



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FIN DE MÁSTER

Máster en Ingeniería Mecatrónica

Universidad Politécnica de Valencia

Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero

Autor: Diego Aldair Chango Figueroa

Tutor : Vicente Fermín Casanova Calvo

Departamento: Ingeniería de Sistemas y Automática



Septiembre 2021

RESUMEN

En el presente Trabajo Fin de Máster se muestra el desarrollo del diseño e implementación de un mecanismo móvil. Dicho mecanismo es capaz de leer el movimiento de vuelo de un cuadricóptero y transferir dicha información de forma correcta a un programa externo para la simulación del vuelo en tiempo real.

Las piezas principales que forman el mecanismo han sido diseñadas mediante el programa de *SolidWorks* y fabricadas con la tecnología de impresión 3D. Además, se ha realizado una etapa previa de modelado y simulación del mecanismo, mediante la librería *Simscape Multibody* accesible desde el entorno de simulación de Simulink de *Matlab*, para asegurar y comprobar su comportamiento dinámico antes de su implementación real.

El mecanismo cuenta una plataforma sobre la que se acopla el cuadricóptero de estudio y 3 grados de libertad sensorizados que recogen los movimientos correspondientes al alabeo, cabeceo y guiñada. Posteriormente, la información de vuelo se transfiere al entorno de *Simulink* del programa de *Matlab* y se realiza una recreación del vuelo fidedigna en tiempo real sin necesidad de disponer de un espacio abierto.

Palabras clave TFM:

Simscape Multibody; Sensor movimiento angular; Actitud cuadricóptero; Simulador de vuelo.

RESUM

En el present Treball Fi de Màster es mostra el desenvolupament del disseny i implementació d'un mecanisme mòbil. Aquest mecanisme és capaç de llegir el moviment de vol d'un quadricòpter i transferir aquesta informació de manera correcta a un programa extern per a la simulació del vol en temps real.

Les peces principals que formen el mecanisme han sigut dissenyades mitjançant el programa de *SolidWorks* i fabricades amb la tecnologia d'impressió 3D. A més, s'ha realitzat una etapa prèvia de modelatge i simulació del mecanisme, mitjançant la llibreria *Simscape Multibody* accessible des de l'entorn de simulació de Simulink de *Matlab*, per a assegurar i comprovar el seu comportament dinàmic abans de la seua implementació real.

El mecanisme compta una plataforma sobre la qual s'acobla el quadricòpter d'estudi i 3 graus de llibertat sensoritzat que recullen els moviments corresponents al balanceig, capcineig i guinyada. Posteriorment, la informació de vol es transfereix a l'entorn de *Simulink* del programa de *Matlab* i es realitza una recreació del vol fidedigna en temps real sense necessitat de disposar d'un espai obert.

Paraules clau TFM:

Simscape Multibody; Sensor moviment angular; Actitud quadricòpter; Simulador de vol.

ABSTRACT

This Master's Thesis shows the development of the design and implementation of a mobile mechanism. This mechanism is able to read the flight movement of a quadricopter and transfer this information correctly to an external program for the simulation of the flight in real time.

The main parts of the mechanism have been designed using SolidWorks software and manufactured using 3D printing technology. In addition, a previous stage of modelling and simulation of the mechanism has been carried out using the Simscape Multibody library, accessible from the Simulink simulation environment of Matlab, to ensure and check its dynamic behaviour before its real implementation.

The mechanism has a platform on which the quadcopter is attached and 3 sensorised degrees of freedom that read the movements corresponding to roll, pitch and yaw. Subsequently, the flight information is transferred to the Simulink environment of the Matlab program, and a reliable recreation of the flight is carried out in real time without the need for an open space.

Palabras clave TFM:

Simscape Multibody; Angular motion sensor; Attitude quadcopter; Flight simulator.

DOCUMENTOS DEL TFM

- DOCUMENTO I MEMORIA DESCRIPTIVA
- DOCUMENTO II PRESUPUESTO
- DOCUMENTO III PLIEGO DE CONDICIONES
- DOCUEMNTO IV PLANOS

ÍNDICE DE LA MEMORIA DESCRIPTIVA

1. Introducción.....	14
1.1 Motivación del TFM.....	14
1.2 Objetivo	14
1.3 Aplicación.....	15
1.4 Ámbito del proyecto	15
2. Consideraciones Teóricas	16
2.1 Ángulos de navegación.....	16
2.2 Disposición y configuración de las hélices de un cuadricóptero.....	17
2.3 Relación de las velocidades angulares	17
3. Metodología.....	20
4. Diseño del mecanismo.....	22
4.1 Descripción breve del mecanismo	22
4.2 Piezas de CAD 3D.....	23
4.2.2 Piezas diseñadas 3D.....	23
4.2.2 Componentes escogidos en 3D.....	25
4.3 Modelo en 3D	26
4.4 Componentes escogidos	27
4.4.1 Componentes electrónicos	27
4.4.2 Componentes mecánicos	30
5. Simulación del mecanismo en Matlab.....	32
5.1 Configuración del entorno de <i>Simulink</i>	32
5.2 Exportación del modelo 3D.....	35
5.2.1 Bloques de Simscape Multibody	35
5.2.2 Construcción del mecanismo.....	38
5.3 Condiciones de fuerzas	41
5.4 Condiciones de velocidad de las hélices.....	44

5.5	Introducción de velocidades	47
5.6	Resultados de la simulación	49
6.	Implementación del sistema real	58
6.1	Montaje mecánico.....	59
6.2	Esquema eléctrico.....	61
6.2.1	Etapa de acondicionamiento	61
6.2.2	Esquema de conexión	62
6.3	Prototipo de implementación.....	64
6.4	Comunicación serie	65
6.4.1	Entorno Arduino	66
6.4.2	Entorno Simulink.....	68
6.5	Simulador de vuelo	69
6.5.1	Modelo del cuadricóptero implementado	69
6.5.2	Implementación de los ejes de rotación y desplazamiento.....	70
6.5.3	Fuerzas que intervienen en el movimiento	71
6.5.4	Entorno del simulador de vuelo.....	72
7.	Comprobaciones de la implementación.....	74
7.1	Lectura de los sensores	74
7.2	Movimiento mecánico del mecanismo	75
7.3	Movimiento real y simulado.....	75
8.	Resultados.....	78
9.	Conclusiones y recomendaciones	82
10.	Bibliografía.....	84
11.	Anexos	85
11.1	Código de implementación Arduino	85
11.2	Vista general Simulinks.....	87
11.3	Ficha técnica sensor 6120 Series	90

ÍNDICE PRESUPUESTO

1. Costes del Hardware.....	94
2. Costes del Software	94
3. Coste de los Recursos Humanos.....	95
4. Coste Total del Proyecto.....	95

ÍNDICE PLIEGO DE CONDICIONES

1. Definición y Alcance del Pliego	97
2. Condiciones Generales y Normativa	97
3. Condiciones de Carácter Económico.....	98
4. Especificaciones de Ejecución.....	98

ÍNDICE DE PLANOS

Plano 1: Ensamblado explosionado del mecanismo.....	101
Plano 2: Marco exterior	101
Plano 3: Marco interior.....	101
Plano 4: Plataforma	101
Plano 5: Ejes	101
Plano 6: Pilares	101
Plano 7: Base	101

ÍNDICE FIGURAS DE LA MEMORIA DESCRIPTIVA

Figura 1: Ángulos de navegación cuadricóptero.....	16
Figura 2: Disposición y sentido de giro de las hélices de un cuadricóptero tipo “X”...	17
Figura 3: Movimiento de un cuadricóptero.....	18
Figura 4: Diagrama de flujo de la metodología.....	20
Figura 5: Modelo 3D del prototipo	26
Figura 6: Movimientos de la plataforma.....	27
Figura 7: Combinación de movimientos	27
Figura 8: Potenciómetro magnético TT Electronics de la serie 6127	28
Figura 9: Placa de Arduino DUE	29
Figura 10: Eje del prototipo	30
Figura 11: Rodamientos del prototipo.....	31
Figura 12: Configuración del Solver de Simulink	33
Figura 13: Configuración del Simulation Pacing	33
Figura 14: Bloques de configuración de la simulación	34
Figura 15: Propiedades del bloque Solid	35
Figura 16: Propiedades del bloque Rigid Transformation	36
Figura 17: Propiedades del bloque Revolute Joint.....	37
Figura 18: Propiedades del bloque External Force and Torque	38
Figura 19: Propiedades del bloque Simulink-PS Converter	38
Figura 20: Simulink bloques de unión entre Base y Pilares.....	39
Figura 21: Simulink bloques del Marco exterior y componentes	40
Figura 22: Simulink bloques de unión entre Marco interior y Plataforma.....	40
Figura 23: Simplificación cuerpo sólido del cuadricóptero	41
Figura 24: Simulink de la implementación del cuadricóptero y sus fueras.....	42
Figura 25: Puntos de aplicación de las fuerzas lineales	43
Figura 26: Identificación de hélices y movimientos del cuadricóptero en la simulación	44
Figura 27: Simulink de la implementación de las expresiones de velocidad angular ...	47
Figura 28: Controles usados del mando	48
Figura 29: Simulink para la lectura y consigna de las velocidades.....	48
Figura 30: Captura simulación mecanismo 1 Cabeceo	56
Figura 31: Captura simulación mecanismo 2 Alabeo	56
Figura 32: Captura simulación mecanismo 3 Guiñada	57

