figura 5.6, el cual fue seleccionado por las condiciones que demanda el motor descrito anteriormente, en específico debido a que tiene la capacidad de soportar hasta 75 amperes máximos de trabajo y por su disponibilidad en el LTA. Además, incorpora internamente un circuito eliminador de batería (BEC) que suministra de forma continua 5 [V], destinados a alimentar diferentes componentes electrónicos.



Figura 5.6: Variador de velocidad

5.2.3 Batería

La batería es el dispositivo el cual almacena y proporciona la energía eléctrica durante la operación de la aeronave. Se decide usar baterías de polímeros de iones de litio recargables (Lipos) debido a su disponibilidad en el LTA, por su densidad energética y la capacidad de entregar una cantidad de energía considerable. Las baterías que se utilizarán en el avión KADET LT-40 son baterías de seis celdas. Cada celda tiene un voltaje nominal de 3,7 [V] y un voltaje máximo de 4,2 [V] lo que da como resultado un voltaje nominal total de 22,2 [V]. El peso de la batería es de 600 gramos.

5.2.4 Servomotor

El servomotor es un dispositivo electromecánico que tiene la capacidad de ajustarse a una posición angular específica dentro de un rango acotado. Un servomotor está formado por un motor de corriente continua, una caja reductora (engranajes) y un controlador. Estos dispositivos usan la modulación por ancho de pulsos (PWM) para su funcionamiento, así según el periodo del ancho de pulso el controlador del servo ajustará la posición angular de este.

Para la habilitación de la aeronave se utilizarán cuatro servomotores, con el fin de controlar las superficies de control. Se utiliza un servo para cada alerón de cada semi ala, para el elevador y el timón. Se usan servos Futaba FP-S148 los cuales admiten un voltaje máximo de 6 [V] y un torque máximo de 3 [kg – cm].



Figura 5.7: Servomotor FP-S148.

5.2.5 Receptor

El receptor RC tiene como función principal recibir las señales emitidas por el transmisor (control remoto) y convertirlas en diferentes señales eléctricas, las cual se utilizarán directamente para activar los servomotores y motor. En el caso de utilizar un controlador de vuelo, el receptor es el intermediario entre el radio control (manejado por el piloto) y el autopiloto. En la aeronave se utiliza el receptor FrSky X8R el cual funciona con PWM y consta de 8 canales de salida.



Figura 5.8: Receptor X8R

Con estos componentes mencionados el avión tiene un peso en vacío de 3590 [g].

5.2.6 Radio control

El radio control es el sistema que permite al piloto realizar las diferentes maniobras. Tiene como función principal transmitir la acción realizada por el piloto (diferentes movimientos de las palancas, pulsadores, etc) al receptor RC para así controlar la actitud del avión mediante las superficies de control. Se decide configurar la radio en modo de vuelo 1, es decir la palanca derecha tiene el control sobre el motor y los alerones, mientras que la palanca de la izquierda controla el elevador y timón. La radio que se utilizara es la RADIO-MASTER-TX12 debido a su disponibilidad en el laboratorio.



Figura 5.9: Radio Master TX12

Es importante mencionar que los modos de vuelos están estandarizados, por lo que al elegir el modo de vuelo 1, la conexión de los servomotores y motor al receptor tienen un canal específico.

Tabla 5-5: Canales del modo de vuelo.

Actuador	Canal
Servomotor de alerones	1
Servomotor del elevador	2
Motor	3
Servomotor del timón	4

5.3 Conexión de Hardware primario

En la Figura 5.10 se presenta un esquema de conexión del hardware primario, en el cual se identifican los pines principales de cada componente y la conexión entre ellos.

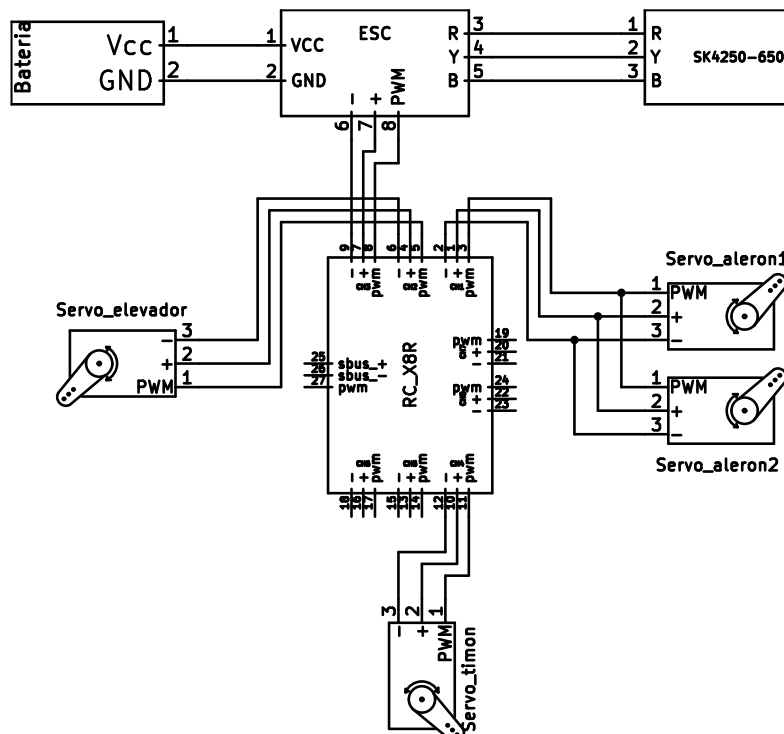


Figura 5.10: Diagrama de conexión

5.4 Veletas pivotantes

Como se mencionó en el Capítulo 1 se validarán los métodos de medición y estimación a diseñar asumiendo que el ángulo de ataque real es el que se mide con la veleta pivotante. Por lo que es fundamental realizar la calibración y habilitación de estas. La aeronave tiene instalada dos veletas pivotantes, una se encarga de medir el AOA demarcada en la Figura 5.11 con rojo y la otra el AOS demarcada en azul.

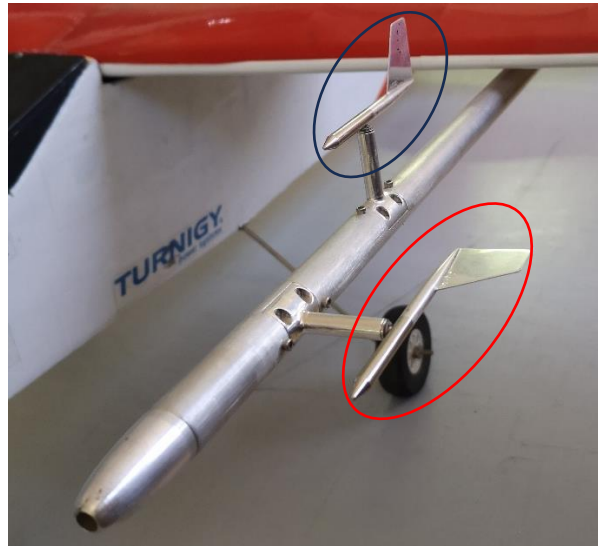


Figura 5.11: Veletas pivotantes.

Cada veleta cuenta con un potenciómetro incorporado, el cual cambia su resistencia eléctrica cuando estas sufren un cambio angular. Por esta razón, ambas veletas deben ser alimentadas con un voltaje continuo, con el fin de relacionar el cambio angular con la caída de voltaje en estas. Así el ángulo de ataque se obtiene con la relación entre el voltaje en el potenciómetro y el voltaje de alimentación. Sin embargo, como la batería de la aeronave se descarga en la operación, el voltaje de alimentación de las veletas varía, por lo que es necesario realizar la calibración en base a porcentajes y no valores específicos.

Para lograr realizar la calibración de ambas veletas se debe conocer el voltaje de alimentación y el ángulo de las veletas pivotantes. Se utiliza un motor paso a paso para poder medir con precisión la posición angular de las veletas, esto ya que este tipo de motores tienen la capacidad de realizar movimientos angulares discretos y conocidos (el motor utilizado tiene 200 pasos, donde cada paso equivale a $1,8^\circ$). Para realizar las mediciones se utiliza un microcontrolador Arduino, con el cual se logra la medición del voltaje de alimentación, voltaje de las veletas y posición angular de las veletas. La metodología para realizar esta calibración fue alimentar las veletas con ciertos voltajes en específico, donde en cada voltaje se realiza un barrido de la posición angular de las veletas, midiendo en cada posición la caída de voltaje en estas. Es importante mencionar que para efectuar la calibración se fija el ala, con el fin de minimizar errores.

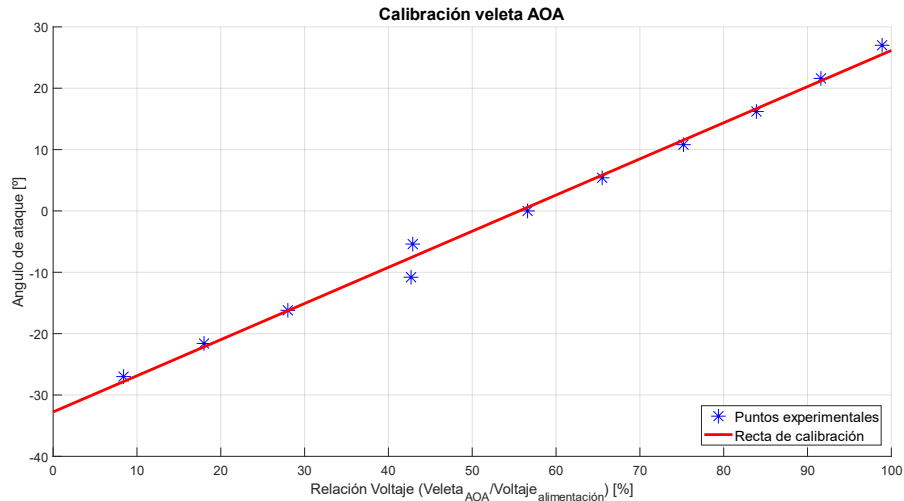


Figura 5.12: Resultado de calibración de la veleta AOA.

En la Figura 5.12 los puntos azules son las mediciones que se realizaron para la calibración, y la recta roja es el resultado de la calibración, la cual se obtiene realizando una recta de regresión lineal en las mediciones. La ecuación de la línea de tendencia es la que se presenta a continuación, la cual tiene un coeficiente de determinación de 0,994.

$$AOA = 0,589 \cdot \% \frac{V_{AOA}}{V_{CC}} - 32,76 \quad (5.1)$$

Con respecto a la calibración de la veleta que medirá el AOS se realiza el mismo procedimiento que en la veleta anterior. Se obtiene la ecuación lineal (5.2) la cual tiene un coeficiente de determinación de 0,984. En la Figura 5.13 se presenta el resultado de la calibración junto con las mediciones experimentales.

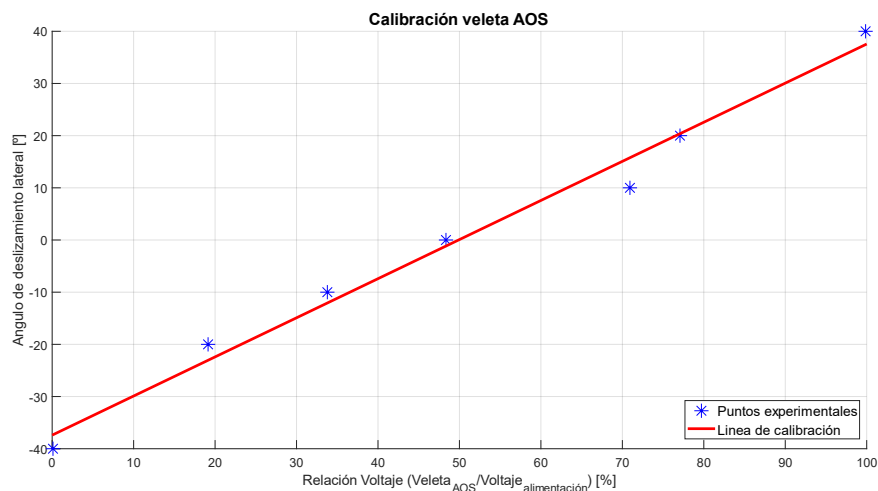


Figura 5.13: Resultado de calibración de la veleta AOS.

$$AOS = 0,749 \cdot \% \frac{V_{AOS}}{V_{CC}} - 37,41 \quad (5.2)$$

CAPITULO 6: Diseño de método de estimación y medición

6.1 Método de medición

6.1.1 Método de calibración y sonda diseñada

Para realizar la medición del ángulo de ataque se decide usar la técnica recomendada por [17], la cual se basa en buscar una correlación entre una diferencia de presión con el AOA. Para lograr encontrar dicha curva se medirá la presión en dos puntos diferentes, el primero se denominará P_{long} y se encargará en tomar la presión en la dirección longitudinal de la aeronave, mientras que el segundo P_{45} tomará la presión en un plano a 45° respecto al eje longitudinal. Sin embargo, la diferencia de presión dependerá de la velocidad de avance de la aeronave, debido a la presión dinámica.

$$P_{total} = \frac{1}{2}\rho V^2 + P_{atmosferica} \quad (6.1)$$

Si se realiza la diferencia de presión en los dos puntos utilizando la ecuación anterior, se obtiene lo siguiente,

$$\Delta P = \frac{1}{2}\rho(V_{long}^2 - V_{45}^2) \quad (6.2)$$

Es claro que ΔP depende de la velocidad, lo que implica que para cada valor de velocidad existirá una curva que correlacione el AOA con la diferencia de presión. Para evitar esto, se realiza una normalización, es decir ΔP se divide por la presión en el punto P_{45} con el fin de que independientemente de la velocidad la curva que relacione el AOA con ΔP sea única. En [17] se sugiere obtener la relación mediante la división de presiones, sin embargo, se llegan a resultados similares.

$$\frac{\Delta P}{P_{45}} = \frac{P_{long} - P_{45}}{P_{45}} \quad (6.3)$$

Dicho lo anterior se utilizará la relación (6.3) para realizar la correlación con el AOA. Es importante mencionar que para encontrar la ecuación que finalmente describa el AOA en función de (6.3) es necesario realizar una regresión polinomial (calibración), donde se necesitaran los datos de presión longitudinal, presión a 45° y de la veleta pivotante que se asume que es el AOA real. Esta calibración se presenta en capítulos posteriores.

La sonda diseñada se presenta en la Figura 6.1, la cual se realizó tomando en consideración la implementación en la aeronave. Dicha sonda se compone solo de una pieza, la cual permite la medición de la presión longitudinal y la presión a 45° grados, la fabricación se realiza mediante proceso de manufactura aditiva utilizando impresora 3D, se utilizó filamento de Ácido Poliláctico

(PLA) para la impresión en 3D, debido a que la sonda no está sometida a cargas de magnitud considerable y por su disponibilidad

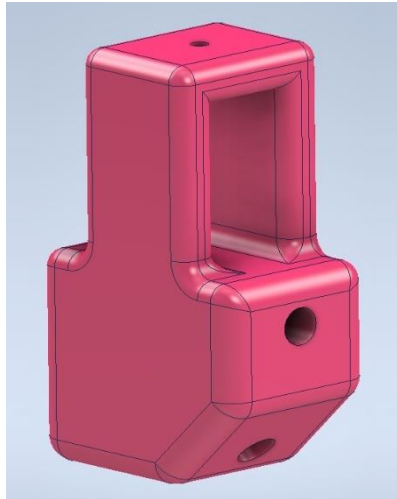


Figura 6.1: Sonda diseñada.

La sonda tiene un peso de 16 gramos, esta se decide instalar bajo el semi ala derecha, debido a que en la semi ala izquierda está montado el tubo de Pitot junto con las veletas pivotantes, mecanismos que afectarían la toma de presión, además con la finalidad de equilibrar el peso de la aeronave con respecto al eje longitudinal. Para lograr la toma de presión se deben insertar mangueras dentro de los orificios para conectarlas con los sensores que están dentro del fuselaje. La instalación en la aeronave se presenta en la siguiente figura.

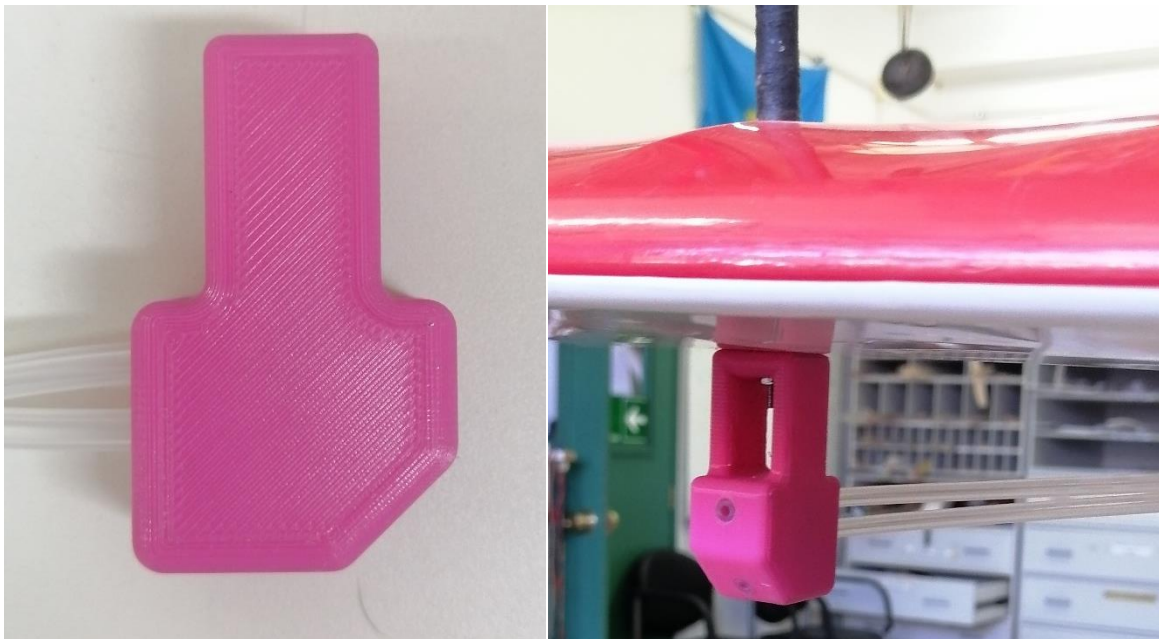


Figura 6.2: Sonda fabricada e instalada en la aeronave.

6.1.2 Rango de trabajo y precisión

En [17] analizan la precisión que debe tener un sensor para lograr detectar la entrada en pérdida de un aeronave. se destaca que la mayoría de los aviones entran en pérdida a los 20-22° grados de ángulo de ataque, donde pasar de una condición estable a una condición crítica tiene el orden de un grado. Además, se hace mención que para operar a una máxima eficiencia aerodinámica el ángulo de ataque debe mantenerse dentro de un rango de $\pm 1^\circ$. Como el rango es de 1° la resolución de la medición debe ser menor, con el fin de lograr detectar previamente la entrada en pérdida, es por esto que se impone que la precisión de la medición sea de $0,25^\circ$. Con respecto al rango de trabajo, se asume que la aeronave Kadet LT-40 tiene una velocidad mínima de 5 [m/s], lo que implica una presión dinámica mínima de 15,3 [Pa], además aplicando la ecuación proporcionada por [17] que relaciona el cambio de AOA con la diferencia de presión normalizada por la presión dinámica, se obtiene que si el cambio de AOA es de $0,25^\circ$ la diferencia de presión es de 4,42 [Pa]. Por otro lado, si se asume un vuelo desfavorable donde la aeronave avanza a una velocidad de 5 [m/s] y con un AOA de 15° , se llega a la conclusión que el sensor de presión diferencial P_{45} medirá una presión de 7,05 [Pa]. Juntando ambas condiciones, la diferencia de presión es la que condiciona al sensor a seleccionar, así el sensor de presión diferencial debe tener una resolución menor a 4,42 [Pa].

6.1.3 Hardware

Sensores: Los sensores de presión diferencial a utilizar son los Ms4525do, estos sensores son fabricados a base de materiales cerámicos. Son sensores digitales, que tienen la capacidad de medir tanto la diferencia de presión con una resolución digital de 14 bits como la temperatura con una resolución digital de 11 bits. Las características específicas de este sensor se presentan en la siguiente tabla.

Tabla 6-1: características del sensor Ms4525do

Parámetro	Valor
Voltaje de alimentación	3,3-5.0 [V]
Rango de presión	[0-6894] [Pa]
Rango de temperatura	[-40 125] [°C]
Corriente máxima de salida	3 [mA]
Resolución	0,84 [Pa]
Comunicación	I2C-SPI

Estos sensores cuentan con dos puertos destinados a la medición de la presión. El puerto número 1 se utiliza comúnmente para medir la presión dinámica, mientras que el puerto número 2 se emplea para medir la presión estática. Por esta razón, se ha optado por utilizar un sensor para la toma de presión longitudinal y otro para la toma de presión a 45 grados. Además, es importante destacar que la elección de estos sensores se basó en su bajo costo y disponibilidad en el mercado.

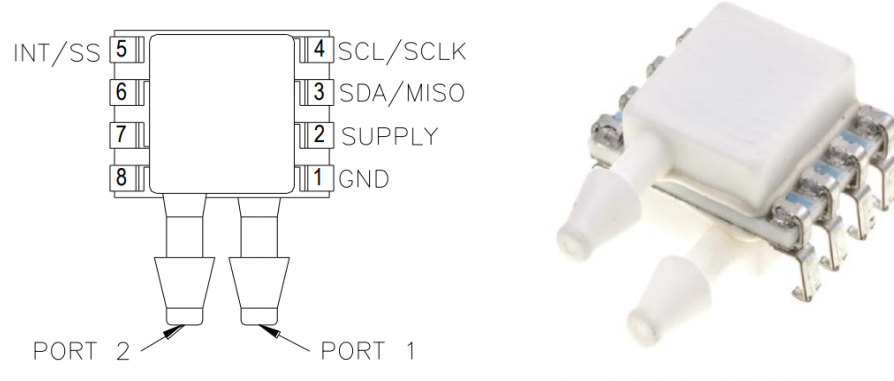


Figura 6.3: Sensores Ms4525do

Arduino: Se optó por emplear un microcontrolador Arduino Uno debido a su capacidad para establecer comunicación con los sensores de presión diferencial a través del protocolo I2C. Además, permite la programación del software necesario para obtener la curva que relacionará la diferencia de presión con el Ángulo de Ataque (AOA). Es relevante señalar que, al tratarse de sensores del mismo fabricante y modelo, comparten una identidad común, lo que implica la necesidad de habilitar dos juegos de pines diferentes para lograr la comunicación simultánea mediante el protocolo I2C (2 SDA, 2 SCL). Por otro lado, este microcontrolador se utilizará para leer la caída de voltaje de las veletas pivotantes mencionadas en el Capítulo 5.