Figura 33: Divisor resistivo.....	61
Figura 34: Esquema eléctrico de conexión	62
Figura 35: Mecanismo real con 3 ejes de libertad.....	64
Figura 36: Conexiones implementadas	65
Figura 37: Diagrama de flujo del código de implementación	66
Figura 38: Diagrama de flujo de la Comunicación serie.....	67
Figura 39: Simulink de la comunicación serie	68
Figura 40: Simulink del modelo del cuadricóptero.....	69
Figura 41: Cuadricóptero de simulación	70
Figura 42: Simulink de las articulaciones	70
Figura 43: Simulink en la implementación de fuerzas	71
Figura 44: Entorno de simulación y perspectivas.....	73
Figura 45: Movimeinto de Alabeo	76
Figura 46: Movimiento de Cabeceo	77
Figura 47: Movimiento de Guiñada	77
Figura 48: Video demo Secuencia 1	78
Figura 49: Video demo Secuencia 2	79
Figura 50: Video demo Secuencia 3	79

ÍNDICE GRÁFICAS DE LA MEMORIA DESCRIPTIVA

Gráfica 1: Seguimiento Cabeceo	50
Gráfica 2: Velocidades angulares Cabeceo	51
Gráfica 3: Seguimiento Alabeo	52
Gráfica 4: Velocidades angulares Alabeo	53
Gráfica 5: Seguimiento Guiñada	54
Gráfica 6: Velocidades angulares Guiñada	55
Gráfica 7: Video demo ángulos de Alabeo y Cabeceo.....	80
Gráfica 8: Video demo ángulo de guiñada.....	80
Gráfica 9: Video demo trayectoria descrita.....	81

DOCUMENTO I

MEMORIA DESCRIPTIVA

1. INTRODUCCIÓN

1.1 Motivación del TFM

La idea de realizar este proyecto surge por la motivación de implementación un sistema mecatrónico real y funcional. En este proceso se pretenderá abarcar todas las fases: concepción de la idea, diseño, simulación, construcción e implementación.

Asimismo, se pondrán a prueba los conocimientos adquiridos de las distintas ingenierías como son la mecánica, informática, control y electrónica durante el Máster de Ingeniería Mecatrónica en la realización de este proyecto. Concretamente se usarán conocimientos de las asignaturas **Dinámica de Sistemas Mecánicos** para el diseño de la estructura, **Sistemas de Medición y Actuación** para escoger los sensores y **Control Aplicado a Sistemas Mecatrónicos** para la simulación del modelo y su implementación.

Para ello, se ha buscado una idea nueva y de alguna forma desarrollar un prototipo de algo no existente cuyo objetivo principal es que sea funcional y abra puertas a nuevos proyectos que persigan la misma finalidad.

1.2 Objetivo

El objetivo del presente Trabajo Fin de Máster es el diseño e implementación de un mecanismo móvil con 3 grados de libertad sensorizados. Esta deberá ser capaz de leer el movimiento de vuelo de un cuadricóptero que se le acople encima y transferir esta información a un programa externo para la simulación de vuelo en tiempo real.

Concretamente se busca la capacidad de la estructura de leer de la forma más real posible el vuelo del cuadricóptero que se le posicione encima de la plataforma principal. Con ello, se pretende realizar simulaciones más “reales” del vuelo del cuadricóptero, ya que se llevaría a cabo mediante la lectura de los sensores físicos que incorpora el mecanismo. Y por medio de este sistema se podría crear una simulación de vuelo fidedigna y en tiempo real sin la necesidad de disponer un espacio abierto.

Siendo el mecanismo un conjunto de elementos mecánicos, componentes electrónicos y una interfaz se marcan a cumplir las siguientes condiciones:

- Mecanismo articulado que no dificulte o entorpezca el vuelo de un cuadricóptero mediante un cuerpo ligero y articulaciones con un bajo rozamiento.

- Asegurar una lectura estable de los ángulos de navegación: cabeceo, alabeo y guiñada.
- Mostar en tiempo real el movimiento del cuadricóptero.

1.3 Aplicación

La aplicación real surge de la necesidad de una alternativa de la lectura de la actitud del cuadricóptero a los sensores propios que cada uno incorpora y que generalmente no se tiene acceso de forma directa mientras estos están en funcionamiento o carecen de estos por ser drones de bajo presupuesto. Aún en el caso en el que se tenga acceso a dichos sensores durante el vuelo surge otro problema: el ruido.

Generalmente los sensores usados para el control de vuelo de los cuadricópteros como son los giróscopos o magnetómetros son especialmente sensibles al ruido generado por los motores en marcha, dificultando así su lectura. Por eso, mediante este mecanismo independientemente de que se disponga acceso o no de dichos sensores serán capaces de leer la actitud del cuadricóptero sin verse afectado por el ruido.

Por ello, una de sus posibles aplicaciones serían la de una plataforma experimental para comprobar y calibrar el vuelo de un dron o un prototipo de este. Se detectaría cualquier problema en el control del vuelo del dron en un entorno seguro donde no se pongan en peligro a las personas o al equipo.

Todo ello, se podría llevar a cabo sin verse afectado por el Real Decreto 1036/2017 que rige en nuestro país sobre la utilización civil de las aeronaves pilotadas por control remoto, ya que en ningún momento el dron va a abandonar la plataforma.

1.4 Ámbito del proyecto

El ámbito que engloba este proyecto abarca desde el diseño físico del mecanismo como es la elección de materiales y dispositivos electrónicos. Además, es necesario la programación de un dispositivo electrónico y la creación de una interfaz para visualizar y tratar la información procedente del mecanismo real, siendo un dispositivo totalmente funcional.

2. CONSIDERACIONES TEÓRICAS

Para entender la relación de movimiento entre la estructura diseñada y el movimiento de un cuadricóptero es necesario tener en cuenta algunas consideraciones teóricas: los ángulos de navegación y la relación del giro las hélices con el movimiento del cuadricóptero.

2.1 Ángulos de navegación

En aeronáutica es común usar los ángulos de Euler para describir la orientación del objeto móvil respecto a un sistema de referencia fijo y por medio de tres coordenadas angulares es posible saber la posición de dicho sistema móvil.

Concretamente se distinguen tres ángulos definidos a lo largos de los ejes de cuadricóptero como se indica en la siguiente figura:

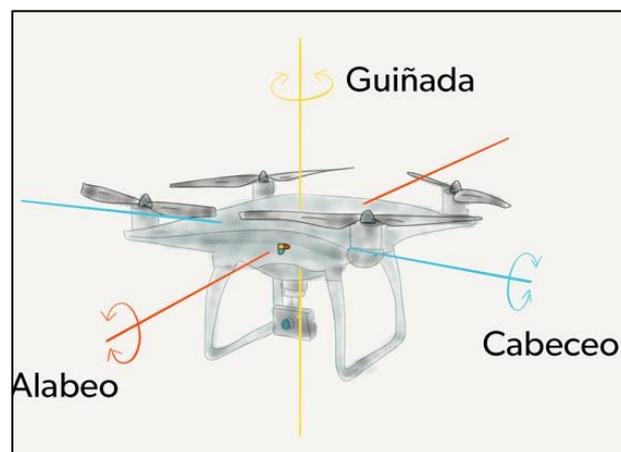


Figura 1: Ángulos de navegación cuadricóptero

- Cabeceo (Pitch) → Ángulo de inclinación definido sobre el eje transversal del cuerpo. Este indicará si el cuadricóptero se desplaza hacia delante o hacia atrás.
- Alabeo (Roll) → Ángulo de inclinación definido sobre el eje longitudinal del cuerpo. Este indicará si el cuadricóptero se desplaza a la derecha o izquierda, es decir, lateralmente.
- Guiñada (Yaw) → Ángulo de rotación definido sobre el eje vertical del cuerpo, es decir, perpendicular al eje transversal y longitudinal. Este indicará el ángulo de giro sobre el propio cuadricóptero.

De esta forma, usaremos estos ángulos para determinar la posición y el movimiento de nuestro cuadricóptero y serán las variables que será capaz de recoger nuestro mecanismo.

2.2 Disposición y configuración de las hélices de un cuadricóptero

Un cuadricóptero está compuesto por cuatro hélices equidistantes de su centro de masa. Según la disposición de estas, existen principalmente dos configuraciones: tipo “x” y “+”. Asimismo, dependiendo de la configuración la relación de las velocidades angulares y el movimiento son distintas. En nuestro caso se tomará como referencia la configuración tipo “X”, tal como se muestra en la siguiente figura:

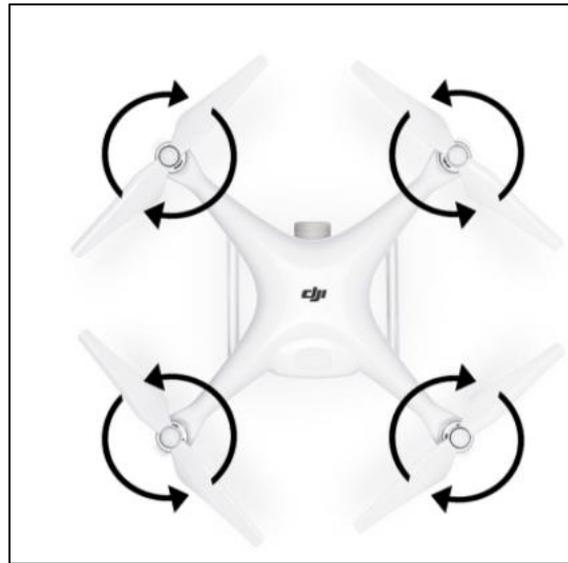


Figura 2: Disposición y sentido de giro de las hélices de un cuadricóptero tipo “X”

Como se observa las hélices están situadas de forma diagonal formando una ”X“, donde dos de ellas giran en sentido horario y las otras dos en sentido antihorario. Las que giran en el mismo sentido están opuestas diagonalmente entre ellas.

Gracias a esta disposición, cada hélice generará una fuerza de sustentación y un momento de fuerza resultante en el centro de masa del cuadricóptero que permitirá tanto el movimiento como el giro de este.

2.3 Relación de las velocidades angulares de las hélices y el movimiento del cuadricóptero

Una vez introducidos los ángulos de navegación y la configuración del dron es posible establecer la relación entre la disposición de las hélices y su velocidad angular que originan el movimiento del cuadricóptero.

En la Figura 3, se indican la numeración de referencia de cada motor, el sentido de giro de las hélices, sus velocidades y cómo se originan los ocho posibles movimientos por el espacio

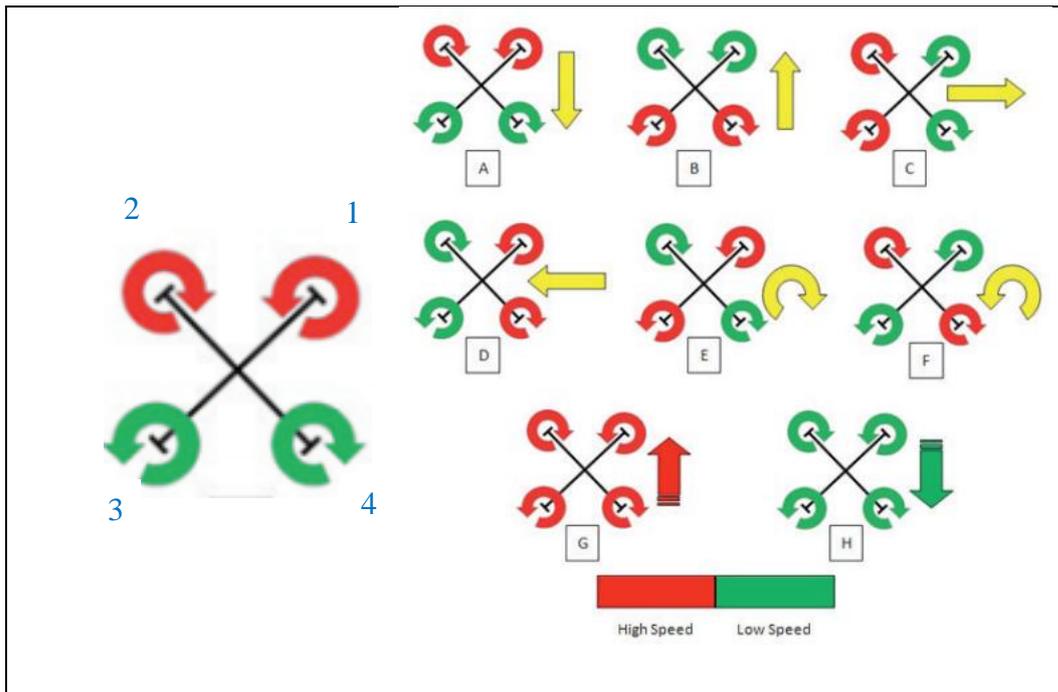


Figura 3: Movimiento de un cuadricóptero

De esta forma, dichos movimientos se agrupan en los cuatro tipos de movimiento que tiene un cuadricóptero:

(Describir en relación de la velocidad de los motores (https://www.edubcn.cat/rcs_gene/treballs_recerca/2015-2016-03-1-TR.pdf)

- Cabeceo (Pitch) → Da origen al movimiento de retroceso (A) o avance (B) del cuadricóptero. Para retroceder, los motores 1 y 2 giran a alta velocidad mientras que el motor 3 y 4 a lenta. Esto hace que se incline hacia atrás cambiando el ángulo de la fuerza de empuje, lo que produce el retroceso. En el avance se produce de forma análoga pero esta vez los motores que giran a alta velocidad son el 3 y 4 mientras que los motores 1 y 2 a lenta.
- Alabeo (Roll) → Da origen al movimiento de desplazamiento lateral derecha (C) o izquierda (D) del cuadricóptero. Para desplazarse a la derecha, los motores 2 y 3 giran a alta velocidad mientras que el motor 1 y 4 a lenta. Esto hace que se incline hacia la derecha cambiando el ángulo de la fuerza de empuje, lo que produce el desplazamiento a la derecha. El desplazamiento a la izquierda se produce de forma análoga pero esta vez los motores que giran a alta velocidad son el 1 y 4 mientras que los motores 2 y 3 a lenta.
- Guiñada (Yaw) → Da origen al giro horario (E) o antihorario (F) del cuadricóptero mediante la descompensación del momento resultante a favor de unos de estos

dos. Para el giro en sentido horario, los motores 1 y 3 que giran en sentido antihorario giran a alta velocidad y los motores 2 y 4 que giran en sentido horario a baja velocidad. El giro se mantendrá hasta mientras no se apliquen otras condiciones de velocidad a los motores. Para el giro horario se produce de forma análoga pero esta vez los motores que giran a alta velocidad son el 2 y 4 mientras que los motores 1 y 3 a lenta.

- Altitud (Throttle) → Da origen al ascenso (G) o descenso (H) del cuadricóptero. En este caso, los 4 motores giran a la misma velocidad. Se produce el ascenso cuando la velocidad de giro genera una fuerza de empuje mayor al peso del propio dron. Por otro lado, para el descenso los 4 motores giran a baja velocidad siendo la fuerza de empuje menor al peso de este.

Se observa como el desplazamiento por el espacio del cuadricóptero se rige por la velocidad angular de cada hélice. A su vez, cada hélice genera una fuerza de empuje que según la combinación entre estas es capaz de originar los movimientos de tipo cabeceo, alabeo y altitud.

Por otra parte, la suma de momentos de cada hélice da lugar a un único momento de fuerza resultante que cuenta con motores con sentidos opuestos dos a dos es capaz de anular el momento para estabilizar el cuadricóptero. Gracias a la descompensación voluntario del momento resultante en el centro de masa es posible controlar el momento de guiñada.

Por ello, si todas las hélices giran a la misma velocidad y la fuerza de sustentación es equivalente al peso del dron, este se mantendrá estable.

3. METODOLOGÍA

La metodología empleada para el éxito de los objetivos planteados en este proyecto se refleja en la siguiente figura:

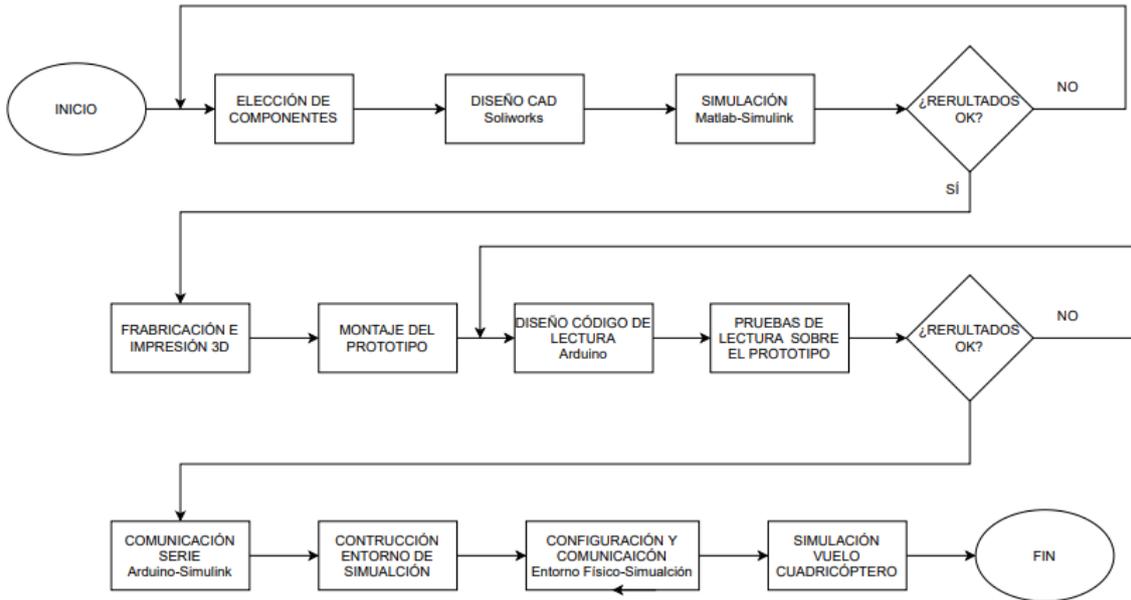


Figura 4: Diagrama de flujo de la metodología

En primer lugar, se eligen los componentes electrónicos y mecánicos que serán necesarios para la correcta lectura de los ángulos de inclinación de la plataforma, el giro de esta y que el movimiento de la estructura sea adecuado.