Figura 6.4: Arduino UNO R3.

6.1.4 Conexión método de medición

En la Figura 6.5 se presenta un esquema de conexión entre los componentes para el método de medición.

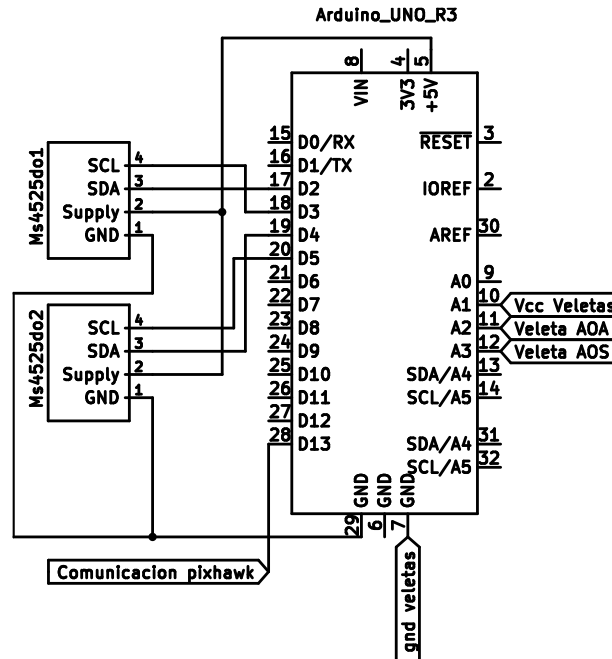


Figura 6.5: Conexión método de medición.

Se ha configurado la conexión del pin digital D13 con el controlador de vuelo Pixhawk. Esta elección se basa en la limitada capacidad de almacenamiento del Arduino, lo que hace necesario externalizar la retención de datos. En consecuencia, se ha optado por establecer una comunicación con el Pixhawk para almacenar en su tarjeta SD la información correspondiente a la presión y los voltajes de las veletas. Además de superar la limitación de almacenamiento, esta configuración posibilita la recopilación de datos tanto del método de estimación como del método de medición con una frecuencia idéntica, simplificando el procesamiento de esta información. Se explicará en más detalles en capítulos posteriores.

6.2 Método de estimación

6.2.1 Modelación del problema

Para la estimación del ángulo de ataque se propone la fusión de un algoritmo de Kalman extendido y un método algebraico, basado en la medición del sistema inercial. Los resultados del método algebraico entran como una pseudomedición al EKF. La formulación del problema se basa en las ecuaciones (3.15) que describen la dinámica de una aeronave.

Como primer paso se calcula el ángulo de ataque y el ángulo de deslizamiento lateral de forma simultánea con el método de Newton Raphson utilizando las siguientes ecuaciones rescatadas de [19],

$$\begin{aligned} \dot{\alpha} &= \frac{1}{V \cos(\beta)} \left(-\frac{F_x}{m} \sin(\alpha) + \frac{F_z}{m} \cos(\alpha) \right) + q - (p \cos(\alpha) + r \sin(\alpha)) \tan(\beta) \\ \dot{\beta} &= \frac{1}{V} \left(-\frac{F_x}{m} \cos(\alpha) \sin(\beta) + \frac{F_y}{m} \cos(\beta) - \frac{F_z}{m} \sin(\alpha) \sin(\beta) \right) + p \sin(\alpha) - r \cos(\alpha) \end{aligned} \quad (6.4)$$

Estas son el resultado de derivar las ecuaciones del AOA, AOS y posteriormente sustituirlas en las ecuaciones de velocidad en el cuerpo. Para obtener el valor de los ángulos deseados mediante el

método algebraico, se realiza una simplificación que asume las siguientes igualdades: $\dot{\alpha} \approx q$ y $\dot{\beta} \approx -r \cos(\alpha)$ resultando las siguientes ecuaciones.

$$\begin{aligned} \frac{1}{V \cos(\beta)} \left(-\frac{F_x}{m} \sin(\alpha) + \frac{F_z}{m} \cos(\alpha) \right) - (p \cos(\alpha) + r \sin(\alpha)) \tan(\beta) &= 0 \\ \frac{1}{V} \left(-\frac{F_x}{m} \cos(\alpha) \sin(\beta) + \frac{F_y}{m} \cos(\beta) - \frac{F_z}{m} \sin(\alpha) \sin(\beta) \right) + p \sin(\alpha) &= 0 \end{aligned} \quad (6.5)$$

Para capturar la parte de la dinámica rotacional que se eliminó anteriormente, las tasas angulares ponderadas se suman con la solución de Newton Raphson, obteniendo el resultado final del AOA y AOS por el método algebraico.

$$\begin{aligned} \alpha_{pseudo} &= \alpha_{NR} + (TC_1 \cdot \Delta t \cdot q) \\ \beta_{pseudo} &= \beta_{NR} + (TC_2 \cdot \Delta t \cdot r \cdot \cos(\alpha_{pseudo})) \end{aligned} \quad (6.6)$$

En la ecuación anterior los valores de TC_1, TC_2 son constantes que se obtienen experimentalmente por [18] y Δt es el paso de tiempo. El resultado de AOA que se obtienen con la ecuación (6.6) se utilizará en el paso de corrección del filtro de Kalman extendido, con esto se logra simular que se cuenta con un sensor para la medición del AOA, lo que contribuirá a la convergencia en el algoritmo. Es decir, el α_{pseudo} estará en el vector de medición.

El segundo paso para la estimación del AOA es aplicar el filtro de Kalman extendido con el objetivo de estimar las velocidades lineales en el marco del cuerpo, para luego remplazarlas en las ecuaciones de AOA dictada por literatura y así obtener la estimación de dicho ángulo.

Las ecuaciones de propagación para este algoritmo son derivadas de las ecuaciones de traslación de la aeronave presentadas en el Capítulo 3.1.3. Por otra parte, las ecuaciones de corrección se obtienen a partir de la relación entre la velocidad aerodinámica, velocidad inercial y ecuación del AOA con el vector estados, es decir con las velocidades lineales en el cuerpo.

$$\begin{aligned} f(x, u) &= \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{Bmatrix} -qw + rv - g \sin(\theta) + a_x \\ -ru + pw + g \sin(\phi) \cos(\theta) + a_y \\ -pv + qu + g \cos(\phi) \cos(\theta) + a_z \end{Bmatrix} \\ h(x_k) &= \begin{bmatrix} V \\ v_x \\ v_y \\ v_z \\ \alpha_{pseudo} \end{bmatrix} = \begin{Bmatrix} \sqrt{u^2 + v^2 + w^2} \\ B_1 \\ B_2 \\ B_3 \\ \tan^{-1}\left(\frac{w}{u}\right) \end{Bmatrix} \end{aligned} \quad (6.7)$$

En (6.7) se puede observar que tanto las ecuaciones de propagación y de corrección son no lineales. Los términos B_1, B_2 y B_3 son el resultado de multiplicar la transpuesta de la matriz R por el vector de velocidad lineal en el cuerpo de la aeronave. La expansión de dichos términos se presenta a continuación,

$$\begin{aligned}
B_1 &= [(c\theta \cdot c\psi) \cdot u + (s\phi \cdot s\theta \cdot c\psi - c\phi \cdot s\psi) \cdot v + (c\phi \cdot s\theta \cdot c\psi + s\phi \cdot s\psi) \cdot w] \\
B_2 &= [(c\theta \cdot s\psi) \cdot u + (s\phi \cdot s\theta \cdot s\psi + c\phi \cdot c\psi) \cdot v + (c\phi \cdot s\theta \cdot s\psi - s\phi \cdot c\psi) \cdot w] \\
B_3 &= [-s\theta \cdot u + (c\theta \cdot s\phi) \cdot v + (c\phi \cdot c\theta) \cdot w]
\end{aligned} \tag{6.8}$$

Dado que las funciones de estados y de salida (6.7) son no lineales, se requiere derivarlas con respecto al vector de estado con la finalidad de obtener las matrices de transición de estados y observación, para luego aplicar el Filtro de Kalman Extendido (EKF). Es importante mencionar que la derivación de estas ecuaciones es propia para cada paso de tiempo, esto debido a que se realiza alrededor del estado resultante del paso de predicción. El resultado de la derivación en forma general se presenta a continuación.

$$\begin{aligned}
\frac{df(x, u)}{dx} = F &= \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \\
\frac{dh(x_k)}{dx} H &= \begin{bmatrix} \frac{u}{\sqrt{u^2 + v^2 + w^2}} & \frac{v}{\sqrt{u^2 + v^2 + w^2}} & \frac{w}{\sqrt{u^2 + v^2 + w^2}} \\ -s\phi c\theta s\psi + c\phi c\psi & -s\phi c\theta c\psi - c\phi s\psi & s\psi s\theta \\ c\phi s\psi c\theta + s\phi c\psi & c\phi c\theta c\psi - s\phi s\psi & -c\phi s\theta \\ s\theta s\psi & s\theta c\psi & c\theta \\ -w & 0 & 1 \\ u^2 \left(\frac{1 + w^2}{u^2} \right) & & u \left(\frac{1 + w^2}{u^2} \right) \end{bmatrix}
\end{aligned} \tag{6.9}$$

Luego, al aplicar el filtro de Kalman extendido, se tiene como resultado las estimaciones de las velocidades en el cuerpo, estas se remplazan en la ecuación (6.10) para obtener el ángulo de ataque.

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \tag{6.10}$$

Con esto se tiene que el vector de estado, vector de entradas y vector de salida para el EKF son los siguientes.

$$\begin{aligned}
x &= [u \ v \ w] \\
u &= [p \ q \ r \ \theta \ \phi \ \psi \ a_x \ a_y \ a_z] \\
y &= [V \ \alpha_{pseudo} \ v_x \ v_y \ v_z]
\end{aligned}$$

6.2.2 Hardware necesario

Para obtener el AOA mediante el método de estimación es necesario medir las variables que componen al vector de entrada de salida. En este contexto, se opta por utilizar un controlador de vuelo modelo Pixhawk 6c presentado en la Figura 6.6, debido su capacidad de adquisición de datos mediante unidades de medida inercial (IMU) y diferentes sensores, además se selecciona por la compatibilidad con diferentes softwares. Es importante destacar que el controlador de vuelo tiene la capacidad de realizar diferentes misiones, como por ejemplo un vuelo autónomo, sin embargo, para este proyecto el autopiloto se utilizará netamente para la adquisición de datos en lazo abierto, es decir el avión será controlado remotamente como se explicó en el capítulo anterior.



Figura 6.6: Pixhawk 6c

Como se puede observar en la figura anterior, el pixhawk presenta diversas entradas y salidas, cada una con una función específica. Sin embargo, solo algunas se utilizan para lograr realizar un vuelo remotamente y adquirir los datos deseados. A continuación, se detalla brevemente las interfaces a utilizar que son obligatorias para un vuelo exitoso.

- **Power:** Esta entrada es por donde se debe alimentar el controlador de vuelo, con un voltaje máximo de 6 Volt. Trae dos entradas de alimentación por temas de redundancia, con el fin de prevenir fallas de alimentación en vuelo.
- **FMU PWM OUT:** A estos pines se conectan los diferentes actuadores necesarios en la aeronave, que funcionan con modulación ancho de pulso como los son los servomotores.
- **GPS:** En este Puerto va conectado un módulo GNSS que trae integrado una brújula, interruptores de seguridad y zumbadores. Este módulo es obligatorio debido a que el Firmware que se instala al controlador, requiere datos de la brújula para determinar la posición de la aeronave como también calcular la actitud de esta. Además, con los datos de este módulo es posible obtener las velocidades en el marco de referencia inercial.
- **BUS RC:** En estos pines se debe conectar el receptor RC que será el intermediario entre el piloto y el controlador de vuelo.
- **TELEM:** Los puertos de telemetría son utilizados principalmente para lograr una comunicación entre el controlador de vuelo y una estación terrestre, además pueden ser utilizados para recibir datos de sensores o microcontroladores que estén a bordo de la aeronave.
- **I2C:** Este puerto se utiliza para la comunicación con diferentes sensores digitales que utilizan el protocolo I2C, en este caso se utilizara para realizar la conexión con un sensor diferencial de presión modelo Ms4525do, el cual es utilizado internamente en el controlador para calcular la velocidad aerodinámica.

Como se mencionó anteriormente el controlador de vuelo necesita ser alimentado con un voltaje máximo de 6 Volt, por lo que se necesita un módulo de potencia que se encargue de reducir el voltaje de alimentación de la batería Lipo a un voltaje adecuado y estable para la alimentación del Pixhawk. El módulo seleccionado fue el modelo PM07 presentado en la Figura 6.7 este módulo además de alimentar al controlador tiene diferentes funciones. La primera es ser un intermediario entre el controlador de vuelo, los servomotores y GPS para facilitar su conexión física. La segunda es tener

puertos específicos para conectar el variador de velocidad, junto con su señal para alimentar el motor. (se pueden conectar hasta 8 motores).

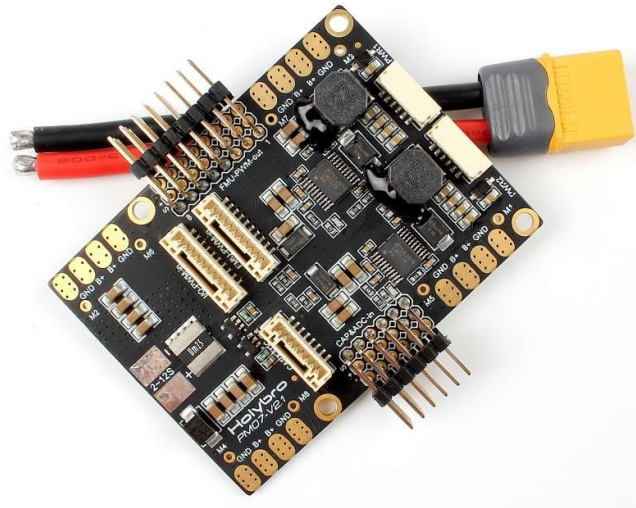


Figura 6.7: Módulo PM07.

La conexión entre los componentes para lograr realizar un vuelo y obtener mediciones desde el controlador de vuelo pixhawk se presenta en la siguiente figura.

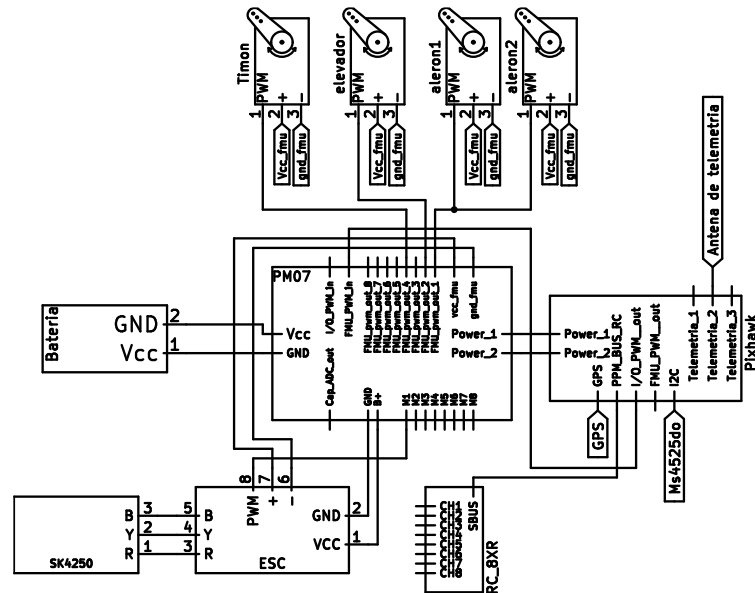


Figura 6.8: Esquema de conexión con Pixhawk y el módulo de potencia.

6.2.3 Software

Para utilizar el controlador de vuelo pixhawk 6c es necesario un software encargado de cargar el firmware apropiado. En este caso se utilizará Mission planner, el cual es una aplicación de código abierto que utiliza Ardupilot como base. Este software se usará por cuatro razones fundamentales: 1) Para cargar el firmware de una aeronave de ala fija, el cual se encargará de ejecutar los diferentes algoritmos que modelan la dinámica de la aeronave, logrando el control de la actitud en caso de ser

necesario, como la estimación de diferentes parámetros claves. 2) Ser la estación terrestre del RPAs, con la finalidad de realizar la comunicación con la aeronave, seleccionando los diferentes parámetros y armado final para la operación. 3) Realizar la calibración de los giroscopios, acelerómetros y brújula. 4) Tener acceso a los datos de vuelo, tanto de las mediciones de la IMU como los datos que internamente estima el firmware cargado al auto piloto.

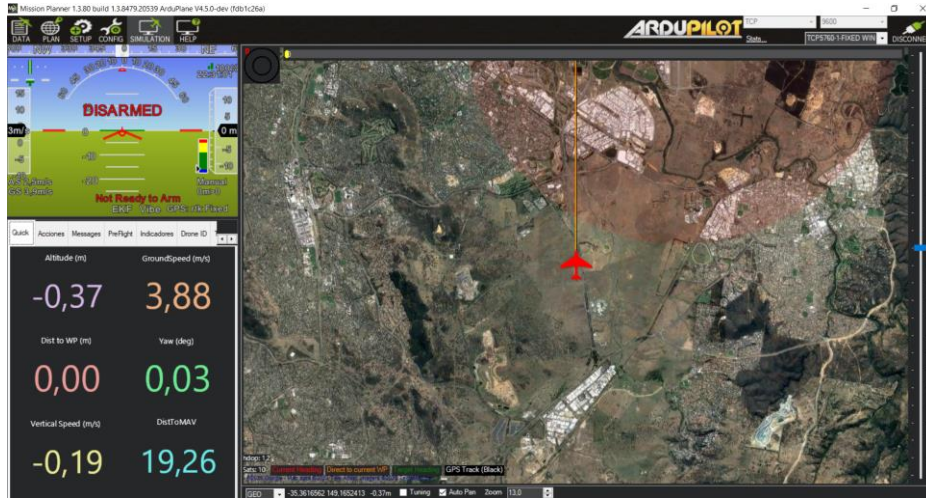


Figura 6.9: Entorno del software Mission Planner.

6.3 Conexión final

La conexión final entre todos los componentes para la implementación del método de medición y estimación en la aeronave se presenta en la Figura 6.10. Como se mencionó en un capítulo anterior es necesario realizar una conexión para comunicación entre un PIN digital del Arduino y una entrada de telemetría del controlador de vuelo, con el fin de guardar los datos de presión correspondiente al método de medición y los voltajes de las veletas. Estos valores se guardarán en la tarjeta de memoria del pixhawk junto con las mediciones de la IMU con una misma frecuencia de muestreo. Para lograr esto se aprovecha la capacidad que tiene Ardupilot de soportar secuencias de comandos de lenguaje LUA (lenguaje de programación), esto permite agregar nuevos comportamientos básicos al piloto automático sin modificar el firmware de vuelo principal. El código que se realizó guarda todas las variables necesarias en la tarjeta SD con una frecuencia promedio de 10 datos por segundo, la que varía según la capacidad del controlador de vuelo en el instante. El código se presenta en anexos.

6.4 Presupuesto de la implementación

Con todo el hardware presentando anteriormente es posible realizar el propuesto necesario para la implementación del método de medición y estimación del AOA en la aeronave KADET LT-40. Se hará la distinción entre los componentes primarios con los componentes asociados a los métodos diseñados. Además, se considera el servicio del ingeniero que realiza el proyecto, para este caso se considera un ingeniero junior que cobra por su servicio 0,5UF/hh (\$18.423) y se contempla que cada día de trabajo es de 8 horas.

Tabla 6-2: Presupuesto.

	Item	Precio unitario (CLP)	Cantidad	Envió (CLP)	Total
Componentes Para métodos diseñados	Pixhawk 6c/ PM07	\$310.000	1	\$21.280	\$331.280
	Ms4525do	\$29.247	3	\$2.600	\$90.341
	RC 8xr	\$24.275	1	\$1.350	\$25.625
	Arduino Uno	\$2.425	1	\$1.104	\$3.529
	Antenas de telemetría	\$57.000	1	\$2.400	\$59.000
	Subtotal				
Componentes primarios	Motor	\$43.000	1	-	\$43.000
	Hélice	\$2.830	1	-	\$2.830
	ESC	\$104.840	1	-	\$104.840
	Servomotor	\$15.720	4	-	\$62.880
	KADET LT-40	\$292.500	1	-	\$292.500
	Subtotal				
Servicios de ingeniero	Actividad		Días requeridos	Precio por día	Total
	Diseño de Hardware		2	\$147.388	\$294.776
	Diseño de Software y Programación		3	\$147.388	\$442.014
	Habilitación de la aeronave e implementación de Hardware		6	\$147.388	\$884.028
	Habilitación de sensores y controlador de vuelo (calibración, etc.)		1	\$147.388	\$147.388
	Prueba de vuelo		1	\$147.388	\$147.388
	Mediciones		1	\$147.388	\$147.388
	Procesamiento de datos		2	\$147.388	\$294.776
Subtotal					\$2.358.208
Total					\$3.368.033

Capítulo 7: Vuelos “Adquisición y procesamientos de datos”

7.1 Metodología para realizar los vuelos

Para validar los métodos presentados en el capítulo anterior se realizaron una serie de vuelos con el avión KADET LT-40 habilitado anteriormente. Los vuelos se realizan en el club de aeromodelismo de Concepción, en los cuales se guardan todos los datos necesarios para el método de medición y estimación. Se realizan 2 salidas a terreno, en las cuales en cada una se efectúan 3 vuelos diferentes.