Una vez escogidos los componentes, se diseñará un modelo en 3D de la estructura en base a las medidas y características de dichos componentes. Este modelo será realizado mediante el programa de CAD SolidWorks. Finalmente se contará con un modelo CAD de la estructura y los componentes.

Seguidamente, para estudiar el comportamiento de la estructura en condiciones de movimiento se realizará mediante el entorno de Simulink que proporciona el programa de cálculo numérico MATLAB. En él se podrá apreciar si en el modelo es necesaria alguna modificación antes de su posterior fabricación.

Se comprobará que en la etapa de simulación no hay fallos y se procederá a la fabricación y montaje del prototipo con las piezas impresas de la estructura final y los componentes escogidos. Sobre este se realizarán pruebas físicas de movimiento y pruebas de vuelo para

constatar los resultados obtenidos en la simulación. Dichas pruebas de vuelo se realizarán con el cuadricóptero sobre la plataforma.

A continuación, se diseñará el programa para la lectura de los datos del mecanismo y se realizarán pruebas con el cuadricóptero encima. Se monitorizará los valores obtenidos en tiempo real de los ángulos y se ajustarán los sensores para que correspondan con los valores de ángulos deseados en el entorno de simulación.

Una vez, nos aseguremos que el movimiento corresponde con la lectura se establecerá la comunicación serie entre *Arduino* y *Simulink*, de tal forma que los datos de lectura estén disponibles desde dicho entorno en tiempo real.

Asimismo, se construirá un entorno de simulación con un cuadricóptero virtual que sea capaz de desplazarse por dicho entorno por medio de consignas de ángulos de vuelo correspondientes a los leídos por el mecanismo móvil.

Finalmente, se establecerá la comunicación entre el dispositivo físico y el entorno de simulación de vuelo mediante la comunicación serie, donde el cuadricóptero virtual sea capaz de reproducir los movimientos del real a través de los datos proporcionados por los sensores.

4. DISEÑO DEL MECANISMO

El diseño del mecanismo de este proyecto se basa en una estructura capaz de moverse solidariamente con el vuelo de un cuadricóptero. Para ello se han tenido en cuenta principalmente dos condiciones.

La primera de carácter funcional es que debe tener 3 grados de libertad que corresponden a los ángulos de navegación de cabeceo, alabeo y guiñada, explicados en el punto 2.1 de este proyecto. Estos ángulos deberán ser recogidos por algún sensor para poder tener a disposición dicha información.

Y la segunda de carácter mecánico, es que su diseño se realizará mediante impresión 3D. Por tanto, se tendrán en cuenta las limitaciones de esta tecnología como el tamaño de las piezas y perfiles o formas a usar que sean posibles de imprimir. Además, en el diseño se tendrán en cuenta las medidas de los componentes mecánicos y electrónicos escogidos previamente que facilitará su implementación.

4.1 Descripción breve del mecanismo

El mecanismo consta de una primera parte formado por un bastidor exterior, un bastidor interior y una plataforma. Los cuales mediante sensores de posición de efecto hall, y un par de ejes irán conectados entre sí, obteniendo 2 grados de libertad correspondientes al alabeo y cabeceo del dron.

Por otro lado, la segunda parte constará de una estructura en forma de pilares y una base hueca para almacenar el tercer sensor correspondiente a la lectura de la guiñada. Estas serán las encargadas de sujetar y permitir el giro de la parte superior del mecanismo sobre la base.

Asimismo, el dron se acoplará encima de la plataforma central cuyo movimiento de vuelo se transferirá a las articulaciones de la estructura y por tanto a los sensores que la componen. De esta forma, mediante los sensores dispuestos en los ejes de dichas articulaciones, se podrá tener acceso a la información del movimiento de vuelo.

A su vez, estos datos serán recogidos y tratados por un microcontrolador, concretamente, por ARDUINO y mediante este se transferirá a un programa de simulación en tiempo real en el entorno de Simulink de Matlab para la visualización y almacenamiento de los datos de vuelo.

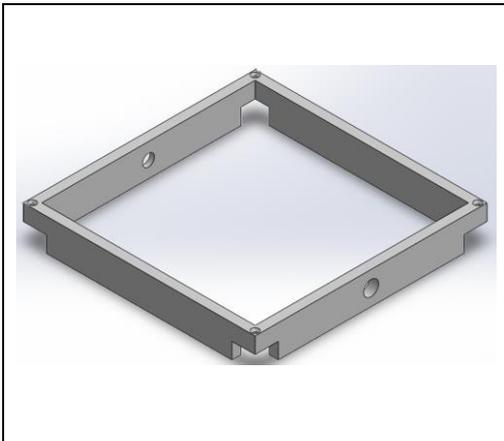
4.2 Piezas de CAD 3D

Para su diseño se ha utilizado el entorno de diseño CAD 3D de SolidWorks ya que uno de los más usados en la industria. Además, proporciona herramientas que nos permiten diseñar pieza a pieza el modelo y después ensamblarlo para establecer relaciones de posición entre las piezas y visualizar su movimiento.

La primera limitación del mecanismo es la plataforma de impresión. Esta cuenta con una superficie de 205 x 205mm, por lo que se el bastidor exterior se ha limitado a 200x200mm. Partiendo de esta base, se han ido adaptando las medidas a los componentes escogidos y siempre teniendo en cuenta no limitar el movimiento mecánicamente para que las piezas no choquen entre sí.

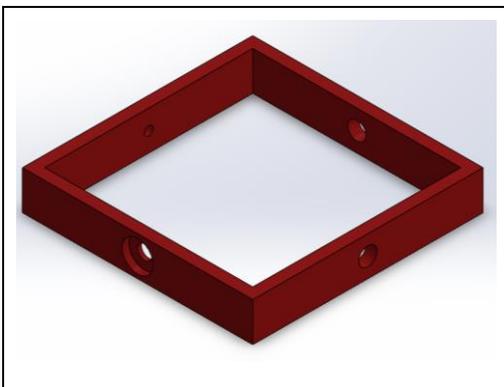
4.2.1 Piezas diseñadas 3D

Teniendo en cuenta estos puntos, las piezas de diseño propio para su posterior impresión han sido las siguientes:



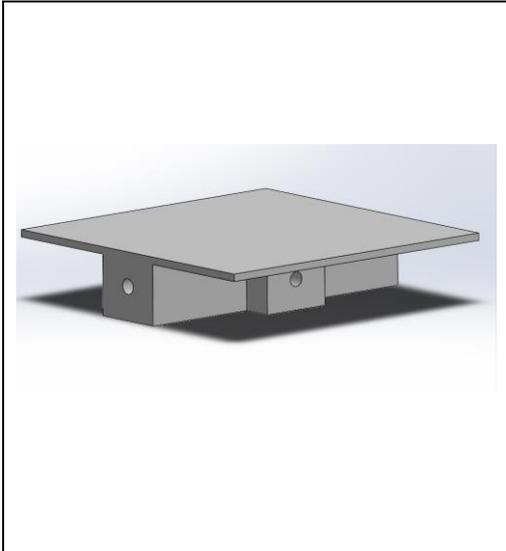
Bastidor exterior

Marco exterior de 200x200 mm que cuenta con dos cavidades en los laterales. En uno se alojará un rodamiento con su respectivo eje y en el otro parte de un potenciómetro magnético. Además, en las esquinas dispone de entradas que servirán para acoplar los pilares y de cuatro orificios para atornillarlos entre sí



Bastidor interior

Marco interior de 164x164 mm que permite el movimiento del mecanismo en el eje transversal para la lectura del ángulo de cabeceo. Para ello, cuenta con orificios donde se acoplará el sensor correspondiente y los rodamientos más los ejes.



Plataforma

Superficie cuadrada de 128x128 mm que permite el movimiento en el eje longitudinal para la lectura del ángulo de alabeo. Sobre este se situará y bloqueará el cuadricóptero, transmitiendo así el movimiento al resto de la estructura.

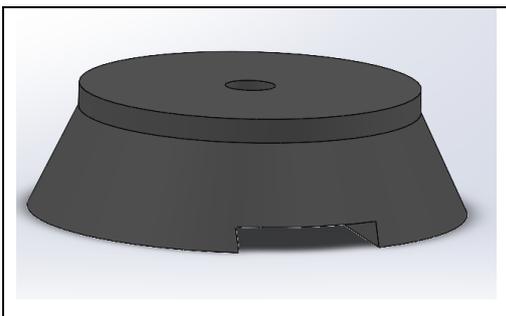
Además, cuenta con orificios para atornillar y poner contrapesos para estabilizar la plataforma.



Pilares

Son pilares que sirve de unión entre la base y el bastidor exterior. Esta estructura permitirá el movimiento transversal y longitudinal del bastidor interior y la plataforma, respectivamente sin producir choques.

En su base cuenta con un saliente donde se acoplará el eje del potenciómetro magnético que registrará la lectura del giro del mecanismo correspondiente al ángulo de guiñada del cuadricóptero.

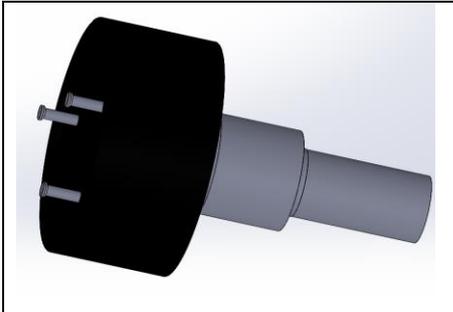


Base

Es la base donde descansará toda la estructura. Dispone de un orificio donde se situará el potenciómetro magnético de 360° que a su vez servirá de unión con los pilares.