Para obtener vuelos exitosos se sigue los pasos de una lista de pre-vuelo presentada en Anexo E, en la cual se verifica que todos los componentes de la aeronave estén en buenas condiciones, como por ejemplo la correcta conexión de los elementos electrónicos (Autopiloto, Servos, receptor, sensores, etc), las fijaciones estructurales, la correcta posición de la hélice. Además, se verifica la comunicación entre la estación terrestre con la aeronave y la comunicación entre el radio control y receptor.

Cabe destacar que al finalizar cada vuelo se debe cambiar la batería para evitar accidentes en la operación, además se descargan y eliminan los datos guardados en la tarjeta de memoria debido a su limitado espacio de almacenamiento y para tener un orden en los datos.

7.2 Vuelos realizados

Para los vuelos realizados el 10 de enero del 2024 se destaca que las condiciones climáticas en el sector de los vuelos fueron idóneas con un viento aproximado de 13 kilómetros por hora. Se logró obtener la medición de todos los datos necesarios solo en el vuelo N°2, debido a que en el resto de los vuelos hubo un error con la medición del voltaje correspondiente a las veletas pivotantes, causado por la desconexión de los conectores en vuelo. Para lograr realizar una buena toma de datos en la siguiente salida a terreno se decide cambiar el conector de las veletas por uno nuevo, el que cumple con las medidas necesarias para evitar la desconexión al momento de la operación.

Con respecto a las mediciones realizadas el día 17 de enero del 2024, hubo una falla del sensor de presión diferencial encargado de medir la presión a 45° respecto a la horizontal, por lo que esto limita la aplicación del método de medición. El resto de los datos fueron medidos sin problemas. Las condiciones meteorológicas no fueron las óptimas debido a que en el sector de los vuelos había un viento considerable de aproximadamente 30 kilómetros por hora.

7.2.1 Datos medidos en vuelo N°2 del día 10-01-2024.

Se presentan solo los datos de este vuelo por tres razones principales, la primera es debido a la calidad de las mediciones por las condiciones meteorológicas presentadas en el día del vuelo, la segunda razón es debido a que fue el único vuelo donde se guardan todos los datos necesarios para la aplicación de los métodos diseñados, por lo que se utilizarán para la calibración del método de medición. Por último, para no saturar el presente capítulo con datos de vuelo. Los datos medidos en los otros vuelos se presentan en anexos. Es importante mencionar que a cada variable medida se le aplica un filtro de paso bajo, con la finalidad de eliminar el ruido de alta frecuencia el cual no representa la realidad de la actitud de la aeronave y conlleva a resultados erróneos.

Los datos que se observan en las siguientes figuras se presentan en el siguiente orden: Aceleraciones en el cuerpo, actitud de la aeronave, velocidad angular, velocidad inercial, velocidad de avance y por último el voltaje en las veletas pivotantes.

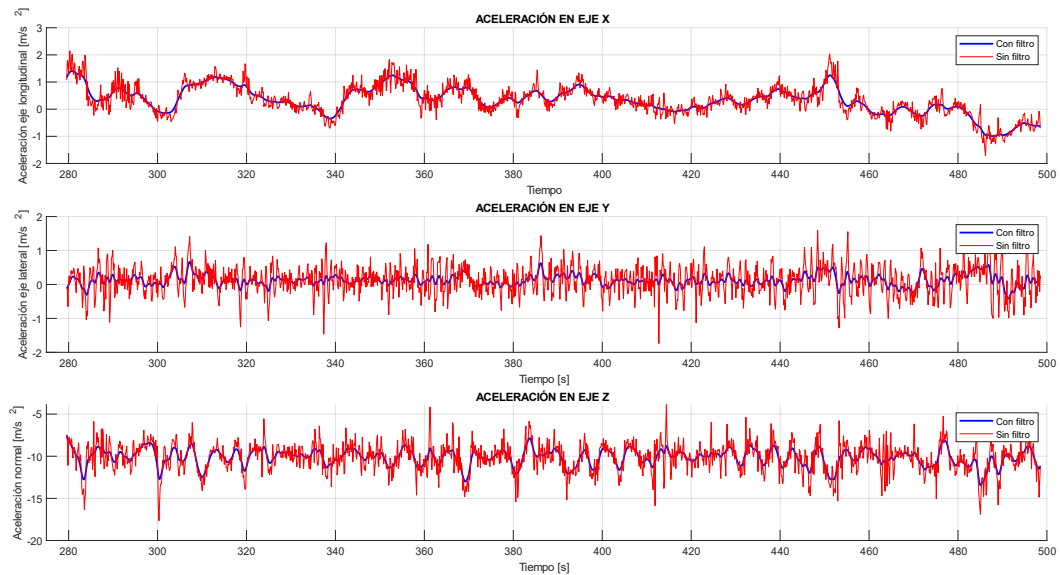


Figura 7.1: Aceleraciones en vuelo N°2.

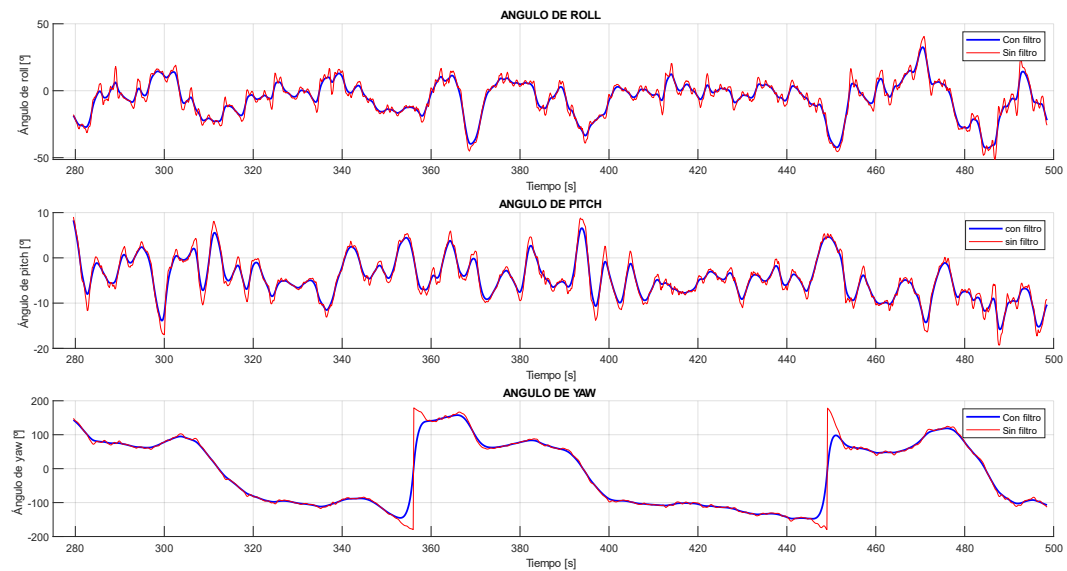


Figura 7.2: Actitud de la aeronave en vuelo N°2.

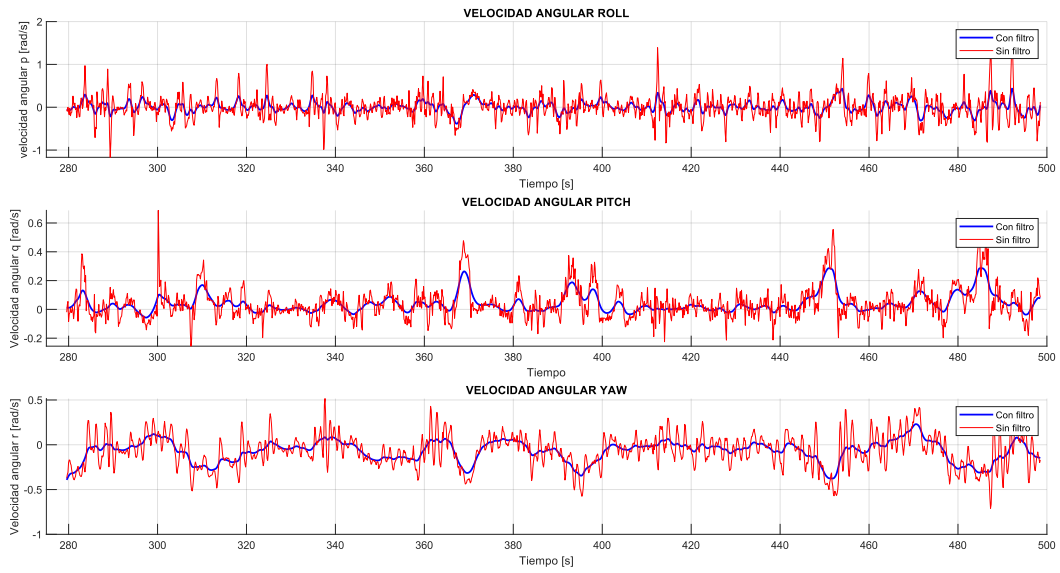


Figura 7.3: Velocidad angular en vuelo N°2.

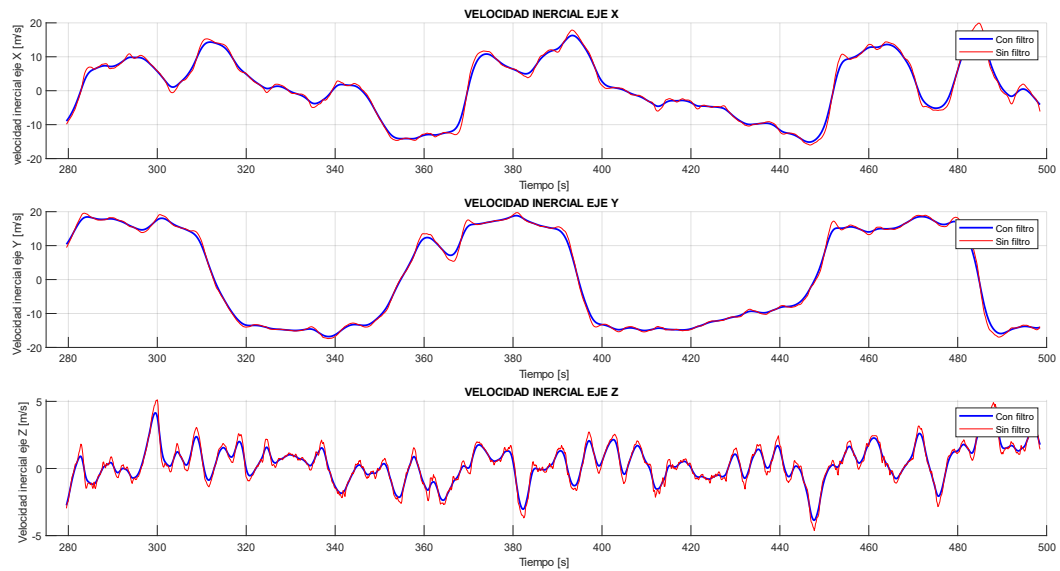


Figura 7.4: Velocidad inercial en vuelo N°2.

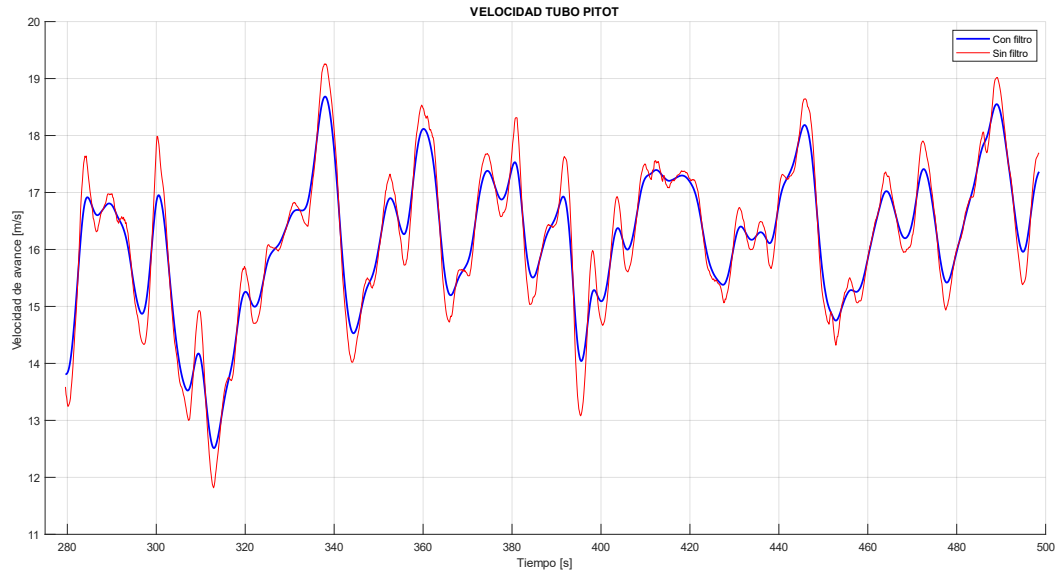


Figura 7.5: Velocidad aerodinámica del vuelo N°2.

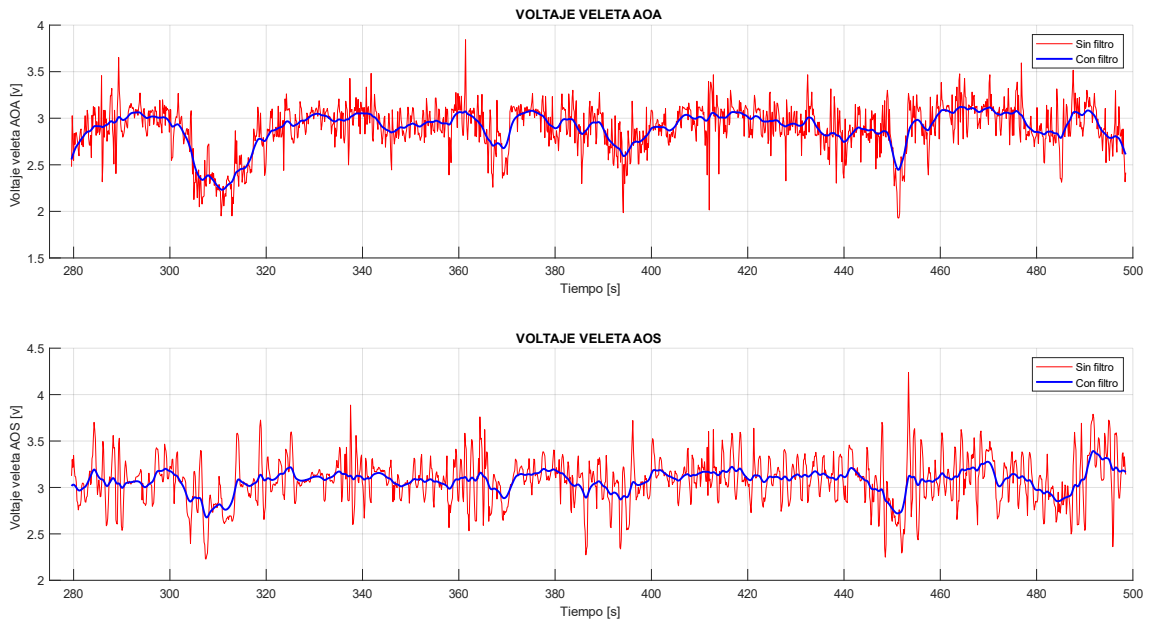


Figura 7.6: Voltajes en veletas pivotantes del vuelo N°2.

Capítulo 8: Resultados de métodos diseñados y análisis de resultados.

8.1 Método de medición

8.1.1 Calibración de la sonda

Como se mencionó en los capítulos anteriores es necesario calibrar la sonda diseñada utilizando datos medidos de algún vuelo. Para realizar la calibración se decide utilizar los datos medidos del vuelo N°2 del día 10 de enero del 2024.

La finalidad de esta calibración es encontrar un curva que relacione la diferencia de presión normalizada (6.3) con el AOA. Para esto es necesario obtener el ángulo de ataque medido por la veleta pivotante, el cual se asumirá como el AOA real, esto se logra aplicando la ecuación (5.1) que relaciona la caída de voltaje en el potenciómetro de la veleta con el AOA.

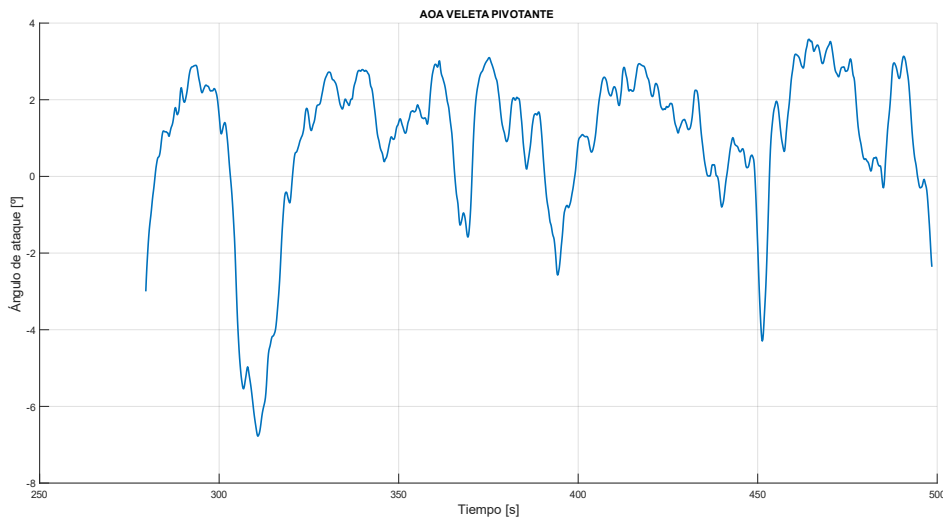


Figura 8.1: AOA obtenido de la veleta pivotante, vuelo N°2 10-01-2024.

Una vez obtenido el AOA de ataque real, se realiza un ajuste de curvas (regresión) entre los datos de diferencia de presión normalizados y los del AOA presentado en la Figura 8.1. La presión medida por los sensores se presenta en la Figura 8.2. La presión longitudinal se valida al compararla con la curva de la velocidad de avance (ver Figura 7.5), ya que tiene la misma forma y al convertir los valores de presión en velocidad estos tienen una desviación estándar promedio de 0,65 [m/s]. Por otro lado, la presión P_{45} tiene una tendencia a aumentar su magnitud cuando la presión longitudinal disminuye. Esta tendencia se puede observar en el intervalo de tiempo de 300 a 320 segundos, lo cual es coherente, ya que al observar la actitud de la aeronave justo en ese instante de tiempo se tiene un ángulo de pitch elevado, lo que implica que por la forma de la sonda diseñada la presión P_{45} debiese aumentar.

Aplicando la relación (6.3) se obtiene la relación de presiones normalizada para el vuelo N°2, la cual se presenta en la Figura 8.3.

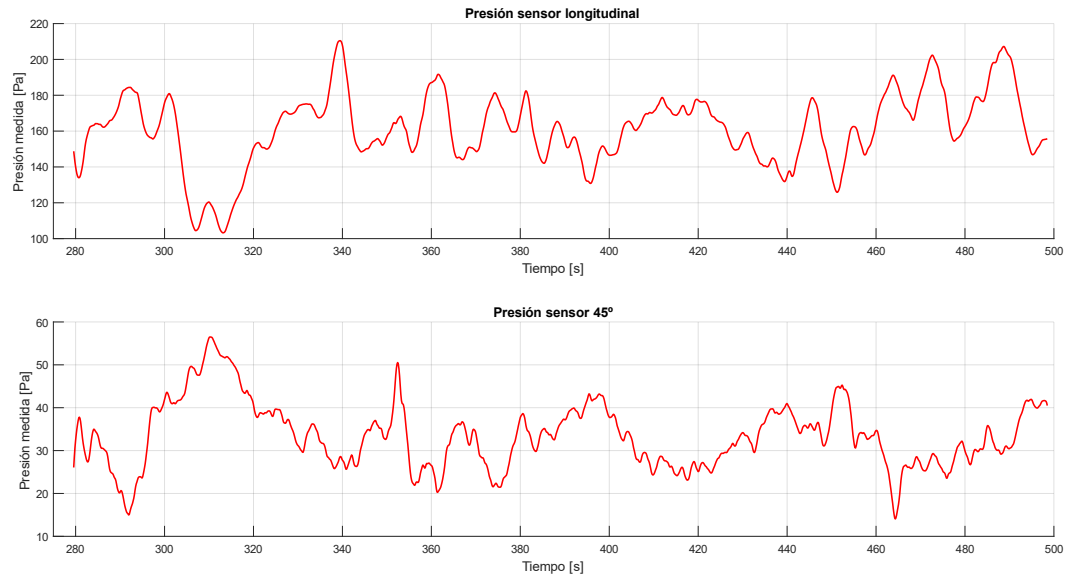


Figura 8.2: Presiones medidas en el vuelo N°2.

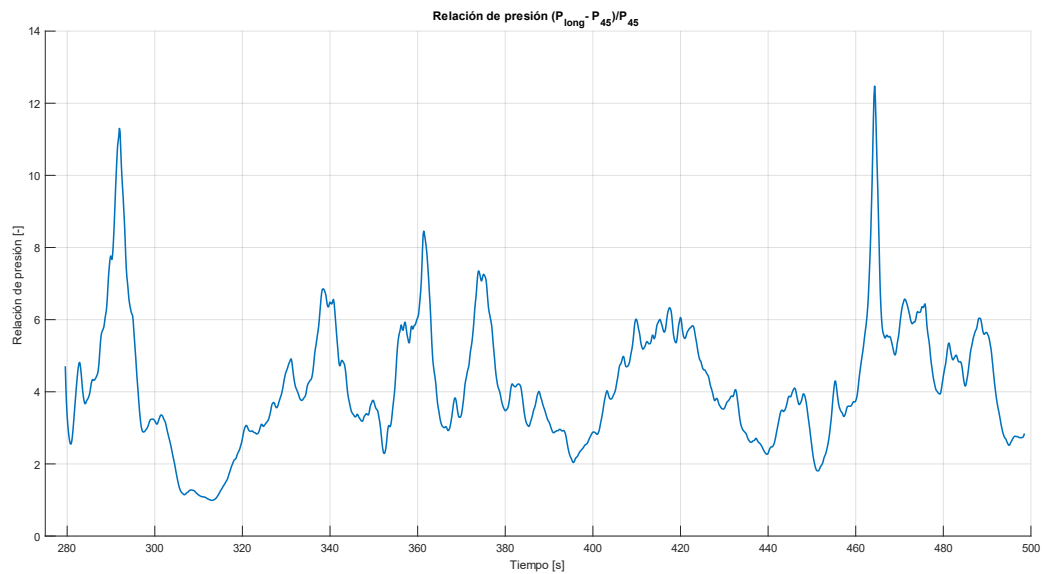


Figura 8.3: Relación de presión normalizada para vuelo N°2.

Como se menciona en el Capítulo 7 solo fue posible recopilar todos los datos necesarios para aplicar el método de medición en el vuelo N°2, por lo que se decide realizar la regresión polinomial con la mitad de dichos datos, con el fin de validar el método de medición con el resto de los datos del vuelo. En específico se utiliza los datos para la regresión hasta el segundo 389.

Para realizar la regresión polinomial se decide utilizar un polinomio de grado 3, debido a que fue el que mejor se ajustó a los datos. En la Figura 8.4 se presentan los datos de diferencia de presión normalizada vs AOA, además en la misma figura se muestra en color rojo la curva obtenida al aplicar la regresión polinomial. Con la curva de regresión polinomial obtenida, es posible relacionar la

diferencias de presión normalizada con el ángulo de ataque de la aeronave, es decir que para conocer el AOA por el método de medición se debe aplicar la siguiente relación.

$$AOA_{medición} = 0,0387 \cdot \left(\frac{\Delta P}{P_{45}}\right)^3 - 0,839 \cdot \left(\frac{\Delta P}{P_{45}}\right)^2 + 5,83 \cdot \frac{\Delta P}{P_{45}} - 10,7 \quad (8.1)$$

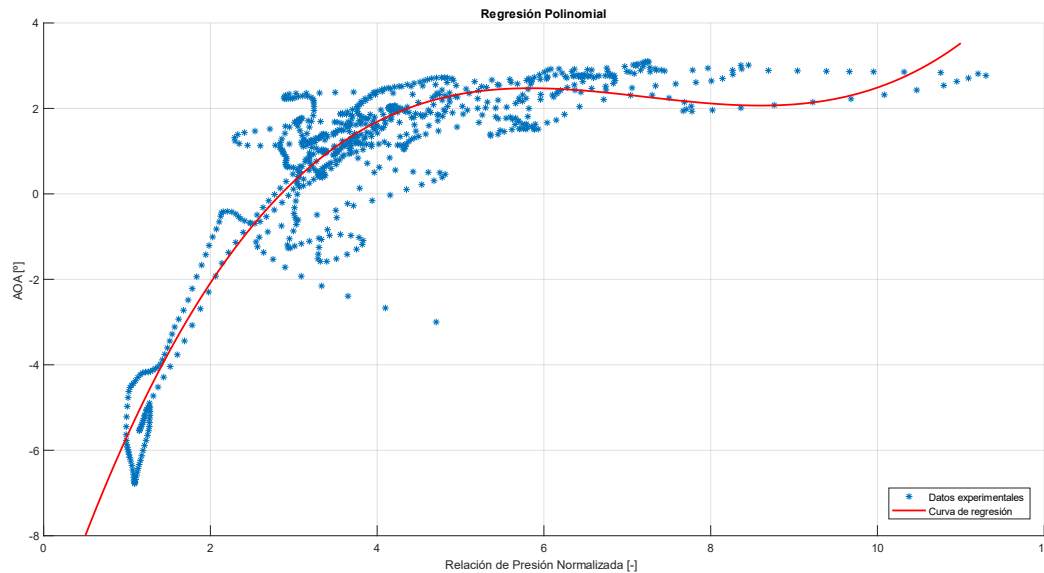


Figura 8.4: Curva obtenida de la regresión polinomial.

8.1.2 Resultados método de medición.

A continuación, se presentan los resultados del método de medición para los datos correspondientes al vuelo N°2 y que no se utilizaron para realizar el ajuste de curvas, es decir desde el segundo 772 en adelante (datos de validación). Se presenta el AOA real y el AOA obtenido con el método de medición.

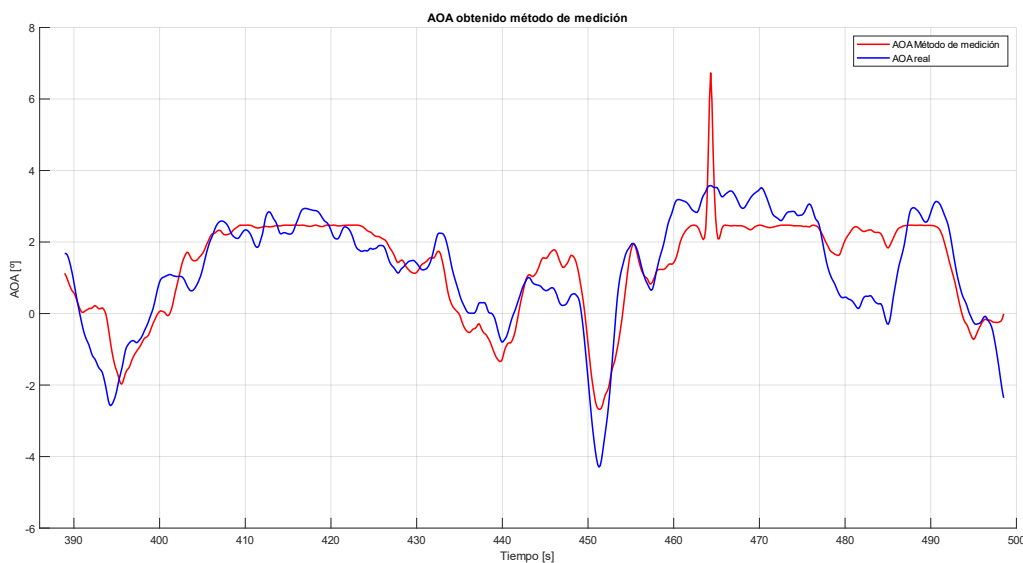


Figura 8.5: Ángulo de ataque obtenido del método de medición, para vuelo N°2.

En la figura anterior, se puede observar que el ángulo de ataque (AOA) obtenido mediante el método de medición se ajusta al AOA real. No obstante, en ciertos momentos se observan diferencias considerables. Estas discrepancias pueden atribuirse a diversas razones como lo es la calidad de las mediciones de los sensores de presión diferencial como también a la sonda diseñada, la cual no es capaz de captar en forma más precisa el comportamiento de la aeronave. A continuación, se presenta una tabla con las medidas estadísticas que describen las diferencias entre ambas curvas, además se incluye la diferencia entre AOA real y el AOA obtenido con el método de medición, pero utilizando los datos con los que se realiza la regresión polinomial (se denominaran datos de estimación).

Tabla 8-1: Error medio, desviación estándar y varianza para resultado de método de medición en vuelo N°2

Datos	Error medio	Desviación estándar	Varianza	Error medio cuadrático
Estimación	0,632	0,831	0,69	0,705
Validación	0,663	0,846	0,716	0,718

8.1.3 Analisis de resultados método de medición.

Como se puede observar en la Figura 8.5, el AOA obtenido del método diseñado si bien se tiende a ajustar a la curva del ángulo de ataque real, el resultado no es óptimo, ya que tiene una desviación estándar de $0,846^\circ$ lo cual si bien es un error menor a 1° no es un valor óptimo según lo presentado en el apartado 6.1.2, el cual habla sobre la precisión aceptable que se debe tener para evitar la entrada en pérdida o volar a una máxima eficiencia aerodinámica.

La principal razón de la diferencia entre el ángulo de ataque medido y el AOA real es la naturaleza de la regresión polinomial, debido a que esta busca encontrar ciertos coeficientes de una ecuación que mejor se ajusten a los datos reales, por lo que es imposible que el AOA por el método de medición se ajuste de manera igual al AOA real. Esto se puede corroborar con la Figura 8.4, donde se observa que existe una gran dispersión en los datos de presión diferencial normalizada con respecto a la curva de calibración obtenida.

Otro factor importante en la discrepancia entre el ángulo de ataque (AOA) medido y el AOA real se atribuye a las limitaciones en la calidad de las mediciones de presión diferencial efectuadas por los sensores MS4525do. Al analizar los datos, se observa que, en ciertos momentos temporales, la presión medida P_{45} experimenta fluctuaciones abruptas sin un fundamento físico coherente. Estos cambios indicarían, de manera inapropiada, que la aeronave está experimentando un AOA considerablemente alto o un AOA notablemente bajo. Además, debido a que la diferencia de presión se normaliza con respecto a la presión P_{45} , en los intervalos temporales en los cuales esta presión es errónea se generan resultados similares a los observados en la Figura 8.6 correspondiente a la presión normalizada del vuelo N°3, donde se evidencia un comportamiento inestable del AOA en el segundo 350 y 540.

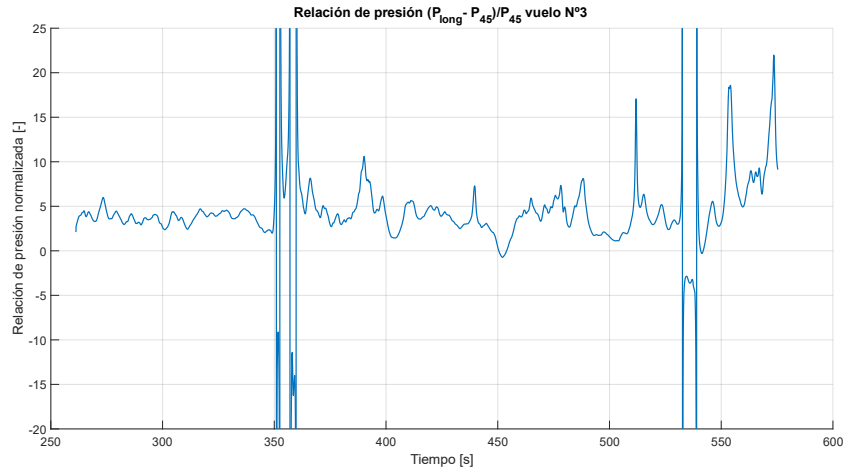


Figura 8.6: Relación de presión normalizada en vuelo N°3.

Sumado a lo anterior, la calidad de las mediciones de los sensores puede corroborarse al comparar la presión medida por el sensor de presión diferencial longitudinal con la presión dinámica. Esta última se calcula aplicando la fórmula (6.1) a la velocidad de avance obtenida a través de Mission Planner. Este software utiliza la medición del tubo de Pitot, junto con algoritmos internos del firmware, para estimar y corregir la velocidad de avance.

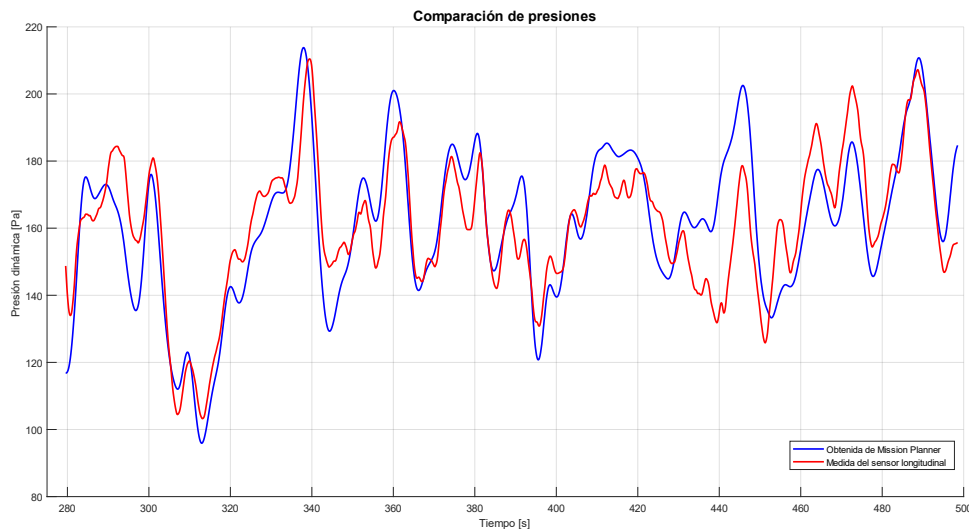


Figura 8.7: Comparación de presión dinámica en vuelo N°2.

En la figura anterior se observan ciertas diferencias entre ambas curvas, las cuales influyen en la obtención del AOA. Además, analizando la Figura 8.5 con la Figura 8.7 se observa que en los instantes de tiempo donde el AOA medido presenta menos error (instante donde se tiene una tendencia al aumento o disminución del AOA) coinciden con los instantes de tiempo donde las presiones mostradas en la Figura 8.7 son prácticamente iguales.

Además de la razón explicada anteriormente, existen diferentes causas que provocan los errores en la medición del AOA, como lo es la dinámica inestable de la actitud de la aeronave, lo que se traduce en fluctuaciones considerables en la medición de un instante de tiempo k a un instante de tiempo $k + 1$.

Esto se relaciona directamente con la frecuencia de muestreo de la medición la cual es de 10 datos por segundo (Frecuencia que se obtuvo del resultado de la Programación en LUA). Esta frecuencia es demasiado baja en comparación a otros sensores, como la IMU integrada en el auto piloto que tiene una frecuencia de muestreo de 1MHz. Esto conlleva a no capturar de buena manera las presiones asociadas a la actitud de la aeronave.

8.2 Método de estimación

8.2.1 Resultados y análisis de resultados.

A continuación, se presenta el resultado del AOA_{pseudo} al aplicar las ecuaciones (6.5) y (6.6) para los datos del vuelo N°2. En las cuales se utiliza una constante TC con un valor de 1,75 obtenida de las diferentes referencias aplicables y un Δt de 0,1 [s] aproximado a la cantidad de mediciones por cada segundo.

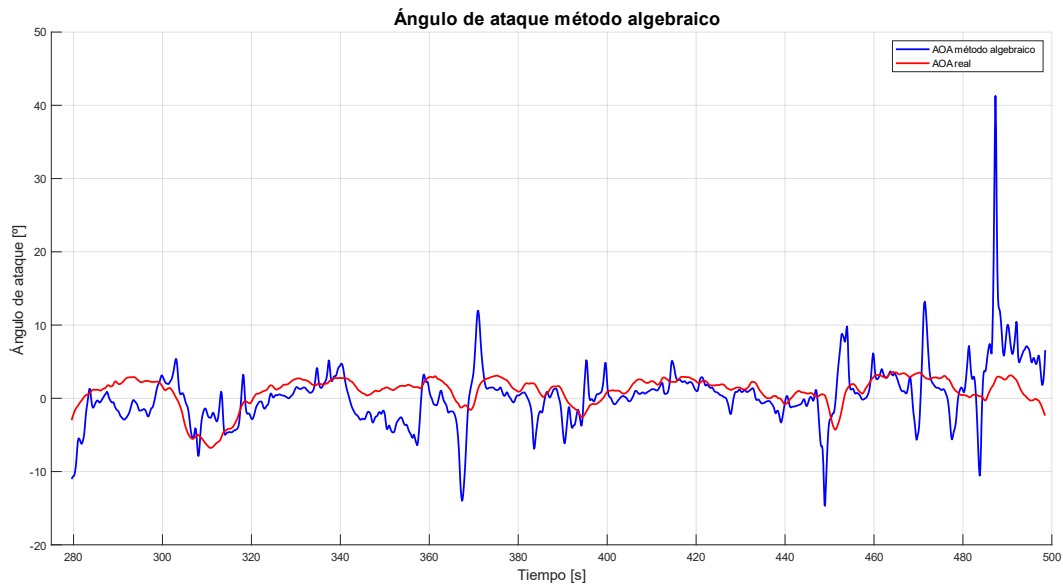


Figura 8.8: Resultado del ángulo de ataque por método algebraico para el vuelo N°2.

Se puede observar que el resultado obtenido mediante el método algebraico no se acerca exitosamente al ángulo de ataque real, sin embargo, el comportamiento de ambas curvas en la mayoría del tiempo es similar, lo que indica que el AOA del método algebraico fue capaz de capturar el comportamiento de la aeronave. No obstante, es importante destacar que existen ciertos instantes de tiempo donde el resultado obtenido no es coherente, como lo es en el segundo 490, donde el α_{pseudo} es aproximadamente de 40° , lo que es lejano a la realidad.

El segundo paso es aplicar el algoritmo de filtro de Kalman extendido, con el fin de obtener las velocidades en el cuerpo. Se aplica dicho algoritmo explicado en los capítulos anteriores, en el cual se hacen diferentes iteraciones dando énfasis en los parámetros críticos de los cuales depende el EKF, como son el paso de tiempo, ruido de la medición y ruido del proceso. Los resultados que se presentan a continuación son con un paso de tiempo (Δt) de 0,01 [s], un ruido en la medición de una varianza de 0,5 con el fin de ayudar a la convergencia del filtro y por último un ruido del proceso de varianza 1. Es importante mencionar que estos valores no representan la realidad de las mediciones, pero sin

embargo se seleccionan para lograr obtener resultados. Los resultados se presentan en la siguiente figura.

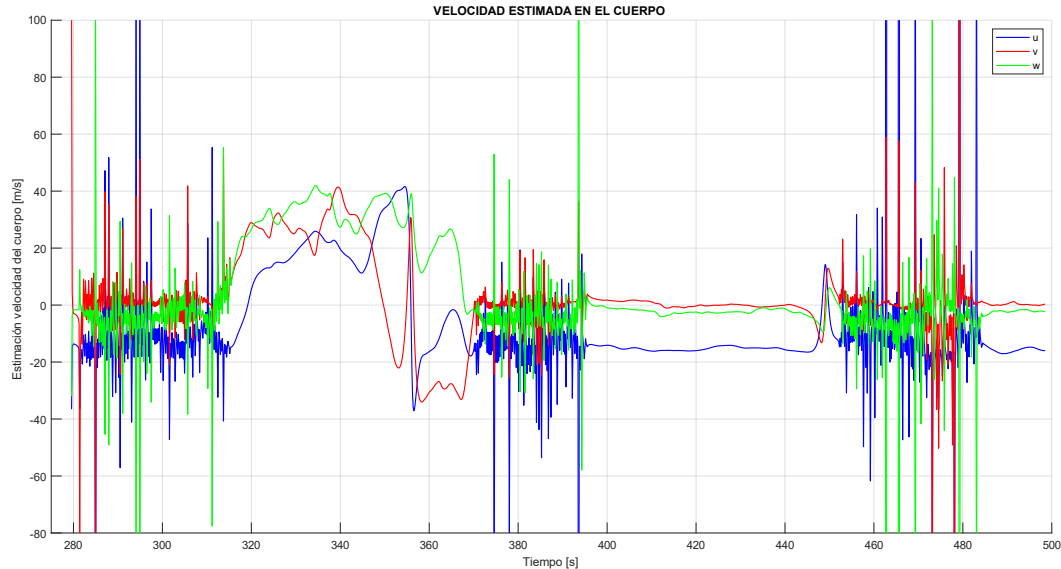


Figura 8.9: Resultados del EKF para vuelo N°2.

Se observa que los resultados derivados de la aplicación del EKF utilizando los datos del vuelo N°2 muestran inconsistencias. En ciertos instantes de tiempo el algoritmo no logra converger a ningún valor, coincidiendo dichos instantes con la discontinuidad observada en la gráfica del ángulo de yaw (Figura 7.4). Además, durante los intervalos en los que el algoritmo logra converger, se han detectado valores de estimación que carecen de coherencia física. Por ejemplo, se observa que la velocidad "u" adopta valores negativos, lo cual contradice los principios físicos, ya que implicaría que la aeronave estuviese retrocediendo, por otro lado, la velocidad "w" debiese estar cercano a cero, debido a que en los vuelos se hizo el esfuerzo de tener una altitud constante por lo que valores cercanos a 30 metros por segundos no representan la realidad del vuelo. Es importante destacar que al aplicar el EKF con los datos medidos en el resto de los vuelos las estimaciones tienen el mismo comportamiento por lo que no se presentan dichos resultados.

Los errores en la estimación vienen dados por diferentes factores, los cuales se explican a continuación:

1) En primer lugar, se tiene que uno de los principales problemas en el algoritmo utilizado para la estimación son las ecuaciones que se utilizaron para el paso de actualización. La función principal de estas ecuaciones en el algoritmo de Kalman es corregir la estimación a priori realizada en el paso de predicción, lo que se obtiene relacionando ecuaciones que dependan del vector estado y que su valor sea medible. En el caso de este proyecto se utilizan como ecuaciones de corrección la velocidad aerodinámica (velocidad de avance), las velocidades inerciales y por último la ecuación del AOA. Se detectó que el problema en este algoritmo fueron las ecuaciones que relacionan las velocidades en el cuerpo con las velocidades inerciales, esto se debe a que estas ecuaciones dependen de la actitud de

la aeronave y como se puede observar en la Figura 7.3 tanto el ángulo de pitch y roll son altamente variables en el tiempo por lo que afecta a la convergencia del filtro, además el ángulo de yaw presenta en ciertos momentos una discontinuidad por lo que el filtro pasa de un estado estable a un estado de inestabilidad. Otro factor que afecta a las ecuaciones en cuestión es la calidad de las velocidades inerciales, estas velocidades no se obtuvieron directamente de mediciones, se rescatan del software Mission Planner, el cual las estima utilizando como información principal los datos entregados por la brújula instalada en la aeronave. La brújula mencionada funciona con información GPS, esta presenta ciertas falencias en la operación, debido a que la exactitud de la información entregada no fue optima (no fue precisa), por lo que al usar esta información las estimaciones de las velocidades inerciales realizadas internamente en Mission Planner tienen un error. Lo explicado en este punto se puede corroborar con las siguientes figuras.