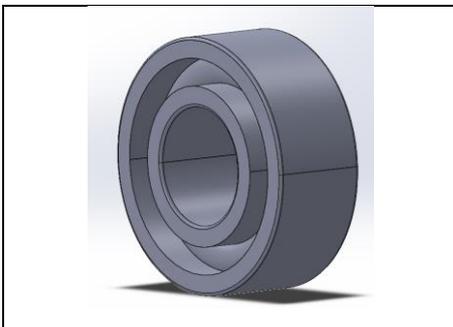
4.2.2 Componentes escogidos en 3D

Por otra parte, para la recreación del modelo se han modelado los componentes escogidos (explicados en el punto 4.4) que intervienen en el movimiento del mecanismo. Estos son los que hacen de eje transversal y longitudinal que permiten el alabeo y cabeceo y, de eje de rotación.



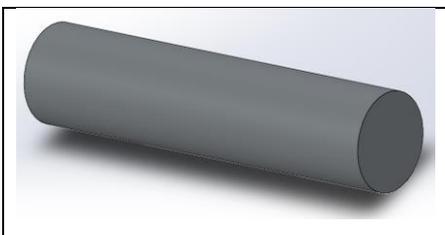
Potenciómetro magnético

Modelo CAD del potenciómetro magnético 6127 de “TT ELECTRONICS / BI TECHNOLOGIES”.



Rodamiento

Modelo CAD del rodamiento donde van acoplados los ejes con las medidas según la norma DIN 628.



Ejes

Modelo CAD del eje que permite el movimiento en el eje transversal y longitudinal

4.3 Modelo en 3D

El modelo 3D del conjunto del mecanismo ensamblado en SolidWorks se muestra en la Figura X. Los planos del montaje y las piezas diseñadas para su fabricación en 3D se adjuntan en los anexos.

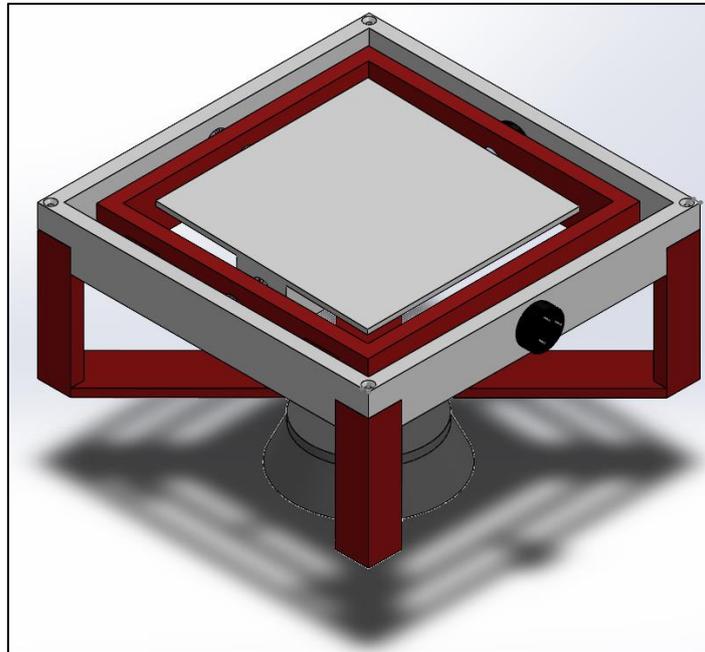


Figura 5: Modelo 3D del prototipo

Como se observa entre el bastidor exterior e interior, mediante el eje transversal formado por un eje + rodamientos y un potenciómetro, permitirá la lectura del cabeceo. De la misma forma, el bastidor interior y la plataforma, mediante el eje longitudinal constituido por un eje + rodamiento y un potenciómetro, permitirá la lectura del alabeo. Para la lectura del ángulo de guiñada, el sensor con lectura de 360° hace de único eje entre la base y los pilares que sujeta el resto del mecanismo. Los pilares, además de soporte, transmiten el movimiento de giro.

De esta forma, en la secuencia siguiente se muestra los movimientos posibles en SolidWorks por separado:

- Giro transversal
- Giro longitudinal
- Giro en guiñada

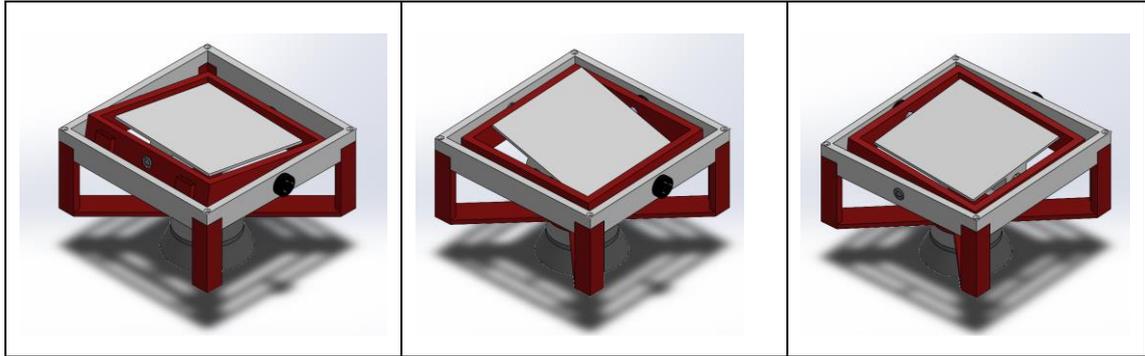


Figura 6: Movimientos de la plataforma

De la misma forma, la estructura es capaz de combinar los movimientos anteriores sin producirse un choque entre las partes mecánicas que forman el mecanismo.

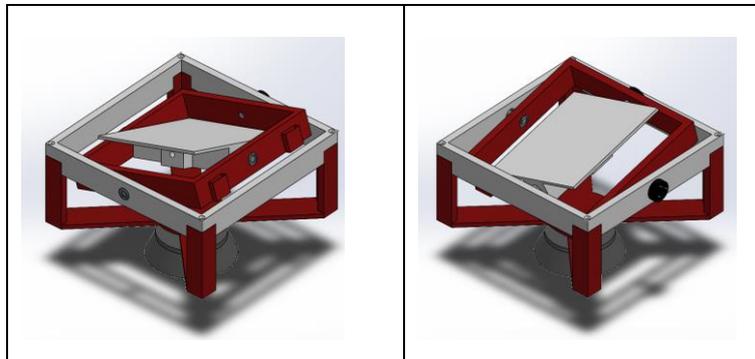


Figura 7: Combinación de movimientos

4.4 Componentes escogidos

El diseño del mecanismo se realiza en base a los componentes que satisfacen nuestras necesidades y compatibles físicamente con este. Dichos componentes se dividen en componentes mecánicos y electrónicos.

4.4.1 Componentes electrónicos

Los dispositivos electrónicos escogidos que se usarán en la implementación del prototipo real son:

Sensor de posición angular

Es un dispositivo electrónico capaz de transformar los cambios de posición angular en una magnitud eléctricamente medible. Entre las distintas tecnologías usadas para estos sensores se ha optado por los de efecto Hall.

La tecnología de los sensores de efecto Hall, se basa en la inyección de corriente a un conductor o semiconductor el cual está bajo los efectos de un campo magnético

produciendo una tensión. Dicha tensión, también llamada tensión de Hall es proporcional al campo magnético del que se ve influenciado.

Por ello, al tratarse de un sensor de posición angular con efecto Hall se denomina potenciómetro magnético. Siendo la tensión de salida proporcional al giro gracias a la existencia de un imán permanente (fuente de campo magnético) que puede girar alrededor del sensor de efecto Hall estático.

Concretamente, se ha optado por la utilización del sensor de “TT ELECTRONICS / BI TECHNOLOGIES” de la serie 6127 representado en la siguiente Figura



Figura 8: Potenciómetro magnético TT Electronics de la serie 6127

El rango de movimiento de alabeo y cabeceo se ha establecido en $\pm 45^\circ$ y el de guiñada de 360° . Se ha optado por usar 2 sensores distintos para ajustarse a las necesidades del proyecto y no desaprovechar las características que ofrece.

Las características principales de los sensores escogidos son las siguientes:

Modelo	6127V1A90L.5	6127V1A360L.5 FS
Uso	Lectura ángulo de Cabeceo/Alabeo	Lectura ángulo de Guiñada
Voltaje de alimentación	5V	5V
Voltaje de salida	0-5V (disponible 4%-98%)	0-5V (disponible 4%-98%)
Recorrido	90°	360°
Peso 170	170g	170°

Placa Arduino DUE

Arduino es una plataforma de creación de electrónica basado en hardware y software libre. Existen varios modelos de placas con características distintas que se ajustan a las necesidades de cada proyecto. Pero en todas se trabaja en el mismo entorno de programación llamado Arduino IDE (Entorno de Desarrollo Integrado).

En nuestro caso, la placa de *Arduino* a usar es *Arduino DUE* mostrada en la Figura X. Dicha placa cuenta con pines de entrada analógica de 3,3 V que no coinciden con las tensiones de salida de los potenciómetros magnéticos (5V). De esta forma, es necesario diseñar una etapa para ajustar el rango de tensiones a la entrada y así no poder bits de resolución y tener acceso a todo el rango de información de los sensores.

Además, cuenta con dos puertos de conexión: el puerto de programación y el puerto USB Nativo. Por el puerto de programación, se carga el código de programación a implementar y además es posible tener acceso a las variables por medio del *Monitor Serie* y *Serial Plotter*. El puerto USB Nativo, se usa en la aplicación para poder establecer la comunicación serie y poder transmitir las lecturas de los potenciómetros magnéticos a *Simulink*.

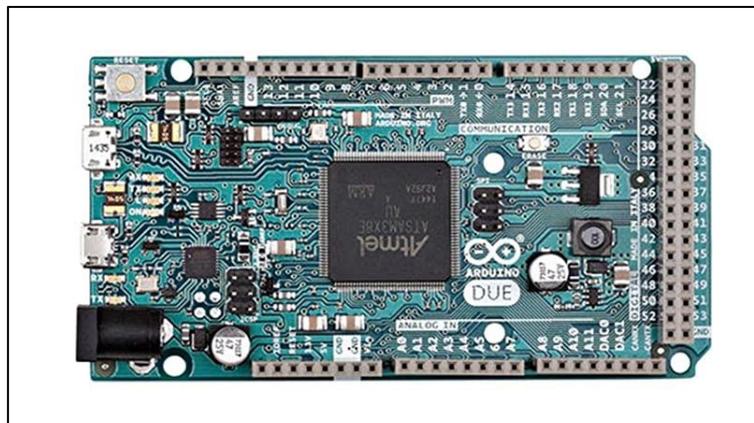


Figura 9: Placa de Arduino DUE

Las características principales se recogen a continuación:

Microcontrolador	32 bit ARM
Voltaje de operación	7 V – 12 V
Voltaje operativo	3,3 V
Pines de entradas/salidas digitales	54 (12 son salidas PWM)
	pines

Pines de entradas analógicas	12 pines
Pines de salidas analógicas	2 (DAC)
Memoria Flash	512 kB
SRAM	96 kB
Velocidad del Reloj	84 MHZ

4.4.2 Componentes mecánicos

Los componentes mecánicos usados son los siguientes:

Ejes

El eje es un elemento cilíndrico que sirve de centro para permitir el movimiento de giro. En este caso, se ha optado por el uso de una varilla lisa de acero inoxidable y cortada a medida.



Figura 10: Eje del prototipo

Para nuestra aplicación serán necesarios 2, ya que junto con los ejes de los potenciómetros magnéticos permitirán el movimiento de alabeo y cabeceo de la plataforma. El diámetro escogido es similar al eje del potenciómetro magnético, ya que de esta forma actuarán de como un único eje transversal y longitudinal.

Sus medidas son:

Diámetro	6 mm
Longitud	27 mm

Rodamientos

Los rodamientos están formados por cilindros concéntricos entre los que van situados un juego de bolas que pueden girar libremente. Estarán dispuestos dentro de la estructura del prototipo donde descansarán los extremos de los ejes



Figura 11: Rodamientos del prototipo

Su función es disminuir el rozamiento para permitir un movimiento más fluido entre los ejes y los bastidores de la plataforma.

Sus medidas son:

Ancho	5 mm
Diámetro exterior	13mm
Diámetro interior	6 mm

5. SIMULACIÓN DEL MECANISMO EN MATLAB

Para la visualización y simulación del comportamiento del movimiento del mecanismo se hace mediante la librería *Simscape Multibody* accesible desde el entorno de simulación de *Simulink* de *Matlab*.

Simscape Multibody proporciona un entorno de simulación de múltiples cuerpos para sistemas mecánicos en 3 dimensiones. En este caso, las piezas son exportadas del modelo en *SolidWorks* previamente diseñadas una a una configurándolas con sus respectivas características como masa e inercia.

Para ello, se disponen de bloques que representan los cuerpos o piezas que conforman el mecanismo y se establecen las uniones y restricciones entre sí. Además, se dispone de sensores para la lectura de variables y elementos de fuerza para establecer las condiciones de funcionamiento deseadas.

De esta forma, con todos los datos proporcionados *Simscape Multibody* formula y resuelve las ecuaciones de movimiento para el sistema mecánico completo.

Una de las grandes ventajas que proporciona este entorno es la capacidad de la simulación en tiempo real. Por lo que, se pueden realizar modificaciones en el modelo y visualizar de forma inmediata los cambios en nuestro sistema.

5.1 Configuración del entorno de *Simulink*

En primer lugar, se establecen las condiciones de simulación del entorno de *Simulink*. Para ello, se configura el *solver* de *Simulink* de la siguiente forma:

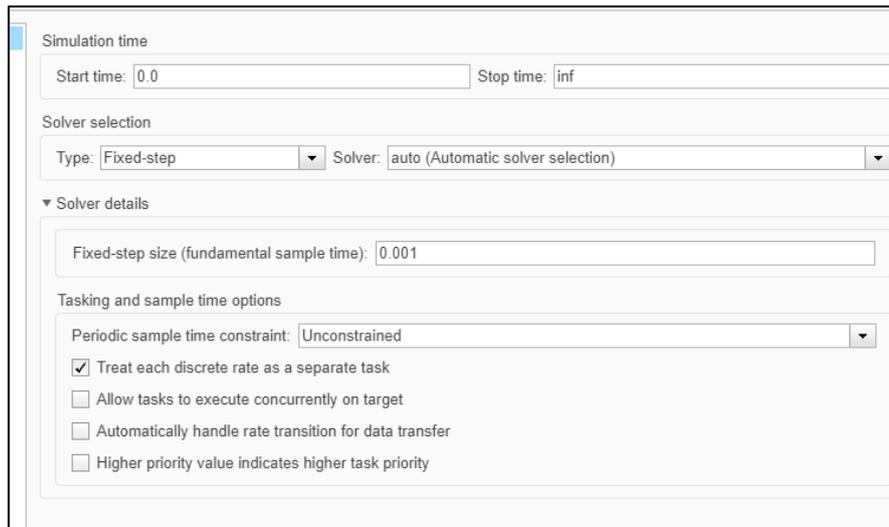


Figura 12: Configuración del Solver de Simulink

Se establece un paso fijo con un tiempo de muestreo de 1ms capaz de ofrecer los resultados de la simulación en tiempo real. Al establecer un tiempo de paso fijo, se asegura que la simulación siga el tiempo de muestreo establecido entre cada instante de la simulación. Asimismo, para que las unidades de tiempo marcado en la simulación correspondan con el tiempo real es necesario establecer que dicho tiempo sea equivalente al tiempo real por medio de la opción de *Simulation Pacing* y establecerla en 1.

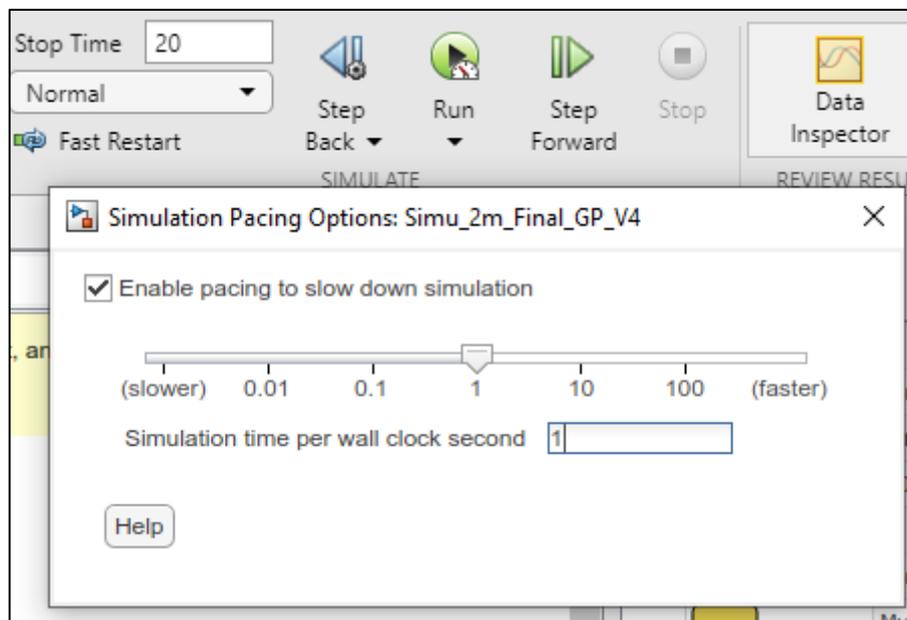


Figura 13: Configuración del Simulation Pacing

Además, es necesario establecer las condiciones bajo las que se someterá el mecanismo en la simulación. La librería de *Simscape Multibody* proporciona tres bloques mostrados en la Figura siguiente:

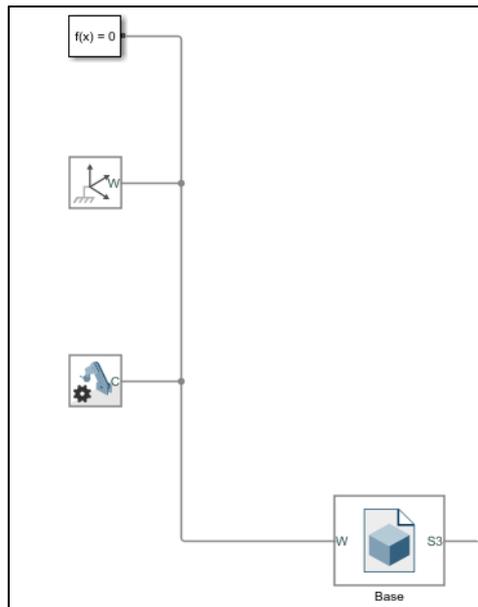


Figura 14: Bloques de configuración de la simulación

- **Solver Configuration**

Permite establecer el método de resolución del modelo en caso de querer resolver la dinámica del sistema por otro que no sea el establecido en el entorno de Simulink.