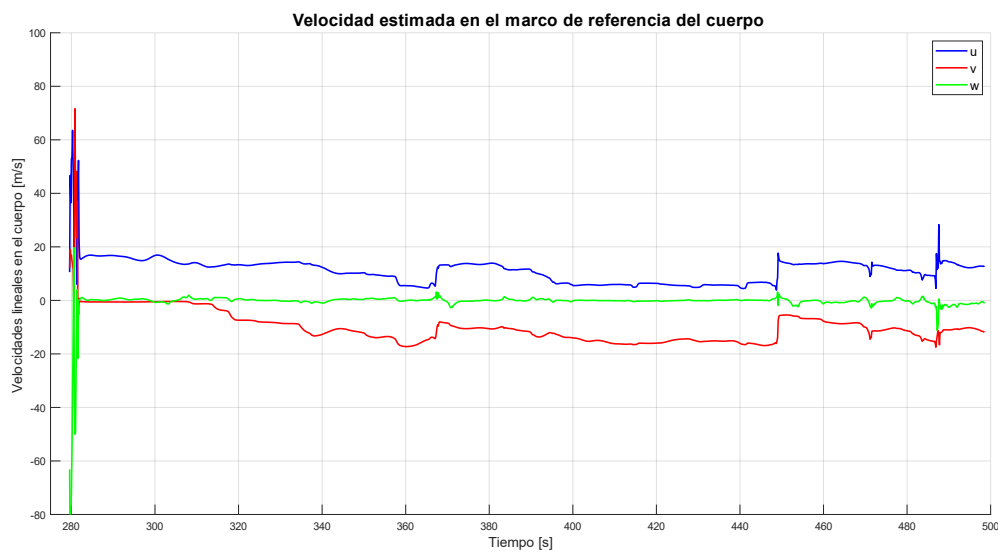


Figura 8.10: Velocidad estimada usando EKF sin las velocidades inerciales.

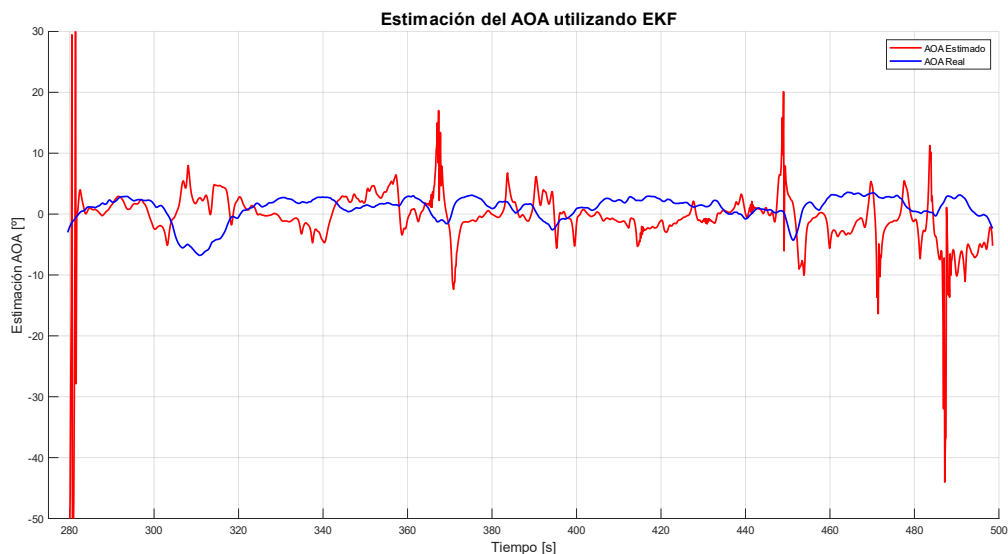


Figura 8.11: AOA estimado utilizando EKF sin las velocidades inerciales.

Las figuras anteriores corresponden a las estimaciones de la velocidad en el cuerpo y el posterior cálculo del AOA, eliminando del paso de actualización las ecuaciones correspondiente a las velocidades inerciales. Si bien las velocidades presentadas en la Figura 8.10 son erróneas debido a la carencia de sentido físico, se observa que el filtro de Kalman si logra converger de manera estable a ciertos valores, por lo que se comprueba que en el caso anterior las ecuaciones que relacionan las velocidades en el cuerpo con las velocidades inerciales llevaban a la inestabilidad en el algoritmo. Por otro lado, en la Figura 8.11 se presenta el cálculo del ángulo de ataque utilizando la ecuación (6.10) donde se observa que, en relación con la magnitud, esta toma valores coherentes en la mayoría del tiempo, esto se debe a que la ecuación de actualización relacionada con la velocidad aerodinámica corrige el cálculo del filtro asegurando que la norma de las velocidades estimadas sea correcta. Sin embargo, es claro que la estimación es errónea por lo que se evidencia que las velocidades inerciales no son el único error.

2) Otro aspecto importante que llevo a la falla el método para la estimación del AOA, fue la no consideración de alguna variable que relacione directamente una entrada de control con el ángulo de ataque y/o con la dinámica longitudinal de la aeronave, como por ejemplo la deflexión del elevador. Las ecuaciones utilizadas para el paso de predicción no consideraron este tipo de entradas y solo se toma en cuenta como variables del vector de control las aceleraciones y velocidades angulares, las cuales si bien representan la dinámica de la aeronave no son las óptimas para representar y relacionar la dinámica longitudinal con el AOA.

3) Si bien los puntos anteriores son las principales razones del porque fallo el método de estimación, el comportamiento de la aeronave igual influyo en los errores de la estimación. Esto debido a que en todo el tiempo de vuelo la aeronave no logro estar en un vuelo recto y nivelado, lo que se evidencia con la figura 7.3 donde se observa la actitud del vuelo. El cabeceo tuvo un comportamiento de ascenso y descenso en todo el instante de tiempo por lo que influye directamente en los resultados del AOA tanta para el método de estimación como para el método de medición. Por otra parte, se observa que el ángulo de roll varia en todo el instante de tiempo, por lo que se deduce que la aeronave en todo el vuelo tuvo un vuelo inestable con respecto al eje longitudinal.

4) La calidad de las mediciones igual tiene una influencia en el método de estimación, debido a que el filtro de Kalman es sensible al ruido de los datos. Como se puede observar en la Figura 7.1 las aceleraciones medidas por la IMU del auto piloto presentan un ruido considerable, el cual aún al aplicar un filtro de paso bajo no se logra obtener el comportamiento real de la aeronave. Este error en las mediciones viene dado por la calidad de la unidad de medida inercial y debido a que, en la implementación el autopiloto se instaló al interior de una caja de espuma genérica y no se implementó un método comprobado para aislar las vibraciones.

En la metodología se hace mención a que el algoritmo de EKF para el método de estimación se validara previamente con datos de vuelo del simulador Xplane. Al aplicar este algoritmo a dichos datos se obtuvieron los mismos errores que con los datos de la medición, por lo que no se presentan en el informe. Además, esto valida que las ecuaciones de propagación y corrección no son las adecuadas para este problema.

Conclusiones y trabajos futuros

9.1 Conclusiones

1. Se logra el diseño de un método de medición y estimación del ángulo de ataque para una aeronave no tripulada de ala fija.
 - ✓ El método de medición diseñado se basa en la correlación entre una diferencia de presión y el AOA. La medición de presión se realizó en dos puntos diferentes, una denominada presión longitudinal y la otra llamada presión a 45°. Con estas presiones se realizó una diferencia de presión normalizada donde se obtuvo una relación que es independiente de la velocidad de avance de la aeronave. Luego se realizó una regresión polinomial entre la relación obtenida de las presiones y el AOA medido por la veleta pivotante, obteniendo como resultado una curva polinomial de grado 3 la cual se utiliza para obtener el AOA.
 - ✓ El método de estimación diseñado se basa en la aplicación de un algoritmo extendido de Kalman, el cual tiene como objetivo estimar las velocidades lineales del cuerpo (u , v , w) y con estas calcular el AOA mediante ecuaciones rescatada de literatura. Por otra parte, para ayudar a la convergencia del filtro de Kalman se calcula algebraicamente el AOA el cual es utilizado como una pseudomedición en las ecuaciones de actualización del EKF.
2. Para la implementación de los métodos de medición y estimación del AOA diseñados, se logra concebir y diseñar el software y hardware necesario los cuales cumplen los requerimientos de dichos métodos.
 - ✓ El software que se utilizó para el método de medición se basa en el lenguaje de programación “C++” debido a la utilización de Arduino. La programación consistió en habilitar 4 pines digitales para la lectura de los sensores de presión mediante comunicación I2C, además debido a que Arduino no tiene la capacidad de almacenar datos se utiliza el lenguaje LUA para comunicar Arduino con Pixhawk, solucionando el problema de almacenamiento de datos y logrando guardar todos los datos necesarios con una misma frecuencia de muestreo lo que facilito el procesamiento de estos. Con respecto al Hardware del método de medición, se utilizó el microcontrolador Arduino UNO como núcleo. Las presiones se logran medir con los sensores de presión diferencial MS4525do, los cuales se seleccionaron por su bajo costo y alta resolución de muestreo de 0,84 [Pa] cumpliendo con el requerimiento del método a implementar de 4,42 [Pa]. La sonda para tomar la presión se diseña en el software Inventor y luego se fabrica mediante manufactura aditiva utilizando PLA, resultando en una sonda con un peso de 16 gramos.
 - ✓ El software utilizado en el método de estimación fue Matlab, debido a que los datos se trabajan post vuelo y por su disponibilidad. Con respecto al hardware concebido para el método de estimación, se decide utilizar el piloto automático Pixhawk 6c debido a su redundancia en las unidades de medida inercial y por la compatibilidad con el software Mission Planner utilizado para la operación de la aeronave. Además, se utiliza dicho piloto automático por la disponibilidad en el mercado. Por último, se utiliza la placa moduladora de potencia PM07 para facilitar la conexión entre los componentes en la aeronave.
3. Se logra la habilitación de la aeronave KADET LT-40 junto con las veletas pivotantes.

- ✓ En primer lugar, se realizó una inspección y análisis sobre el hardware primario necesario a implementar en la aeronave para una operación segura, llegando a la conclusión de reutilizar los servomotores presentes en el avión junto con el ESC y cambiar el sistema de propulsión con el receptor.
 - ✓ Se realizan pruebas en un banco de ensayo a diferentes configuraciones de subsistema de propulsión (motor-hélice), donde se selecciona el motor Turnigy SK4250-650 y una hélice 13x8 pulgadas, debido al empuje generado de 29,5 [N] y por presentar una disminución en las vibraciones.
 - ✓ Se repara la estructura de la aeronave dejando en condiciones óptimas para el vuelo el fuselaje, ala, empenaje y tren de aterrizaje.
 - ✓ Se realizan diferentes pruebas en tierra de comunicación entre el receptor y la radio, corroborando el funcionamiento correcto de las diferentes superficies de control. Igualmente se corrobora la comunicación con una distancia de 300 metros donde no hubo problemas.
 - ✓ Para validar la funcionalidad de la aeronave, se lleva a cabo un vuelo de prueba exclusivamente con el hardware primario. Durante este vuelo, se observa que después de 5 minutos de operación, la batería conserva el 40% de su carga inicial. Por consiguiente, con el fin de garantizar la seguridad en los vuelos durante la aplicación de los métodos desarrollados, se establece un límite máximo de vuelo de 6 minutos.
 - ✓ Se logra la habilitación de las veletas pivotantes, donde se obtiene la curva de calibración que relaciona la caída de voltaje en los potenciómetros con el ángulo AOA y AOS correspondientemente. Además, se obtiene que el rango de medición del AOA, el cual es de los $[-27^\circ, 27^\circ]$.
4. Con respecto a la implementación de los métodos diseñados se logra la implementación del hardware de ambos métodos. Para el método de medición se decidió instalar la sonda para medir las presiones bajo el ala derecha, logrando mejorar la estabilidad lateral de la aeronave y quitando posibles alteraciones del flujo por el tubo de Pitot y veletas pivotantes. Los sensores MS4525d junto con el Arduino se fijan dentro del fuselaje logrando una conexión fácil y accesible. Por otro lado, se implementan los elementos necesarios para el método de estimación, el Pixhawk se instaló cercano al centro de gravedad obteniendo así una medición de la dinámica de la aeronave apta para dicho método. Se logra una calibración exitosa tanto de los acelerómetros y brújula del auto piloto, además de una buena comunicación entre el radio control, con el receptor y la telemetría con la estación terrestre. Se concluye que el hardware está bien diseñado e implementado, debido a una serie de pruebas positivas que se realizaron en el laboratorio previo a los vuelos para la medición de datos.
5. Se logra realizar la validación del método de medición solo con un vuelo, obteniendo como resultado que el AOA medido tiene una desviación estándar de $0,86^\circ$ respecto al AOA real.
- ✓ Se concluye que el método de medición no es óptimo, debido a que no cumple la restricción de resolución de $0,25^\circ$ impuesto anteriormente, por lo que no sería capaz de prever una entrada en pérdida, o volar a una máxima eficiencia aerodinámica.
 - ✓ Las diferencias entre el AOA medido y el AOA real vienen dadas principalmente debido a que la sonda diseñada no representa el comportamiento real del flujo de aire que afecta al ala. Lo

- que implica que, al realizar la regresión polinomial, los datos medidos tengan una dispersión considerable con respecto a la curva obtenida de la calibración y en consecuencia respecto al AOA real.
- ✓ Se concluye que si bien los sensores tienen una resolución de medición de $0,84 [Pa]$ lo que cumple el requerimiento de resolución, la frecuencia de obtención de datos no fue óptima (10 datos por segundo), ya que no se logró captar de buena manera las presiones cuando la aeronave cambia su actitud de vuelo en forma repentina. Además, se evidencia al comparar las curvas de velocidades que la calidad de la medición mediante los sensores no es apropiada.
 - ✓ Si bien observando los resultados del método de medición, en ciertos lapsos de tiempo el AOA medido se acerca de buena manera a la curva del AOA real (Prácticamente idénticos), no se logra verificar en que etapa del vuelo el AOA medido es óptimo, debido a que no se consiguió discriminar las diferentes etapas del vuelo (ascenso, crucero, giros, descenso) en los datos medidos.
 - ✓ Por último, se atribuye un porcentaje de error debido a la inestabilidad (actitud inestable) de los vuelos realizados.
6. Se concluye que el método de estimación implementado no fue apropiado para la estimación del ángulo de ataque. Esto por dos razones principales.
- ✓ Se verifica que las ecuaciones de actualización no son las apropiadas, en específicos las ecuaciones que relacionan las velocidades inerciales con las velocidades del cuerpo.
 - ✓ Aun sacando las ecuaciones anteriormente dichas, el método no logra converger a valores apropiados, por lo que se concluye que las ecuaciones de propagación no son las apropiadas. Esto debido a que en el paso de predicción no se tiene en cuenta alguna variable de control que se relacione directamente con el AOA y describa de manera más específica la dinámica longitudinal.
 - ✓ La medición de los datos de la unidad inercial no fue representativa, debido a la presencia de ruido por la falta un método de aislamiento de vibraciones apropiado.

9.2 Trabajos futuros

- ✓ Para mejorar la Calidad de los datos a medir se propone en un futuro realizar los vuelos con el piloto automático activado en modo estabilizador, además de realizar las mediciones de aceleración y velocidades angulares mediante una IMU externa de mejor calidad.
- ✓ Para el método de medición se deben realizar más vuelos con el fin de obtener diferentes curvas de calibración para las diferentes etapas de vuelo. Además, se debe realizar la validación del método con más vuelos, con el fin de obtener una evaluación más representativa.
- ✓ Se deben modificar las ecuaciones de propagación del filtro extendido de Kalman, en las cuales se incluya la deflexión del elevador, con el objetivo de representar mejor la dinámica longitudinal.
- ✓ Cambiar las ecuaciones de actualización del EKF, por ecuaciones que relacionen directamente el AOA con estados que se puedan medir.
- ✓ Obtener la medición y estimación en tiempo real, con el objetivo de ingresar estos datos como alimentación para diferentes controladores.

Referencias

- [1] A. Israr, E. H. Alkhamash y M. Hadjouni, *Guidance, Navigation, and control for fixed-Wing UAV*, Karachi, Pakistan: Hindawi, 2021.
- [2] H. Shakhathreh, A. Sawalmeh, A. Al-Fuqaha y Z. Dou, *Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges*, Doha, Qatar: IEEE Access, 2017.
- [3] P. Romanowski, M. Mazur y J. McMillan, «Global market for commercial applications of drone technology valued at over \$127bn,» PWC, Warsaw, Polonia, 2016.
- [4] L. Sankaralingam y C. Ramprasadh, *Angle of attack measurement using low-cost 3D printed five hole probe for UAV applications*, Thanjavur, India: ELSEVIER, 2021.
- [5] J. D. Anderson, *Fundamentals of Aerodynamics*, New York: McGraw-Hill, 2011.
- [6] U. D. o. Transportation, *Pilots Handbook of Aeronautical Knowledge*, Oklahoma : Federal Aviation Administration , 2023.
- [7] P. Poksawat, L. Wang y A. Mohamed, *Gain Scheduled Attitude Control of Fixed-Wing UAV With Automatic Controller Tuning*, Melbourne, Australia: IEEE Transactions on control system technology, 2017.
- [8] P. Lu y B. Pierson, «Aircraft Terrain Following Based on a Nonlinear Continuous Predictive Control Approach,» *Journal Of Guidance, Control, And Dynamics*, vol. 18, nº 4, p. 7, 1995.
- [9] K. T. Wood, S. Araujo-Estrada, T. Richardson y S. Windsor, *Distributed Pressure Sensing-Based Flight Control for Small Fixed-Wing Unmanned Aerial Systems*, Bristol, United Kingdom: American Institute of Aeronautics and Astronautics, 2019.
- [10] L. Sankaralingam y C. Ramprasadh, *A comprehensive survey on the methods of angle of attack measurement and estimation in UAVs*, Thanjavur, India: Chinese Journal of Aeronautics, 2019.
- [11] C. Ramprasadh, S. Prem y L. Sankaralingam, *A SIMPLE METHOD FOR STIMATION OF ANGLE ATTACK*, Thanjavur, India: ELSEVIER, 2018.
- [12] G. León, *Estudio de métodos de medición y estimación del ángulo de ataque en aeronaves de ala fija*, Concepcion, Chile: Biblioteca Universidad de Concepción, 2023.
- [13] J. F. Alves Borges, *Desenvolvimento de um medidor de Angulo de ataque para aeronaves de pequeno porte*, Belo Horizonte: Escola de Engenharia de UFMG, 2008.
- [14] W. Gracey, *SUMMARY OF METHODS OF MESAURING ANGLE OF ATTACK ON AIRCRAFT*, Washington: National advisory committee for Aeronautics, 1958.

- [15] L. Sankaralingam y C. Ramprasadh, *A comprehensive survey on the methods of angle of attack measurement and estimation in UAVs*, Thanjavur: Chinese Society of Aeronautics and Astronautics & Beihang University, 2019.
- [16] P. P. Correa, *Sistema de medida del ángulo de ataque mediante sensores de presión diferencial*, Valencia, España: Universidad politecnica de Valencia, 2020.
- [17] D. F. Rogers, *Wind-Tunnel investigation of a general-Aviation Differential Pressure Angle-of-Attack Probe*, Annapolis, Maryland: American Insitute Of Aeronautics and Astronautics, 2013.
- [18] C. Ramprasadh y A. Hemendra, *A MONTE CARLO Analysis of sensor fusion algorithm for the estimation of aerodynamic angles in a mini aerial vehicle*, Bombay, India: Indian Institute of Technology Bombay, 2011.
- [19] C. Ramprasadh y A. Hemendra, *Multistage-Fusion Algorithm for Estimation of Aerodynamic*, Bombay, India: Indian Institute of Technology Bombay, 2012.
- [20] S. Prem, L. Sankaralingam y C. Ramprasadh, *Pseudomeasurement-Aided Estimation of Angle*, Thanjavur: the American Institute of Aeronautics and Astronautics, 2020.
- [21] T. Pengzhi, C. Haiyang, G. Yu y S. G. Hagerott, *UAV Flight Test Evaluation of Fusion Algorithms for Estimation of Angle of Attack and Sideslip Angle*, California: American Institute of Aeronautics and Astronautics , 2016.
- [22] T. Pengzhi y C. Haiyang, *Model Aided Estimation of Angle of Attack, Sideslip Angle, and 3D Wind without Flow Angle Measurements*, Kissimmee: American Institute of Aeronautics and Astronautics , 2018.
- [23] S. L. Morris, D. E. Bossert y W. F. Hallgren, *INTRODUCCTION TO AIRCRAFT FLIHT MECHANICS*, Virginia: American institute of aeronautics ands Astronautics, 2019.
- [24] A. Iqbal, *Applications of an Extended Kalman Filter in nonlinear mechanics*, Lahore, Pakistán: School of Systems and Technology University of Management and Technology, 2019.
- [25] K. Ogata, *Ingeniería de control moderna*, Madrid: PEARSON EDUCACIÓN, 2010.

Anexo A: Carta Gantt

CARTA GANTT - MEMORIA DE TITULO INGENIERÍA AEROSPAIAL

Implementación de un método de medición y estimación del ángulo de ataque en un aerona ve no tripulado de ala fija.

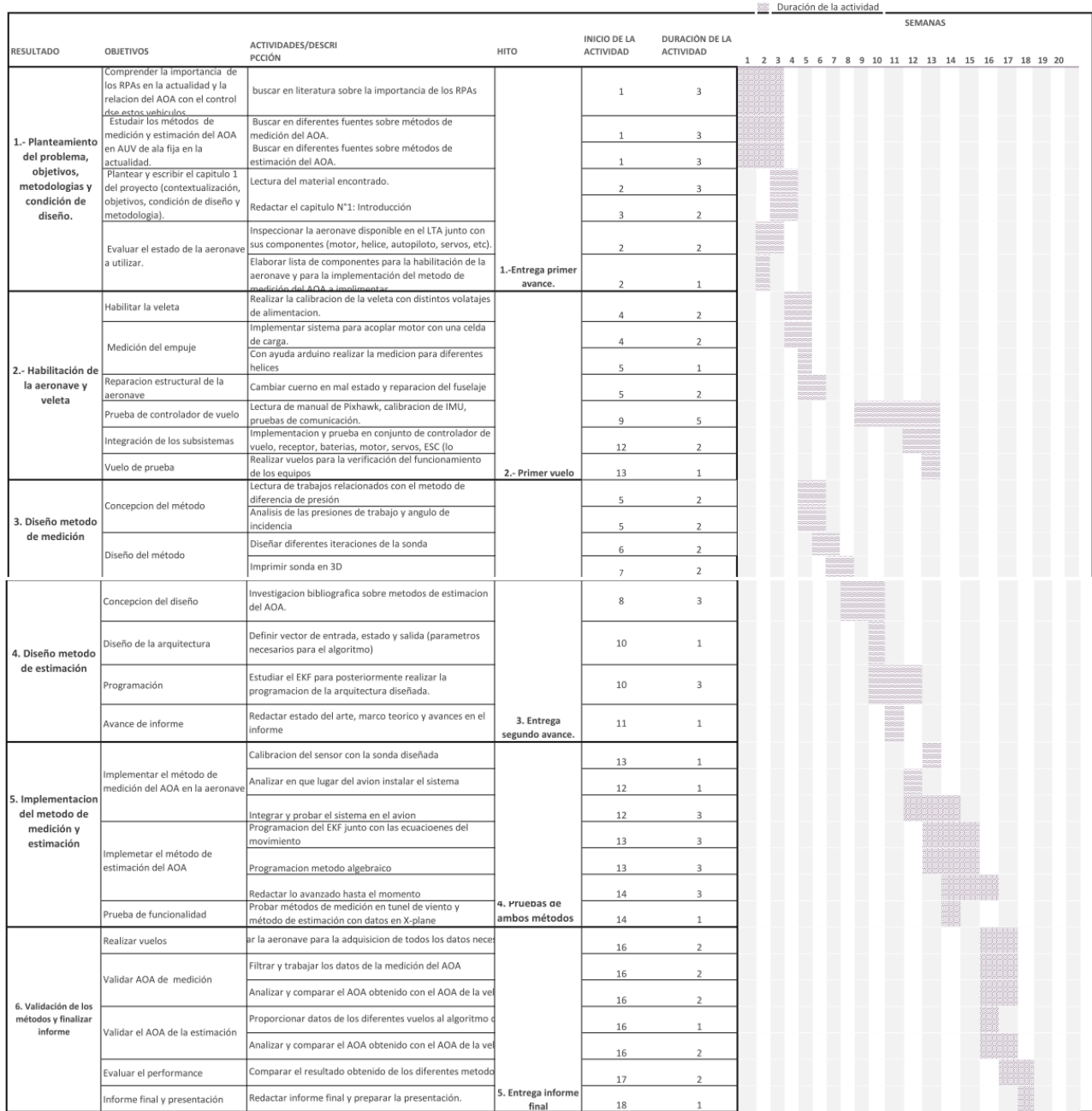


Figura anexo 1: Carta Gantt

Anexo B: Sistema de medición de empuje.

Para la selección del subsistema de propulsión se realiza la medición del empuje generado a diferentes configuraciones de Motor-Hélice. Estas mediciones se logran con la implementación de una celda de carga la cual soporta 10 kilogramos máximo junto con su módulo transmisor de celda de carga HX711. La lectura de los datos se realiza mediante Arduino UNO.



Figura anexo 2: Banco de ensayo para la medición de empuje.

Los valores valores de empuje medido para cada configuración estudiada se presentan en la siguiente tabla.

Tabla Anexo 1: Empuje generado por cada configuración

Motor	Hélice	Empuje generado [N]
Motor air	14"x7"x3"	31,1
SH4260-50	15"x8"	21,6
SH4260-50	13"x8"	14,3
SK4250-650	15"x8"	37,5
SK4250-650	13"x8"	29,5

Anexo C: Datos de los vuelos realizados.

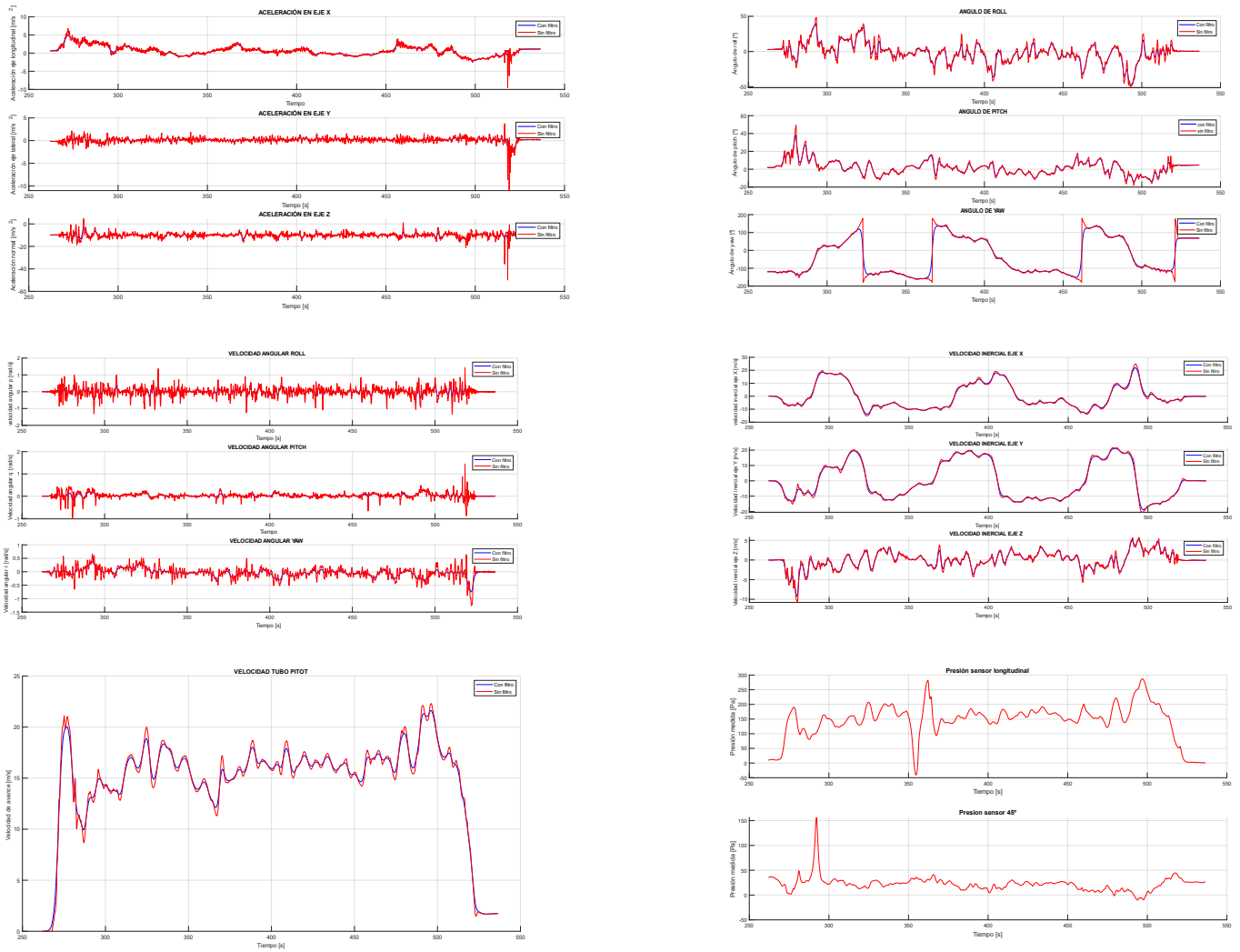


Figura anexo 3: Datos vuelo N°1.

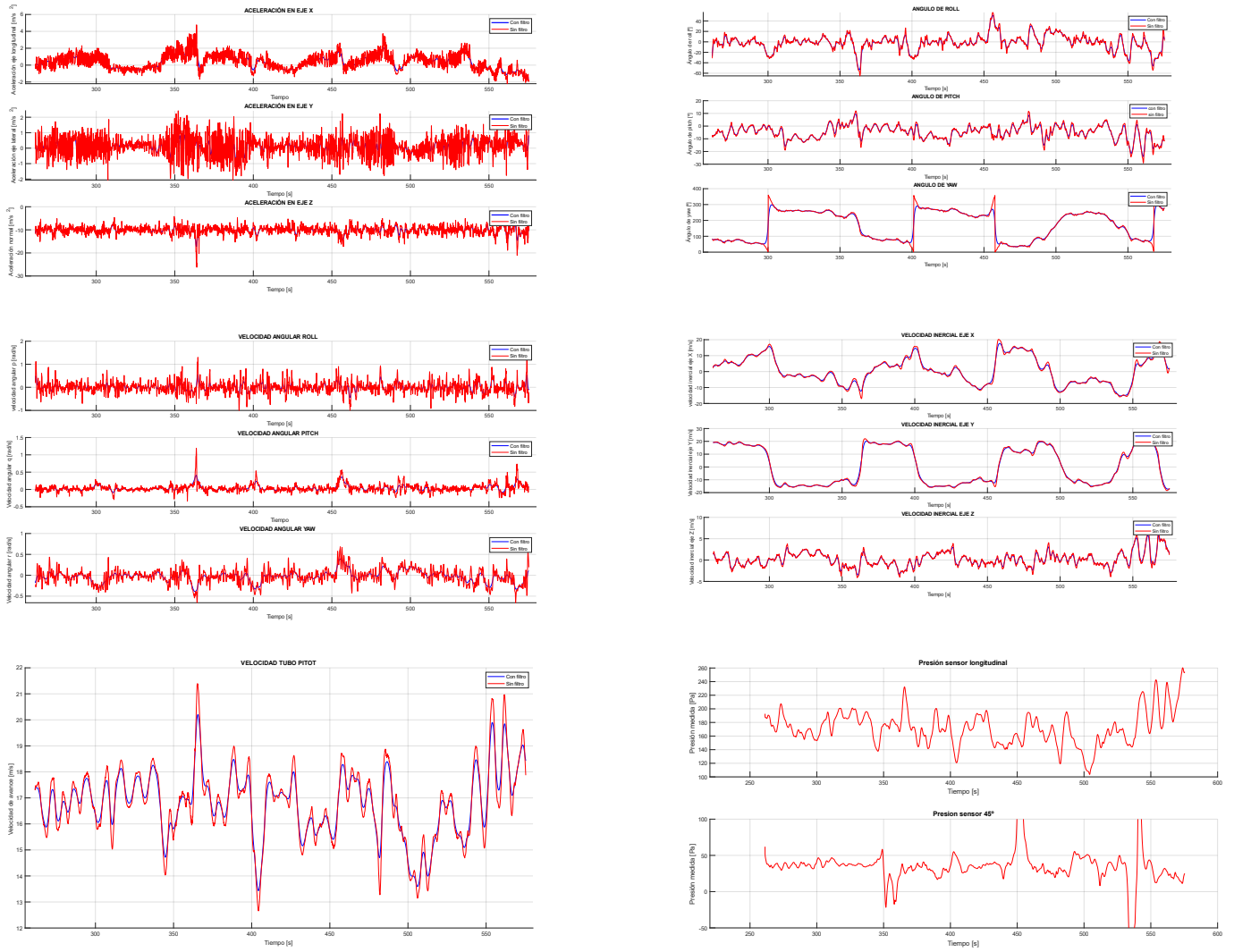


Figura anexo 4: Datos vuelo N°3

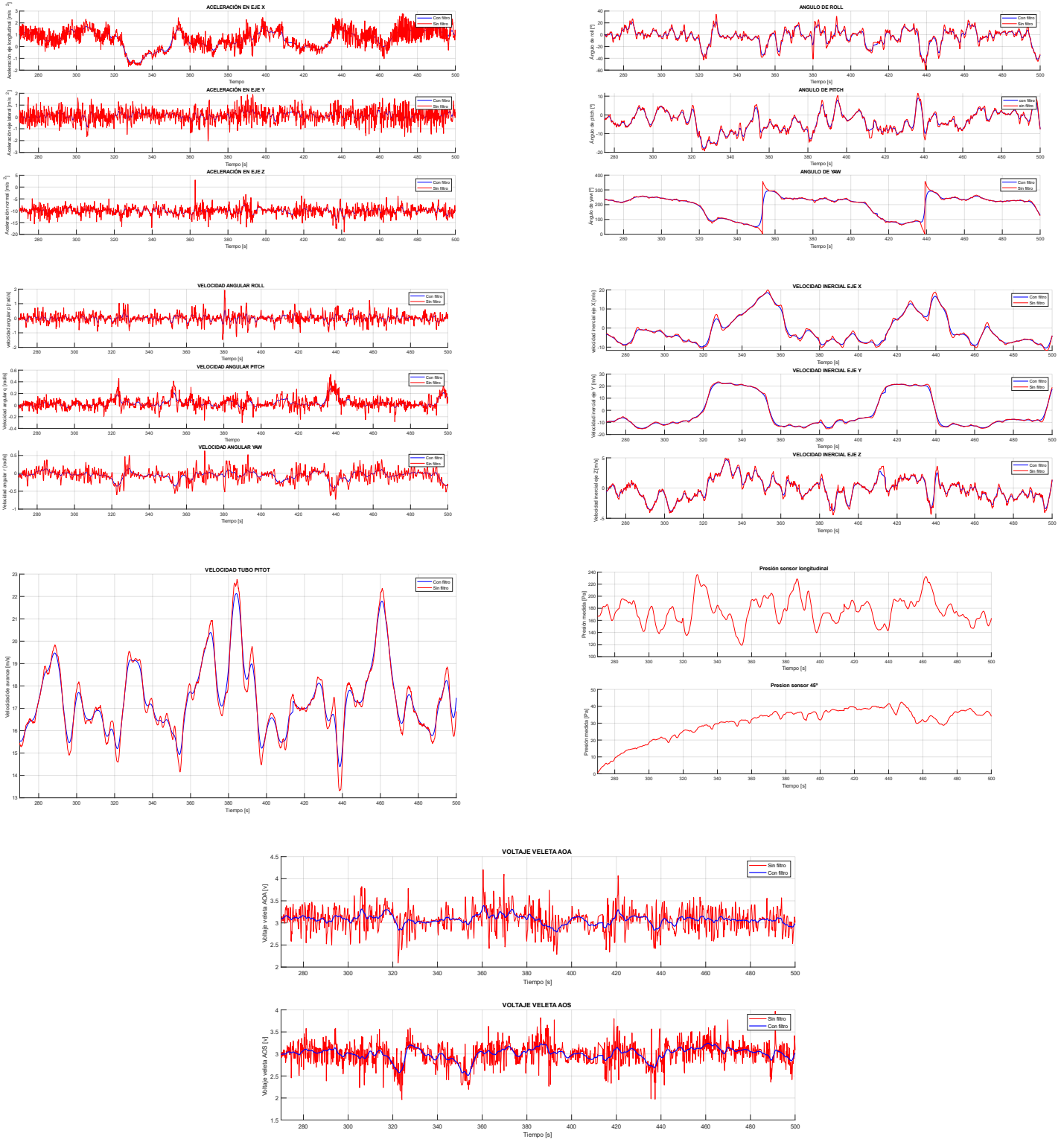


Figura anexo 5: Datos vuelo N°4.

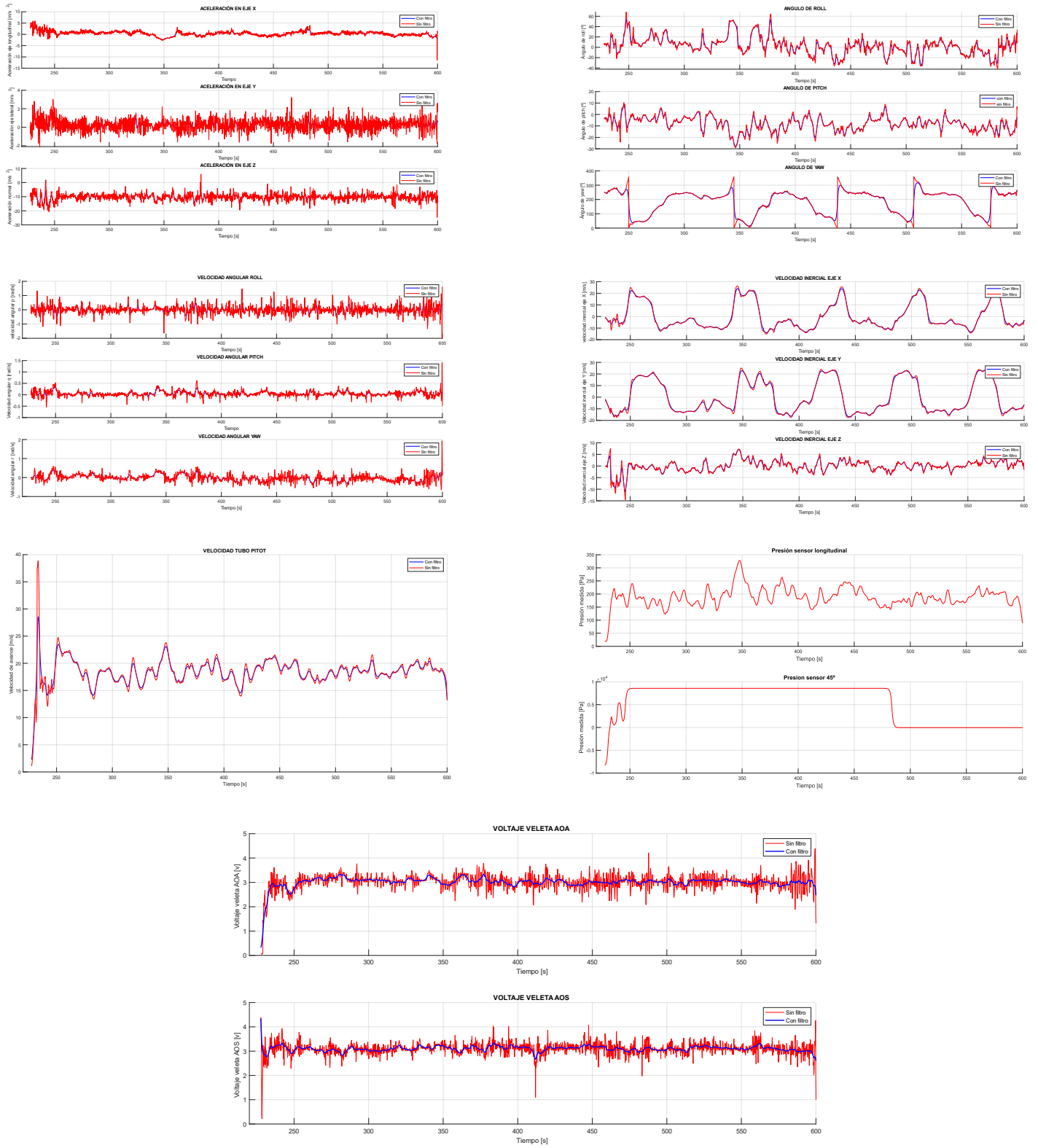


Figura anexo 6: Datos vuelo N°5.

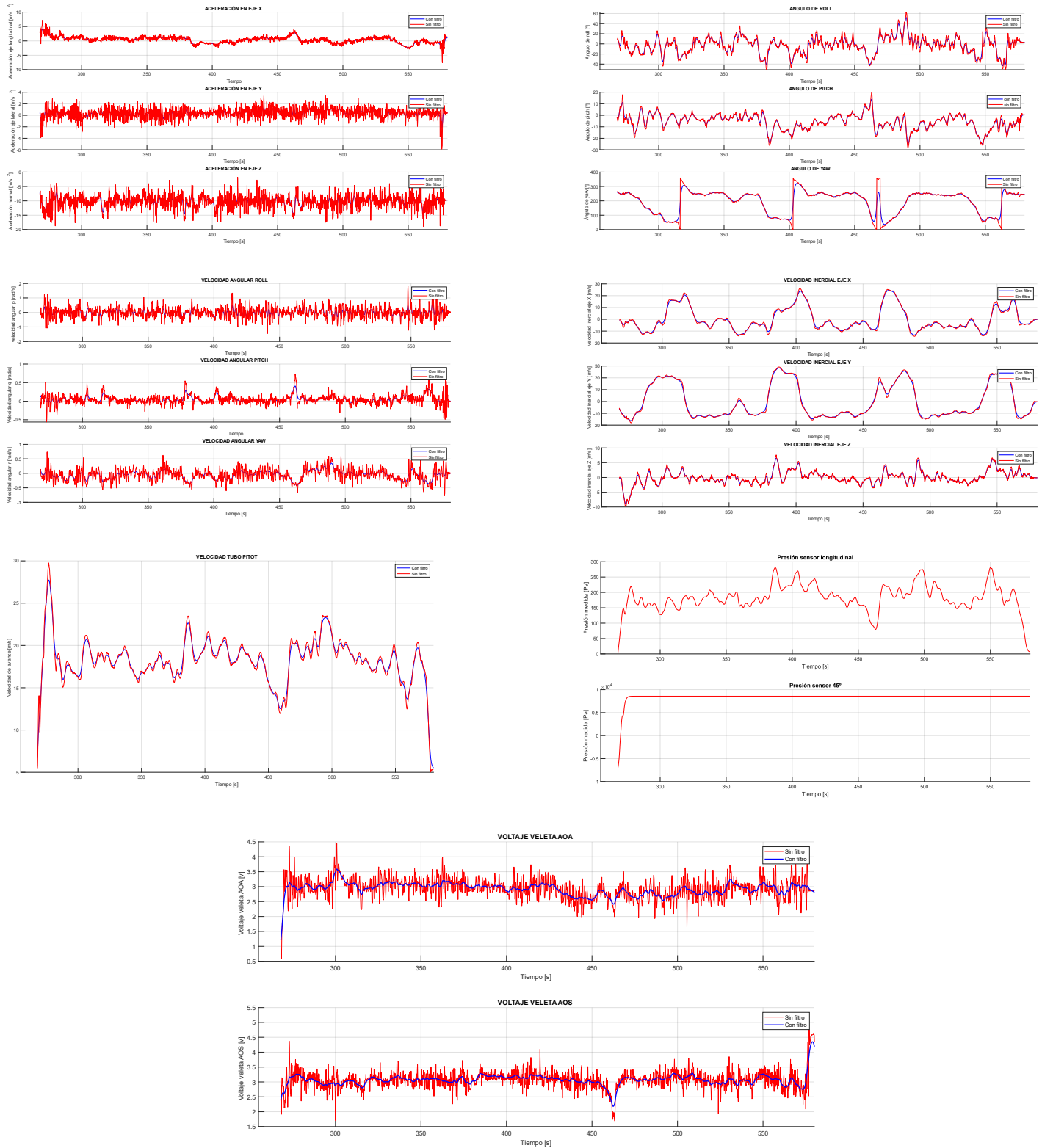


Figura anexo 7: Datos vuelo N°6.

Anexo D: Códigos.