- **World Frame**

Determinar el sistema de referencia global del mecanismo al que va conectado. Dicho sistema es ortogonal, se rige por la regla de la mano derecha y permanece fijo. Además, sirve de base para el resto de los sistemas de referencia auxiliares definidos en el mecanismo.

- **Mechanism Configuration**

Permite establecer los parámetros mecánicos como la gravedad a la que se ve sometido nuestro sistema. En este caso, se especifica un valor de aceleración de $-9,81 \text{ m/s}^2$ en el eje Z, que corresponde a la gravedad en la Tierra.

Para aplicar las condiciones de simulación a nuestro mecanismo se enlaza a la primera pieza que constituye la base de este. Dicha “Base” es el bloque de unión sobre el cual se irán añadiendo de forma secuencial las piezas que componen el mismo a las que se aplicarán las mismas condiciones.

5.2 Exportación del modelo 3D

Una vez diseñado el modelo de nuestro mecanismo en *SolidWorks* se exporta a Simulink pieza a pieza. Para ello, cada una de ellas se exportan a una extensión CAD que es compatible con *Simulink*, siendo en este caso la extensión *.STEP*. Una vez exportadas se establecen las condiciones tanto de posición como de relación de movimiento entre ellas por medio de bloques específicos de la librería de *Simscape Multibody*.

5.2.1 Bloques de Simscape Multibody

A parte de los bloques básicos que dispone *Simulink*, en este proyecto son necesarios el uso de cinco bloques de la librería de *Simscape Multibody* cuya explicación se considera de importancia. Estos bloques son:

- ***Solid***

Permite cargar un sólido en extensión de CAD tipo *.STEP* en el que viene dada su geometría y además definir sus características mecánicas como son su inercia respecto a su masa o densidad. Una de las opciones más importantes de este bloque es que dispone de un apartado en el que se pueden configurar ejes de referencias respecto al sólido (las cuales son accesibles por medio de puertos de unión) y que permiten establecer la relación de posición con las piezas a las que se desee unir.

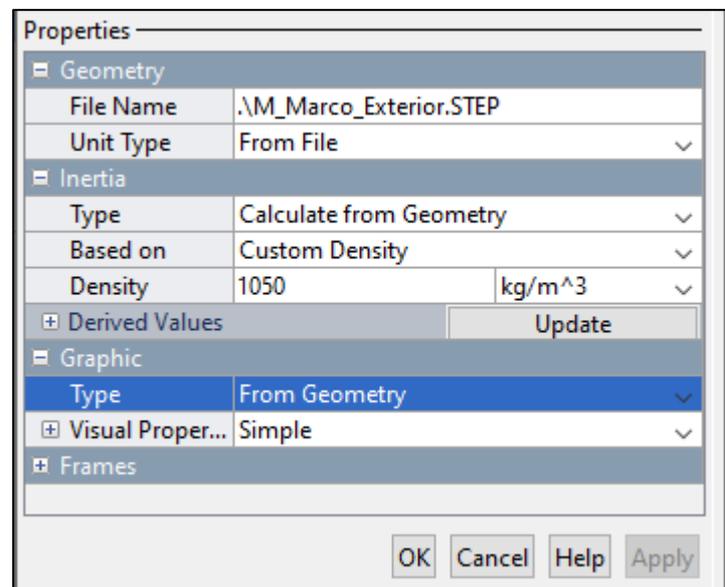


Figura 15: Propiedades del bloque Solid

- ***Rigid transformation***

Permite establecer una transformación rígida tridimensional fija entre dos ejes. Para ello, como se muestra en la Figura X se dispone de dos opciones que pueden ser combinadas entre sí: la relación de rotación y la translación. La ventaja frente a los ejes de referencia configurables en el bloque de *Solid* es que se puede establecer de forma numérica la distancia y la orientación entre dicho sólido y el eje de referencia global del sistema.

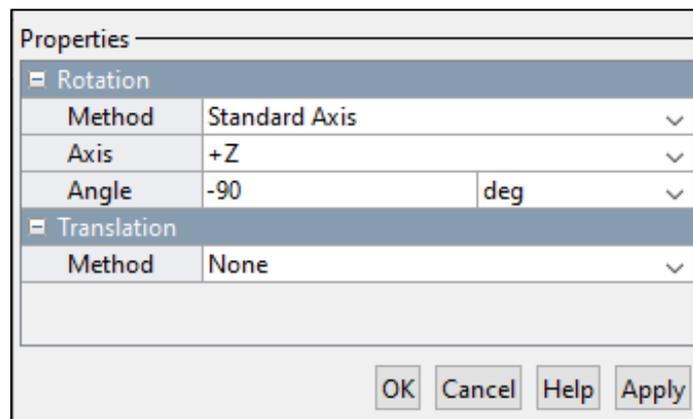


Figura 16: Propiedades del bloque Rigid Transformation

- **Joints**

Permite establecer un tipo de articulación entre dos sólidos, implantando una condición de movimiento. En nuestro caso en concreto, se han utilizado exclusivamente articulaciones del tipo revolución ya que en el mecanismo no existen desplazamientos lineales. Este tipo de articulación es de un grado de libertad y obliga a los orígenes de las piezas de unión y a su eje en Z a ser coincidentes para que los ejes X e Y giren alrededor de este. Además, permite acceder a varias configuraciones entre las que se destacan: las características mecánicas como rozamiento o amortiguamiento, la posibilidad de que sea accionada por una fuerza externa tanto por par como por movimiento y la posibilidad de tener acceso a las variables de posición, velocidad, aceleración y a la actuación del par a modo de sensor.

Properties		
Z Revolute Primitive (Rz)		
+ State Targets		
- Internal Mechanics		
Equilibrium Position	0	deg
Spring Stiffness	0	N*m/deg
Damping Coefficient	0.0015	N*m/(deg/s)
- Limits		
+ Specify Lower Limit	<input checked="" type="checkbox"/>	
+ Specify Upper Limit	<input checked="" type="checkbox"/>	
+ Actuation		
- Sensing		
Position	<input checked="" type="checkbox"/>	
Velocity	<input type="checkbox"/>	
Acceleration	<input type="checkbox"/>	
Actuator Torque	<input type="checkbox"/>	
Lower-Limit Torque	<input type="checkbox"/>	
Upper-Limit Torque	<input type="checkbox"/>	
+ Mode Configuration		
+ Composite Force/Torque Sensing		
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/> <input type="button" value="Apply"/>		

Figura 17: Propiedades del bloque Revolute Joint

- **Force**

Permite aplicar una fuerza o un par de torsión externas al mecanismo en el sistema de coordenadas conectado en dirección de cualquiera de sus tres ejes (X, Y y Z). El valor de la fuerza y el par son especificados por las entradas de señales físicas

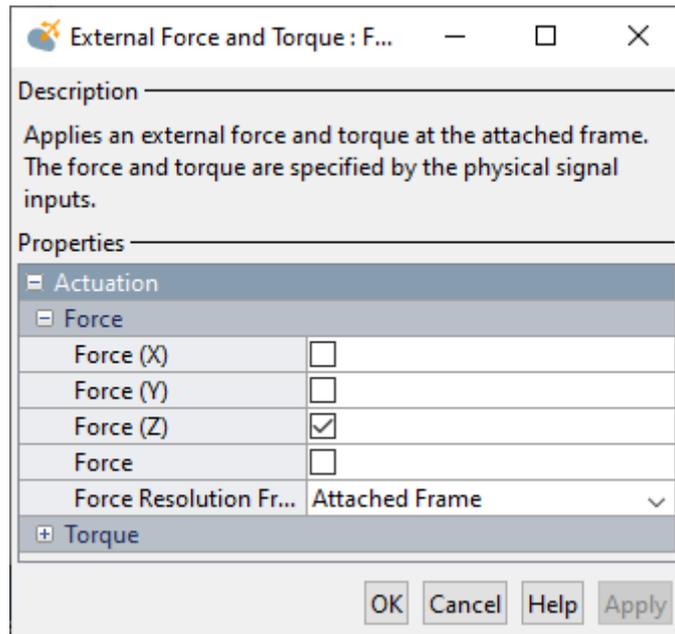


Figura 18: Propiedades del bloque External Force and Torque

- **Simulink-PS Converter**

Permite convertir la señal de entrada de *Simulink* sin unidad a una señal física compatible con la librería de *Simscape Multibody* o viceversa. Dentro de los parámetros se puede escoger la unidad directamente o seleccionar la opción de “1” por defecto que adecua la señal de *Simulink* a las unidades requeridas por el bloque de *Simscape Multibody*.