Código para la calibración de las veletas:

```

1. clc; clearvars;
2. datos=importdata('datosveleta.txt');
3. relacion_AOA_vcc=datos(:,2); %voltaje en veleta/alimentacion
4. angulos=datos(:,1);
5. grado=1;
6. coeficientes_aoa=polyfit(relacion_AOA_vcc,angulos,grado)
7. relacion=linspace(0,100,100);
8. calibracion=coeficientes_aoa(1).*relacion+coeficientes_aoa(2);
9.
10. figure(1);clf
11. hold on
12. grid on
13. plot(relacion_AOA_vcc,angulos, '*', 'MarkerSize', 11, 'Color', 'b')
14. plot(relacion,calibracion, 'r', 'LineWidth', 1.7)
15. xlabel('Relación Voltaje AOA/Vcc [%]', 'FontSize', 14)
16. ylabel('Angulo de ataque [°]', 'FontSize', 14)
17. legend('Puntos experimentales', 'Linea de calibración', 'FontSize', 12)
18. title('Calibración veleta AOA', 'FontSize', 15)
19.
20. %% AOS %%
21. grados_aos=[-40 -20 -10 0 10 20 40];
22. voltajes=[0.005 0.966 1.705 2.44 3.58 3.89 5.04];
23. voltaje_alimentacion=5.047;
24. relacion_porcentajes_aos=(voltajes/voltaje_alimentacion)*100
25. coeficientes_aos=polyfit(relacion_porcentajes_aos,grados_aos,grado)
26. calibracion_aos=polyval(coeficientes_aos,relacion);
27.
28.
29. figure(2);clf
30. hold on
31. grid on
32. plot(relacion_porcentajes_aos,grados_aos, '*', 'MarkerSize', 11, 'Color', 'b')
33. plot(relacion,calibracion_aos, 'r', 'LineWidth', 1.7)
34. xlabel('Relación Voltaje AOS/Vcc [%]', 'FontSize', 14)
35. ylabel('Angulo de deslizamiento lateral [°]', 'FontSize', 14)
36. legend('Puntos experimentales', 'Linea de calibración', 'FontSize', 12)
37. title('Calibración veleta AOS', 'FontSize', 15)

```

Código para lectura de los sensores de presión diferencial.

Código principal:

```

1. #include <SoftwareWire.h>
2. #define sensor_direc 0x28
3.
4. SoftwareWire myWire(2, 3);
5. SoftwareWire myWire2(4, 5);
6.
7. #include <SoftwareSerial.h>
8. const byte rxPin = 7;
9. const byte txPin = 6;
10. SoftwareSerial SerialAPixHawk (rxPin, txPin);
11.
12. float offset;
13. const int veleta_AOA = A1;
14. const int alimentacion=A0;
15. const int veleta_SSA=A2;
16.
17.
18. float deltapres_fwd, deltapres_45;
19. float pfdwnocarga, p45nocarga, pfdwnocargapsi, p45nocargapsi;
20. float pfdwcorr_pas, p45corr_pas;
21. int lectura1=0;
22. int lectura45=0;
23. int valor=63;
24. int Pmax, Pmin, cuenta;
25. float l45, lfwd;
26. float pfdw_p45; //relacion entre pfdw y p45
27. float relacion, lfwd_corr, l45_corr;
28.
29.
30. void setup() {
31.   Serial.begin(9600);
32.   myWire.begin();
33.   myWire2.begin();
34.   SerialAPixHawk.begin(115200);
35.
36.   pfdwnocarga = 8190; //corroborar bn estos valores
37.   p45nocarga = 8230; //corroborar bn estos valores
38.   cuenta=16383;
39.   Pmax=1; //psi
40.   Pmin=-1; //psi
41.   pfdwnocargapsi=((pfdwnocarga-(0.1*cuenta))*((Pmax-Pmin)/(0.8*cuenta))+Pmin);
42.   p45nocargapsi=((p45nocarga-(0.1*cuenta))*((Pmax-Pmin)/(0.8*cuenta))+Pmin);
43.
44. }
45.
46. void loop() {
47.   //conexion con el sensor fwd
48.   myWire.beginTransaction(sensor_direc);
49.   myWire.write(0);
50.   myWire.endTransmission();
51.   myWire.requestFrom(sensor_direc, 2);
52.   if (2<=myWire.available()) {
53.     lectura1= myWire.read();
54.     lectura1=lectura1 & valor;
55.     lectura1=lectura1 << 8;
56.     lectura1|= myWire.read();
57.   }
58.   //conexion con el sensor p45

```

```

59. myWire2.beginTransmission(sensor_direc);
60. myWire2.write(0);
61. myWire2.endTransmission();
62. myWire2.requestFrom(sensor_direc, 2);
63. if (2<=myWire2.available()) {
64.     lectura45= myWire2.read();
65.     lectura45=lectura45 & valor;
66.     lectura45=lectura45 << 8;
67.     lectura45|= myWire2.read();
68.
69. }
70. //lecturas de la presion en bytes como float
71. l45=lectura45;
72. l45=lectura1;
73. l45_corr=l45-p45nocarga;
74. l45_corr=l45-p45nocarga;
75. delpres_fwd=((l45-(0.1*cuenta))*((Pmax-Pmin)/(0.8*cuenta))+Pmin)-
    p45nocargapsi; //presion en psi TIPO A corregida
76. delpres_45=((l45-(0.1*cuenta))*((Pmax-Pmin)/(0.8*cuenta))+Pmin)-
    p45nocargapsi; //presion en psi TIPO A corregida
77. p45corr_pas=delpres_fwd*6894.75; //persion en fwd en pascales
78. p45corr_pas=delpres_45*6894.75; //presion en 45 en pascales
79.
80. p45corr_pas=p45corr_pas/p45corr_pas; //relacion entre presion fwd y 45
81. relacion=l45_corr/l45_corr;
82.
83.
84. //LECTURAS DE LAS VELETAS
85. int lectura_veleta_AOA = analogRead(veleta_AOA);
86. int alimentacion = analogRead(alimentacion);
87. int lectura_veleta_SSA= analogRead(veleta_SSA);
88. offset=0.06;
89. float voltage_AOA =offset+lectura_veleta_AOA*(5.0/1023.0);
90. float alimentacion =offset+lectura_alimentacion*(5.0/1023.0);
91. float voltage_SSA =offset+lectura_veleta_SSA*(5.0/1023.0);
92.
93.
94.
95. Serial.print("presion p45: ");
96. Serial.print(p45corr_pas);
97. Serial.print(" Pa");
98. Serial.print("\t"); // Agrega un carácter de tabulación
99. Serial.print("presion 45: ");
100. Serial.print(p45corr_pas);
101. Serial.print(" Pa");
102. Serial.print("\t");
103. Serial.print("relacion:");
104. Serial.print(p45corr_pas);
105. Serial.print("\t");
106. Serial.print("relacion 2:");
107. Serial.print(relacion); // Imprime y añade un salto de línea al final
108.
109. Serial.print("\t");
110. Serial.print("Voltaje alimentacion:");
111. Serial.print(alimentacion);
112. Serial.print("\t");
113. Serial.print("Voltaje AOA:");
114. Serial.print(voltage_AOA);
115. Serial.print("\t");
116. Serial.print("Voltaje SSA:");
117. Serial.println(voltage_SSA);
118.
119. struct valores {

```



```

120.         float a;
121.         float b;
122.         float c;
123.         float d;
124.         float e;
125.         float f;
126.     };
127.     valores v = {
128.         pfwddcorr_pas,
129.         p45corr_pas,
130.         pfwdd_p45,
131.         alimentacion,
132.         voltage_AOA,
133.         voltage_SSA
134.     };
135.     uint8_t size = sizeof(v);
136.     SerialAPixHawk.print("init");
137.     SerialAPixHawk.write(size);
138.     SerialAPixHawk.write(reinterpret_cast<uint8_t*>(&v), size);
139.
140.     delay(10);
141.
142.     }

```

Librería de los sensores MS4525do:

```

1.  #ifndef SoftwareWire_h
2.  #define SoftwareWire_h
3.
4.  #include <Arduino.h>
5.
6.
7.  // Transmission status error, the return value of endTransmission()
8.  #define SOFTWAREWIRE_NO_ERROR      0
9.  #define SOFTWAREWIRE_BUFFER_FULL  1
10. #define SOFTWAREWIRE_ADDRESS_NACK  2
11. #define SOFTWAREWIRE_DATA_NACK     3
12. #define SOFTWAREWIRE_OTHER         4
13.
14. #define SOFTWAREWIRE_BUFSIZE 32      // same as buffer size of Arduino Wire library
15.
16.
17. class SoftwareWire
18. {
19. public:
20.     SoftwareWire();
21.     SoftwareWire(uint8_t sdaPin, uint8_t sclPin, boolean pullups = true, boolean
detectClockStretch = true);
22.     ~SoftwareWire();
23.     void end();
24.
25.     void begin();
26.
27.     // Generate compile error when slave mode begin(address) is used
28.     void __attribute__((error("I2C/TWI Slave mode is not supported by the SoftwareWire
library"))) begin(uint8_t addr);
29.     void __attribute__((error("I2C/TWI Slave mode is not supported by the SoftwareWire
library"))) begin(int addr);
30.
31.     void setClock(uint32_t clock);
32.     void beginTransmission(uint8_t address);

```

```

33. void beginTransmission(int address);
34. uint8_t endTransmission(boolean sendStop = true);
35. uint8_t requestFrom(uint8_t address, uint8_t size, boolean sendStop = true);
36. uint8_t requestFrom(int address, int size, boolean sendStop = true);
37. size_t write(uint8_t data);
38. size_t write(const uint8_t *data, size_t quantity);
39. int available(void);
40. int read(void);
41. int readBytes(uint8_t* buf, uint8_t size);
42. int readBytes(char * buf, uint8_t size);
43. int readBytes(char * buf, int size);
44. int peek(void);
45. void setTimeout(long timeout); // timeout to wait for the I2C bus
46. void printStatus(Print& Ser); // print information using specified object class
47.
48.
49. private:
50. // per object data
51.
52. uint8_t _sdaPin;
53. uint8_t _sclPin;
54. uint8_t _sdaBitMask;
55. uint8_t _sclBitMask;
56.
57. volatile uint8_t *_sdaPortReg;
58. volatile uint8_t *_sclPortReg;
59. volatile uint8_t *_sdaDirReg;
60. volatile uint8_t *_sclDirReg;
61. volatile uint8_t *_sdaPinReg;
62. volatile uint8_t *_sclPinReg;
63.
64. uint8_t _transmission; // transmission status, returned by endTransmission(). 0 is
    no error.
65. uint16_t _i2cdelay; // delay in micro seconds for sda and scl bits.
66. boolean _pullups; // using the internal pullups or not
67. boolean _stretch; // should code handle clock stretching by the slave or not.
68. unsigned long _timeout; // timeout in ms. When waiting for a clock pulse stretch.
    2017, Fix issue #6
69.
70. uint8_t rxBuf[SOFTWAREWIRE_BUFSIZE]; // buffer inside this class, a buffer per
    SoftwareWire.
71. uint8_t rxBufPut; // index to rxBuf, just after the last valid byte.
72. uint8_t rxBufGet; // index to rxBuf, the first new to be read byte.
73.
74. // private methods
75.
76. void i2c_writebit( uint8_t c );
77. uint8_t i2c_readbit(void);
78. void i2c_init(void);
79. boolean i2c_start(void);
80. void i2c_repstart(void);
81. void i2c_stop(void);
82. uint8_t i2c_write(uint8_t c);
83. uint8_t i2c_read(boolean ack);
84. };
85.
86. #endif // SoftwareWire_h

```

Código para la comunicación entre Arduino y Pixhawk.

```

1. local baud_rate = 115200
2.
3. local port = assert(serial:find_serial(0), "LUA: Puerto serial no configurado")
4.
5. port:begin(baud_rate)
6. port:set_flow_control(0)
7.
8. local file_name = "datos.csv"
9. file = io.open(file_name, "a")
10.
11. local header = "init"
12. local header_count = 0
13. local bytes_to_read = 0
14. local buffer = ""
15.
16. local new_save_time = 0
17.
18. local max_bytes_per_update = 100
19. local lecturas = 0
20.
21. function inicio()
22.     gcs:send_text(0, "LUA: Iniciado")
23.     new_save_time = millis() + 2000
24.     return update, 100
25. end
26.
27. function update()
28.     local bytes = math.min(port:available(), max_bytes_per_update)
29.     while bytes > 0 or bytes_to_read > 0 do
30.         local n = port:read()
31.         if bytes_to_read == 0 then
32.             if n == string.byte(header, header_count + 1) then
33.                 header_count = header_count + 1
34.                 if header_count == string.len(header) then
35.                     header_count = 0
36.                     bytes_to_read = -1
37.                 end
38.             else
39.                 header_count = 0
40.             end
41.         elseif bytes_to_read == -1 then
42.             bytes_to_read = n
43.             buffer = {}
44.         elseif bytes_to_read > 0 then
45.             table.insert(buffer, string.char(n))
46.             bytes_to_read = bytes_to_read - 1
47.             if bytes_to_read == 0 then
48.                 lecturas = lecturas + 1
49.                 data = table.concat(buffer)
50.                 values = table.pack(string.unpack("<ffffff", data))
51.                 table.remove(values, #values)
52.                 str = tostring(millis()) .. ","
53.                 str = str .. table.concat(values, ",") .. ","
54.                 local accel = ahrs:get_accel()
55.                 local gyro = ahrs:get_gyro()
56.                 local pos = ahrs:get_position()
57.                 local vel = ahrs:get_velocity_NED()
58.                 local wind = ahrs:wind_estimate()
59.                 local airspeed = ahrs:airspeed_estimate()
60.                 str = str .. table.concat({accel:x(), accel:y(), accel:z()}, ",") .. ","

```

```

61.         str = str .. table.concat({gyro:x(), gyro:y(), gyro:z()}, ",") .. ","
62.         str = str .. table.concat({ahrs:get_roll(), ahrs:get_pitch(),
ahrs:get_yaw()}, ",") .. ","
63.         if (pos == nil) then
64.             str = str .. "0,0,0,"
65.         else
66.             str = str .. table.concat({pos:lat(), pos:lng(), pos:alt()}, ",") ..
", "
67.         end
68.         if (vel == nil) then
69.             str = str .. "0,0,0,"
70.         else
71.             str = str .. table.concat({vel:x(), vel:y(), vel:z()}, ",") .. ","
72.         end
73.         str = str .. table.concat({wind:x(), wind:y(), wind:z()}, ",") .. ","
74.         if (airspeed == nil) then
75.             str = str .. "0\n"
76.         else
77.             str = str .. tostring(airspeed) .. "\n"
78.         end
79.         file:write(str)
80.     end
81.     end
82.     bytes = bytes - 1
83. end
84. local now = millis()
85. if (now > new_save_time) then
86.     new_save_time = now + 2000
87.     gcs:send_text(0, "LUA: Guardadas " .. lecturas .. " lecturas en la SD")
88.     lecturas = 0
89.     file:flush()
90. end
91.
92. return update, 1
93. end
94.
95. return inicio, 100

```

Código de validación del EKF.

Código principal:

```

1. clc; clearvars;
2. %%simulacion de la dinamica%%
3. global gamma omega beta AMP rho dt
4.
5. gamma=0.3;
6. omega=1;
7. beta=1;
8. rho=1.5;
9. AMP=0.5;
10. T=40;
11. dt=0.01;
12. %dt=2*pi/rho/T;
13. time= 0:dt:T;
14. x0=[1 0];
15. [t, truetrayectoria]=ode45(@(t,x)duffing1(t,x),time,x0);
16. yx=truetrayectoria(:,1);
17. yv=truetrayectoria(:,2);
18.
19. %% error al sensor
20. obsnoise=0.05;
21. rng(2);
22. obs_pos=truetrayectoria(:,1);
23. ruido=randn(size(obs_pos));
24. pos_sensor=obs_pos+obsnoise*ruido; %añade ruido a la solucion de la dinamica
25. error=obs_pos-pos_sensor;
26. covarianceerror=cov(error)
27. %% aplicación del filtro %%
28. %valores iniciales
29. mu=[1;0];
30. sigma=diag([0.1,1]);
31. var_c=10^2;
32. R=obsnoise;
33.
34.
35. elementos=zeros(length(pos_sensor),2);
36. for i=1:length(pos_sensor)
37.     [mu_bar, sigma_bar] = prediction(mu, sigma, var_c);
38.     [mu, sigma] = update(mu_bar,sigma_bar,pos_sensor(i), R);
39.     elementos(i,:)=mu_bar';
40.
41. end
42.
43. %% Obtencion del error medio cuadratico
44. mse_pos = mean((yx-elementos(:,1)).^2);
45. mse_vel = mean((yv-elementos(:,2)).^2);
46. rmse_pos=sqrt(mse_pos)
47. rmse_vel=sqrt(mse_vel)
48.
49. %% Graficos
50.
51. subplot(2,1,1)
52. %figure(1);clf
53. hold on
54. grid on
55. plot(t,pos_sensor,'g','LineWidth',0.1,'Marker','*','LineStyle','none')
56. plot(time,yx,'b','LineWidth',1.4)
57. plot(time,elementos(:,1),'r','LineWidth',1.4)
58. xlabel('Tiempo [s]')

```

```

59. ylabel('Posición [m]')
60. legend('Medición', 'Posición verdadera', 'Estimación EKF')
61. title('Posición')
62.
63. subplot(2,1,2)
64. %figure(2);clf
65. grid on
66. hold on
67. plot(time,yv, 'b', 'LineWidth',1.5)
68. plot(time,elementos(:,2), 'r', 'LineWidth',1.5)
69. xlabel('Tiempo [s]')
70. ylabel('Velocidad [m/s]')
71. legend('Velocidad verdadera', 'Estimación EKF')
72. title('Velocidad')

```

Función de predicción:

```

1. function [mu_bar, sigma_bar] = prediction(mu, sigma, var_c, P)
2. global gamma beta AMP omega rho dt
3. dtt=2*pi/rho/10;
4. x = mu(1);
5. x_dot = mu(2);
6. Aa=[0 1;omega^2-3*beta*x_dot^2 -gamma];
7. F=eye(length(mu))+(Aa*dt)+(Aa^2*dt^2)/factorial(2); % +(Aa^3.*dt^3)/factorial(3);
   %definicion de la matriz de transicion de estado
8. mu_bar = [x + dt*x_dot; x_dot + (-gamma*x_dot*omega^2*x-beta*x^3+AMP*cos(rho*dt))*dt];
9. P=diag([sigma(1,1),sigma(2,2)]);
10. M=[0 0;0 var_c];
11. V=[0 0;0 -AMP*sin(rho*dt)];
12. sigma_bar=F*P*F'+V*M*V';
13.
14. end

```

Función de actualización:

```

1. function [mu,sigma]=update(mu_bar,sigma_bar,pos_sensor, R)
2. x=mu_bar(1);
3. x_dot=mu_bar(2);
4. H=[1 0];
5. K=(sigma_bar*H')/(H*sigma_bar*H'+R);
6. y=pos_sensor-H*mu_bar; %residual (medicion-estimacion)
7. mu=mu_bar+K*(y);
8. sigma=sigma_bar-K*H*sigma_bar;
9. end

```

Código para el cálculo del AOA por el método algebraico.

```

1. clc; clearvars;
2. datosvuelo=importdata('datos_vuelo2.txt');
3. tiempo=datosvuelo(:,1)./1000; %milisegundos
4. presion_lon=datosvuelo(:,2); %pascales
5. presion_45=datosvuelo(:,3); %pascales
6. relacion_presion=datosvuelo(:,4); %-
7. V_cc=datosvuelo(:,5); %voltaje de alimetnacion en veleta
8. V_AOA=datosvuelo(:,6); %volataje veleta AOA
9. V_AOS=datosvuelo(:,7); %volatje veleta AOS
10. ax=datosvuelo(:,8); %aceleración en el eje longitudinal body-frame
11. ay=datosvuelo(:,9); %aceleración en el eje lateral body-frame
12. az=-datosvuelo(:,10); %aceleración en el eje normal body-frame
13. p=datosvuelo(:,11); % velocidad angular p rad/s
14. q=datosvuelo(:,12); % velocidad angular q rad/s
15. r=datosvuelo(:,13); % velocidad angular r rad/s
16. roll=datosvuelo(:,14)*57.29; %en grados
17. pitch=datosvuelo(:,15)*57.29; %en grados
18. yaw=datosvuelo(:,16)*57.29; %en grados
19. VX=datosvuelo(:,20); % velocidad sistema inercial m/s
20. VY=datosvuelo(:,21); % velocidad sistema inercial m/s
21. VZ=datosvuelo(:,22); % velocidad sistema inercial m/s
22. AIRSPEED=datosvuelo(:,26); %velocidsd de avance m/s
23.
24. %% cambiar angulo yaw %%
25. for i=1:length(tiempo)
26.     if yaw(i)<0
27.         YAW(i)=360-(yaw(i)*-1);
28.     else
29.         YAW(i)=yaw(i);
30.     end
31. end
32.
33.
34.
35. %% FUNCION FILTRO PASO BAJO
36. [V_ccfiltrado,V_AOAFiltrado,V_AOSfiltrado,presion_lonfiltrada,presion_45filtrada,ax_filtra
do,ay_filtrado,az_filtrado,p_filtrado,q_filtrado,r_filtrado,roll_filtrado,pitch_filtrado,y
aw_filtrado,VX_filtrado,VY_filtrado,VZ_filtrado,AIRSPEED_filtrado]=filtropasobajo(tiempo,p
resion_lon,presion_45,V_cc,V_AOA,V_AOS,ax,ay,az,p,q,r,roll,pitch,YAW,VX,VY,VZ,AIRSPEED);
37.
38. %% METODO ALGEBRAICO %%
39.
40.
41. X0=[0 0];
42.
43. for i=1:length(tiempo)
44.     fun = @(AA) [((1/(AIRSPEED_filtrado(i)*cosd(AA(2))))*((-
ax_filtrado(i)*sind(AA(1)))+(az_filtrado(i)*cosd(AA(1)))))-
((p_filtrado(i)*cosd(AA(1))+r_filtrado(i)*sind(AA(1)))*tand(AA(2))); % ecuación 1
45.         (1/AIRSPEED_filtrado(i))*((-
ax_filtrado(i)*cosd(AA(1))*sind(AA(2)))+(ay_filtrado(i)*cosd(AA(2)))-
(az_filtrado(i)*sind(AA(1))*sind(AA(2)))+(p_filtrado(i)*sind(AA(1))]); % ecuación
2
46.
47.     angulos_nr=fsolve(fun,X0);
48.     X0=[angulos_nr(1) angulos_nr(2)];
49.     ANGULOS_ALGE(i,:)=angulos_nr;
50. end
51.
52. %% grafico %%

```

```

53. AOA_NR=ANGULOS_ALGE(:,1);
54. AOS_NR=(ANGULOS_ALGE(:,2));
55. AOA_pseudo=AOA_NR+(1.75*0.01*q_filtrado);
56. AOS_pseudo=AOS_NR-(1.75*0.01.*r_filtrado.*cosd(AOA_pseudo));
57. nombre_archivo='AOA_pseudo.txt';
58. delimitador = ' ';
59. dlmwrite(nombre_archivo, AOA_pseudo, 'delimiter', delimitador);
60.
61. figure(1)
62. hold on
63. grid on
64. plot(tiempo,AOS_NR,LineWidth=1.6)
65. xlim([275 500])
66. xlabel('Tiempo [s]', FontSize=13)
67. ylabel('Ángulo de ataque NR [°]', FontSize=13)
68. title('Ángulo de ataque método algebraico',FontSize=16)
69.
70. %% AOA Y AOS REAL %%
71. AOA_filtrado=0.589*(100*(V_AOAfiltrado./V_cc))-32.76;
72. AOS_filtrado=0.749*(100*(V_AOSfiltrado./V_cc))-37.41;
73.
74. figure(2)
75. hold on
76. grid on
77. plot(tiempo,AOA_pseudo,LineWidth=1.6,Color='b')
78. plot(tiempo,AOA_filtrado,LineWidth=1.6,Color='r')
79. xlim([275 500])
80. xlabel('Tiempo [s]', FontSize=13)
81. ylabel('Ángulo de ataque [°]', FontSize=13)
82. title('Ángulo de ataque método algebraico',FontSize=16)
83. legend('AOA método algebraico','AOA real')
84.
85. figure(3)
86. hold on
87. grid on
88. plot(tiempo,AOS_pseudo,LineWidth=1.6,Color='b')
89. plot(tiempo,AOS_filtrado,LineWidth=1.6,Color='r')
90. xlim([275 500])
91. xlabel('Tiempo [s]', FontSize=13)
92. ylabel('Ángulo de deslizamiento lateral NR [°]', FontSize=13)
93. title('Ángulo de deslizamiento lateral método algebraico',FontSize=16)
94. legend('AOS método algebraico','AOS real')

```


Código método de medición.

```

1. clc; clearvars;
2. datosvuelo=importdata('datos_vuelo2.txt');
3. tiempo=datosvuelo(:,1)./1000; %milisegundos
4. presion_lon=datosvuelo(:,2); %pascales
5. presion_45=datosvuelo(:,3); %pascales
6. relacion_presion=datosvuelo(:,4); %-
7. V_cc=datosvuelo(:,5); %voltaje de alimetnacion en veleta
8. V_AOA=datosvuelo(:,6); %volataje veleta AOA
9. V_AOS=datosvuelo(:,7); %volatje veleta AOS
10. ax=datosvuelo(:,8); %aceleración en el eje longitudinal body-frame
11. ay=datosvuelo(:,9); %aceleración en el eje lateral body-frame
12. az=datosvuelo(:,10); %aceleración en el eje normal body-frame
13. p=datosvuelo(:,11); % velocidad angular p
14. q=datosvuelo(:,12); % velocidad angular q
15. r=datosvuelo(:,13); % velocidad angular r
16. roll=datosvuelo(:,14); %en grados
17. pitch=datosvuelo(:,15); %en grados
18. yaw=datosvuelo(:,16); %en grados
19. VX=datosvuelo(:,20); % velocidad sistema inercial m/s
20. VY=datosvuelo(:,21); % velocidad sistema inercial m/s
21. VZ=datosvuelo(:,22); % velocidad sistema inercial m/s
22. AIRSPEED=datosvuelo(:,26); %velocidsd de avance m/s
23. %% filtro de paso bajo para V_AOA %%
24. [V_ccfiltrado,V_AOAfiltrado,V_AOSfiltrado,presion_lonfiltrada,presion_45filtrada,ax_filtra
do,ay_filtrado,az_filtrado,p_filtrado,q_filtrado,r_filtrado,roll_filtrado,pitch_filtrado,y
aw_filtrado,VX_filtrado,VY_filtrado,VZ_filtrado,AIRSPEED_filtrado]=filtropasobajo(tiempo,p
resion_lon,presion_45,V_cc,V_AOA,V_AOS,ax,ay,az,p,q,r,roll,pitch,yaw,VX,VY,VZ,AIRSPEED);
25.
26. figure(1)
27. subplot(2,1,1)
28. hold on
29. grid on
30. plot(tiempo,V_AOA,'r','LineWidth',0.6)
31. plot(tiempo,V_AOAfiltrado,'b','LineWidth',1.6)
32. xlabel('Tiempo [s]')
33. ylabel('Voltaje veleta AOA [v]')
34. legend('Sin filtro','Con filtro')
35. title('VOLTAJE VELETA AOA')
36. xlim([255 580])
37.
38. subplot(2,1,2)
39. hold on
40. grid on
41. plot(tiempo,V_AOS,'r','LineWidth',0.6)
42. plot(tiempo,V_AOSfiltrado,'b','LineWidth',1.6)
43. xlim([255 580])
44. xlabel('Tiempo [s]')
45. ylabel('Voltaje veleta AOS [v]')
46. legend('Sin filtro','Con filtro')
47. title('VOLTAJE VELETA AOS')
48.
49.
50. %% AOA ° %%
51. AOA=0.589*(100*(V_AOA./V_cc))-32.76;
52. AOA_filtrado=0.589*(100*(V_AOAfiltrado./V_cc))-32.76;
53. figure(2)
54. hold on
55. grid on
56. plot(tiempo,AOA_filtrado,'LineWidth',1.6)
57. xlabel('Tiempo [s]','FontSize',13)

```

```

58. ylabel('Ángulo de ataque [°]', 'FontSize',13)
59. title('AOA VELETA PIVOTANTE')
60.
61.
62.
63.
64. %% Regreción entre presiones y AOA %%
65. relacion_p=(presion_lonfiltrada-presion_45filtrada)./presion_45filtrada;
66. [PP SS]=polyfit(relacion_p,AOA_filtrado,3)
67. XX=linspace(0.5,11,1545);
68. CURVA=polyval(PP,XX);
69.
70. figure(3)
71. hold on
72. grid on
73. plot(relacion_p,AOA_filtrado,'*')
74. plot(XX,CURVA,'r','LineWidth',1.6)
75. xlabel(['Relación de Presión Normalizada [-]'])
76. ylabel('AOA [°]')
77. title('Regresión Polinomial')
78. legend('Datos experimentales','Curva de regresión')
79.
80. %% AOA METODO DE MEDICIÓN%%
81. AOA_medido=polyval(PP,relacion_p);
82. error_medio=mean(abs(AOA_filtrado-AOA_medido))
83. desviacion_estandar=std(AOA_filtrado-AOA_medido)
84. varianza=var(AOA_filtrado-AOA_medido)
85. figure(4)
86. hold on
87. grid on
88. plot(tiempo,AOA_medido,'r',tiempo,AOA_filtrado,'b','LineWidth',1.4)
89. xlabel('Tiempo')
90. ylabel('AOA [°]')
91. title('AOA OBTENIDO METODO DE MEDICIÓN')
92. legend('AOA Metodo de medición','AOA real')
93.
94. %% GRAFICOS PRESION %%
95.
96. figure(5)
97. subplot(2,1,1)
98. hold on
99. grid on
100.     plot(tiempo,presion_lonfiltrada,'r','LineWidth',1.4)
101.     xlabel('Tiempo [s]',FontSize=12)
102.     xlim([275 500])
103.     ylabel('Presión medida [Pa]',FontSize=12)
104.     title('Presión sensor longitudinal',FontSize=12)
105.
106.
107.     subplot(2,1,2)
108.     hold on
109.     grid on
110.     plot(tiempo,presion_45filtrada,'r','LineWidth',1.4)
111.     xlabel('Tiempo [s]',FontSize=12)
112.     xlim([275 500])
113.     ylabel('Presión medida [Pa]',FontSize=12)
114.     title('Presión sensor 45°',FontSize=12)

```

Código método de estimación.

Código principal:

```

1. clc; clearvars;
2. datosvuelo=importdata('datos_vuelo2.txt');
3. tiempo=datosvuelo(:,1)./1000; %milisegundos
4. presion_lon=datosvuelo(:,2); %pascales
5. presion_45=datosvuelo(:,3); %pascales
6. relacion_presion=datosvuelo(:,4); %-
7. V_cc=datosvuelo(:,5); %voltaje de alimetnacion en veleta
8. V_AOA=datosvuelo(:,6); %volataje veleta AOA
9. V_AOS=datosvuelo(:,7); %volatje veleta AOS
10. ax=datosvuelo(:,8); %aceleración en el eje longitudinal body-frame
11. ay=datosvuelo(:,9); %aceleración en el eje lateral body-frame
12. az=-datosvuelo(:,10); %aceleración en el eje normal body-frame
13. p=datosvuelo(:,11); % velocidad angular p rad/s
14. q=datosvuelo(:,12); % velocidad angular q rad/s
15. r=datosvuelo(:,13); % velocidad angular r rad/s
16. roll=datosvuelo(:,14)*57.29; %en grados
17. pitch=datosvuelo(:,15)*57.29; %en grados
18. YAW=datosvuelo(:,16)*57.29; %en grados
19. VX=datosvuelo(:,20); % velocidad sistema inercial m/s
20. VY=datosvuelo(:,21); % velocidad sistema inercial m/s
21. VZ=datosvuelo(:,22); % velocidad sistema inercial m/s
22. AIRSPEED=datosvuelo(:,26); %velocidsd de avance m/s
23.
24. AOA_pseudo=importdata('AOA_pseudo.txt');
25. %% cambiar angulo yaw %%
26. % for i=1:length(tiempo)
27. %     if yaw(i)<0
28. %         YAW(i)=360-(yaw(i)*-1);
29. %     else
30. %         YAW(i)=yaw(i);
31. %     end
32. % end
33.
34.
35.
36. %% FUNCION FILTRO PASO BAJO
37. [V_ccfiltrado,V_AOAfiltrado,V_AOSfiltrado,presion_lonfiltrada,presion_45filtrada,ax_filtra
do,ay_filtrado,az_filtrado,p_filtrado,q_filtrado,r_filtrado,roll_filtrado,pitch_filtrado,y
aw_filtrado,VX_filtrado,VY_filtrado,VZ_filtrado,AIRSPEED_filtrado]=filtropasobajo(tiempo,p
resion_lon,presion_45,V_cc,V_AOA,V_AOS,ax,ay,az,p,q,r,roll,pitch,YAW,VX,VY,VZ,AIRSPEED);
38.
39.
40.
41. %% FILTRO DE KALMAN EXTENDIDO
42. dt=0.001;
43. g=9.81;
44.
45. mu=[0 0 0]; %valores iniciales m/s velocidad body 3 ejes
46. sigma=diag([1 1 1]*1); %matriz p de covarianza inicial
47. % var_c=0.1^2;
48. R=0.5^2; %ruido de la medición;
49. Q=diag([1 1 1]*1^2);
50. %Q_variance=1^2;%varianza del ruido del proceso
51. %Q=piecewise_white_noise(3,Q_variance,dt); %matriz Q que se ocupara en algoritmo
52.
53.
54. estimaciones=zeros(length(tiempo),3);
55. for i=1:length(tiempo)

```

```

56.     [mu_apriori, sigma_apriori] = prediccion_ode45terreno(mu,
    sigma,Q,p_filtrado(i),q_filtrado(i),r_filtrado(i),ax_filtrado(i),ay_filtrado(i),az_filtrado(i),dt,g,pitch_filtrado(i),roll_filtrado(i),yaw_filtrado(i));
57.     [mu, sigma,K] =
    actualizacion_datosterreno(mu_apriori,sigma_apriori,AIRSPEED_filtrado(i),VX_filtrado(i),VY_filtrado(i),VZ_filtrado(i),R,pitch_filtrado(i),roll_filtrado(i),yaw_filtrado(i),AOA_pseudo(i));
58.     estimaciones(i,:)=mu;
59.
60. end
61.
62.
63. estimacion_u=-estimaciones(:,1);
64. estimacion_v=estimaciones(:,2);
65. estimacion_w=estimaciones(:,3);
66.
67. cc=estimacion_w./estimacion_u;
68. AOA_estimacion=atand(cc);
69.
70. figure(1)
71. hold on
72. grid on
73. %plot(tiempo,AOA)
74. plot(tiempo,estimacion_u,'b','LineWidth',1.5)
75. plot(tiempo,estimacion_v,'r','LineWidth',1.5)
76. plot(tiempo,estimacion_w,'g','LineWidth',1.5)
77. xlabel('Tiempo [s]',FontSize=13)
78. ylabel('Velocidades lineales en el cuerpo [m/s]',FontSize=13)
79. legend('u','v','w',fontsize=12)
80. title('Velocidad estimada en el marco de referencia del cuerpo',FontSize=16)
81. ylim([-80 100])
82. xlim([275 500])

```

Código función de predicción:

```

1. function [mu_apriori, sigma_apriori] = prediccion_ode45terreno(mu, sigma, Q, p, q, r, ax,
    ay, az, dt, g, pitch, roll, yaw)
2.     % Función que define la ecuación diferencial para ode45
3.     ode_func = @(t, z) [-q * z(3) + r * z(2) - g * sind(pitch) + (ax);
4.         -r * z(1) + p * z(3) + g * sind(roll) * cosd(pitch) + ay;
5.         -p * z(2) + q * z(1) + g * cosd(roll) * cosd(pitch) + (az);
6.     ];
7.
8.     % Condiciones iniciales
9.     z0 = mu';
10.
11.     % Resuelve la ecuación diferencial con ode45
12.     [t, Z] = ode45(ode_func, [0 dt], z0);
13.
14.     % Obtiene el resultado en el tiempo dt
15.     mu_apriori = Z(end, :);
16.
17.     % Calcula la matriz Jacobiana en el último estado estimado
18.     Aa = [0, r, -q; -r, 0, p; q, -p, 0];
19.
20.     % Actualiza la covarianza usando la matriz Jacobiana
21.     F = eye(length(mu)) + (Aa * dt) + (Aa^2 * dt^2) / factorial(2);
22.     sigma_apriori = F * sigma * F' + Q;
23. end

```

Código función de actualización:

```

1. function
   [mu,sigma,K]=actualizacion_datosterreno(mu_apriori,sigma_apriori,AIRSPEED_filtrado,VX_filtrado,
   VY_filtrado,VZ_filtrado,R,pitch_filtrado,roll_filtrado,yaw_filtrado,AOA_pseudo)
2.
3.   u_apriori=mu_apriori(1);
4.   v_apriori=mu_apriori(2);
5.   w_apriori=mu_apriori(3);
6.
7.   %matriz de rotacion de body al inercial%
8.   Q_11=(cosd(yaw_filtrado)*cosd(pitch_filtrado));
9.   Q_12=(-
   sind(yaw_filtrado)*cosd(roll_filtrado)+cosd(yaw_filtrado)*sind(pitch_filtrado)*sind(roll_f
   iltrado));
10.  Q_13=(sind(yaw_filtrado)*sind(roll_filtrado)+cosd(yaw_filtrado)*sind(pitch_filtrado)*cosd
   (roll_filtrado));
11.  Q_21=(sind(yaw_filtrado)*cosd(pitch_filtrado));
12.  Q_22=(cosd(yaw_filtrado)*cosd(roll_filtrado)+sind(yaw_filtrado)*sind(roll_filtrado)*sind(
   pitch_filtrado));
13.  Q_23=(-
   sind(roll_filtrado)*cosd(yaw_filtrado)+sind(yaw_filtrado)*sind(pitch_filtrado)*cosd(roll_f
   iltrado));
14.  Q_31=-sind(pitch_filtrado);
15.  Q_32=cosd(pitch_filtrado)*sind(roll_filtrado);
16.  Q_33=cosd(pitch_filtrado)*cosd(roll_filtrado);
17.
18.  residual=[AIRSPEED_filtrado VX_filtrado VY_filtrado VZ_filtrado AOA_pseudo]'-
   [sqrt(u_apriori^2+v_apriori^2+w_apriori^2);(Q_11*u_apriori+Q_12*v_apriori+Q_13*w_apriori);
   (Q_21*u_apriori+Q_22*v_apriori+Q_23*w_apriori);(Q_31*u_apriori+Q_32*v_apriori+Q_33*w_apriori);
   atand(w_apriori/u_apriori)];
19.  H=[(u_apriori/sqrt(u_apriori^2+v_apriori^2+w_apriori^2))
   (v_apriori/sqrt(u_apriori^2+v_apriori^2+w_apriori^2))
   (w_apriori/sqrt(u_apriori^2+v_apriori^2+w_apriori^2));Q_11 Q_12 Q_13;Q_21 Q_22 Q_23;Q_31
   Q_32 Q_33;-w_apriori/(u_apriori^2*((1+w_apriori^2)/u_apriori^2)) 0
   1/(u_apriori*((1+w_apriori^2)/u_apriori^2))];
20.  K=(sigma_apriori*H')*inv(H*sigma_apriori*H'+R);
21.  mu=mu_apriori+K*(residual);
22.  sigma=(eye(3)-K*H)*sigma_apriori;%*([eye(3)-K*H]')+K*R*K';
23. end

```

Anexo E: Lista Pre-vuelo

La siguiente checklist fue utilizada para comprobar que el pre-vuelo de la aeronave fuera ejecutado correctamente, donde en cada paso se debe corroborar lo mencionado. En el cuadro se debe colocar un ok para hacer mención a que se realizó correctamente o una x en el caso contrario

Batería cargada 100%	
Batería radio RC cargada 100%	
Motor asegurado	
Conexión entre el motor y ESC respetando el orden de colores	
Conexión ESC y PM07	
Conexión PM07 y Arduino	
Conexión telemetría y Arduino	
Conexión antena de telemetría y Pixhawk	
Conexión y fijación de sensores Ms4525do	
Conexión mangueras de sensores Ms4525do	
Conexión de alimentación Power 1 y Power 2 en Pixhawk	
Conexión receptor y Pixhawk	
Conexión brújula y Pixhawk	
Conexión FMW PM07 (servos) y Pixhawk	
Conexión sensor de presión diferencial mediante I2C y Pixhawk (Pitot)	
Verificar tarjeta de memoria Pixhawk	
Caja y Pixhawk asegurados	
Verificar fijación de receptor y antena	
Verificar conexión de servo elevador a canal 1	
Verificar conexión de servo alerón a canal 2	
Verificar conexión de servo motor a canal 3	
Verificar conexión de servo timón a canal 4	
Conexión de veletas pivotantes con Arduino	
Hélice sin daños ni grietas y motor sin contaminaciones	
Hélice instalada y asegurada	
Fuselaje y recubrimientos en buenas condiciones	
Bisagras de superficies de control en buenas condiciones	
Cuernos y servos en buenas condiciones	
Spinner instalado y asegurado	
Tren de aterrizaje en buenas condiciones	
Conexión de cable de alimentación general	
Prueba de comunicación entre Pixhawk y Mission Planner	
Prueba de comunicación entre Radio RC y receptor	
Cerrar la nariz con todos sus tornillos	
Fijar el ala con sus respectivos pernos (2)	
Servos centrados al conectar el radio	

Servos centrados al conectar el radio	
Movimiento de servos con RC en direcciones correctas	
Prueba de potencia de motor	
Las condiciones climáticas son aceptables para el despegue	
Posición del avión con viento en contra	
Pista libre	
Listo para el despegue	

Anexo F: Otros:

En la Figura anexo 8 se presenta los sensores de presión diferencial instalados al interior del fuselaje.

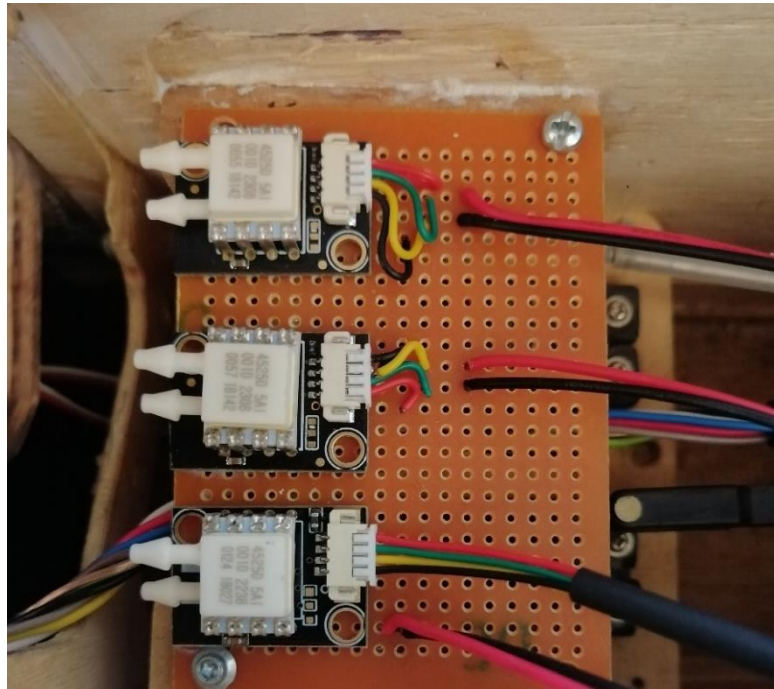


Figura anexo 8: Sensores de presión diferencial instalados.

Figura anexo 9: centro de gravedad de la aeronave.



En la figura anterior se muestra como se encuentra el centro de gravedad de la aeronave. La posición del centro de gravedad se debe respetar para tener vuelos estables. Se coloca un peso de 360 gramos en la nariz del avión para poder dejar el centro de gravedad a 90 milímetros desde el borde de ataque, valor recomendado por fabricante.



Figura anexo 10: Avion post preflight.



Figura anexo 11: Avión instantes antes del despegue (piloto José Abascal).