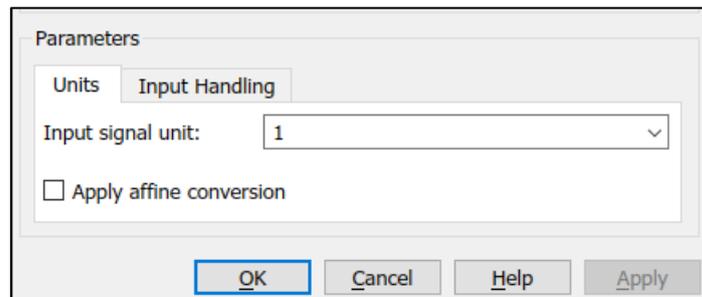


Figura 19: Propiedades del bloque Simulink-PS Converter

5.2.2 Construcción del mecanismo

Una vez explicados los bloques usados y sus parámetros configurables, se explica de forma breve cómo se han usado dichos bloques para la construcción del mecanismo.

En este caso se parte del elemento fijo de nuestro mecanismo constituido por la “Base”, al que se le aplican las condiciones de simulación impuestas por los bloques de configuración de la *Simscape Multibody* tal y como se muestra en la Figura 12. A

continuación, se conectarán el resto de las piezas estableciendo los puntos de unión correspondientes y los pares de revolución que permitan el movimiento del mecanismo.

Para la unión entre piezas es importante configura correctamente dichos puntos de unión o puertos, ya que sus *frames* de conexión tienen que respetar el posicionamiento espacial del mecanismo para que concuerde con el real y se hagan correctamente el cálculo de las ecuaciones de movimiento que describen la dinámica del sistema. Además de ello, es importante configurar los pesos o densidades de cada una de las piezas por medio del bloque *Solid*.

Por otra parte, para todas las articulaciones del mecanismo se ha establecido que el elemento que permite el giro es el sensor. Para ello, el cuerpo del sensor se ha separado en dos piezas: la primera llamada parte fija (cuerpo negro) y la segunda la parte móvil (cuerpo gris). De esta forma, se establece un par de revolución entre ambas y al igual que el sensor real permitirá el giro entre las piezas a las que estén fijadas.

A continuación, se repasan brevemente las uniones de las piezas que componen el mecanismo.

En primer lugar, en la Figura X se representa la unión entre la “Base” y los “Pilares” por medio del sensor. Al tratarse del sensor de guiñada el recorrido no estará limitado, es decir, se permite el giro por completo de 360°. El puerto de salida de los “Pilares” se conecta al “Marco exterior”.

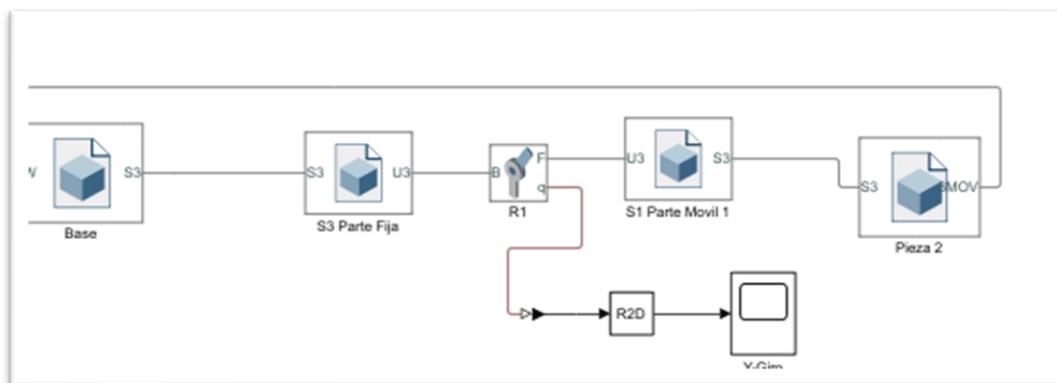


Figura 20: Simulink bloques de unión entre Base y Pilares

En segundo lugar, en la Figura X se representa la unión entre el “Marco interior” y los componentes que contiene como son los rodamientos, ejes, y el sensor que permite el cabeceo del mecanismo. El recorrido del par de revolución del sensor está limitado a

$\pm 45^\circ$ coincidiendo con los 90° de lectura que permite el sensor real. En este caso, se ha usado el bloque de *Rigid transformation* para facilitar el posicionamiento de la parte fija del sensor al “Marco exterior”. Al final se deja accesible el puerto de salida de la parte móvil del sensor de cabeceo que se unirá al “Marco interior”

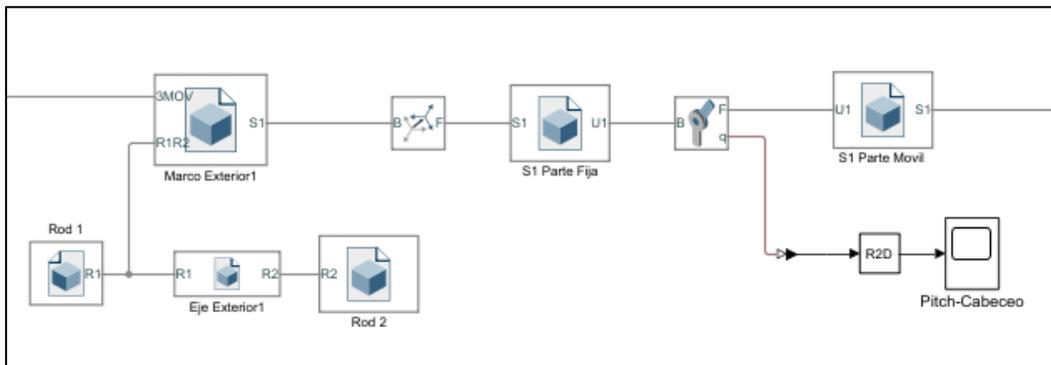


Figura 21: Simulink bloques del Marco exterior y componentes

Finalmente, en la figura X se representa la unión entre el la “Plataforma” y el “Marco interior” con todos los restos de componentes que van adheridas a él: rodamientos, eje, pesos para equilibrar el centro de masas del sistema y el sensor que permite el alabeo del mecanismo. Al igual que en el caso anterior, el recorrido del par de revolución del sensor está limitado a $\pm 45^\circ$ coincidiendo con los 90° de lectura que permite el sensor real.

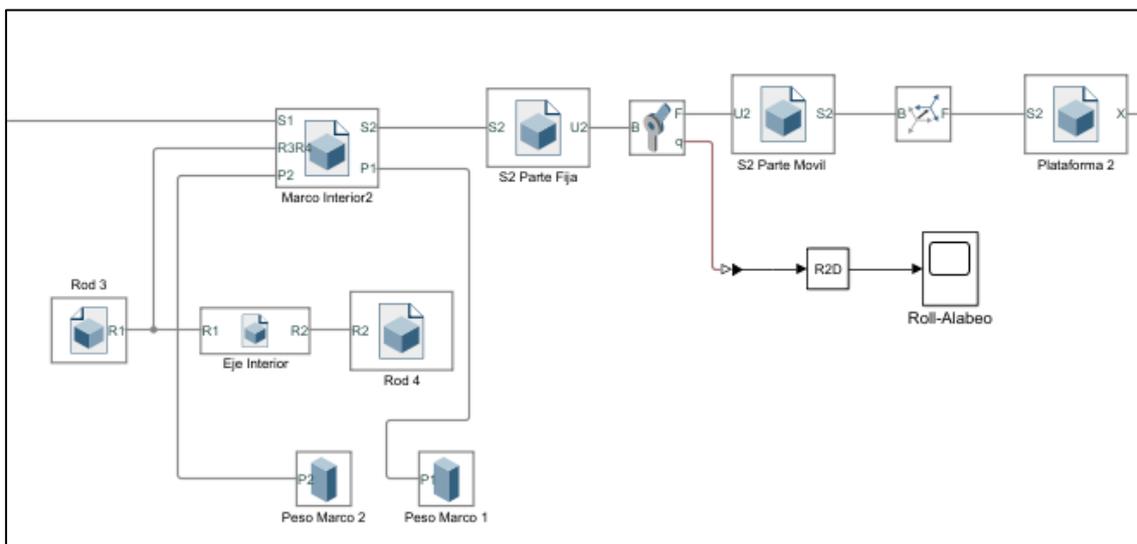


Figura 22: Simulink bloques de unión entre Marco interior y Plataforma

Como se observa en los tres pares de revolución se accesible la lectura del ángulo correspondiente en cada caso. Para ello, se hace uso del bloque *Simulink-PS Converter* que permite extraer las magnitudes físicas de la simulación del mecanismo y poder trabajar con ellas. Por medio del bloque de conversión R2D se obtienen de forma gráfica los valores de los ángulos de giro en cada uno de los pares de revolución.

5.3 Condiciones de fuerza

Para la implementación de las fuerzas responsables del movimiento del mecanismo se transformarán las velocidades angulares de cada una de las hélices, respetando la relación entre sí, en fuerzas y momentos de fuerza que se transmitirán a la plataforma.

Para simular en tiempo real el movimiento del mecanismo, el cuerpo del dron se ha simplificado como un cuerpo sólido simétrico de masa equivalente al dron real con cuatro pares de revolución en cada una de sus esquinas y sus correspondientes hélices.

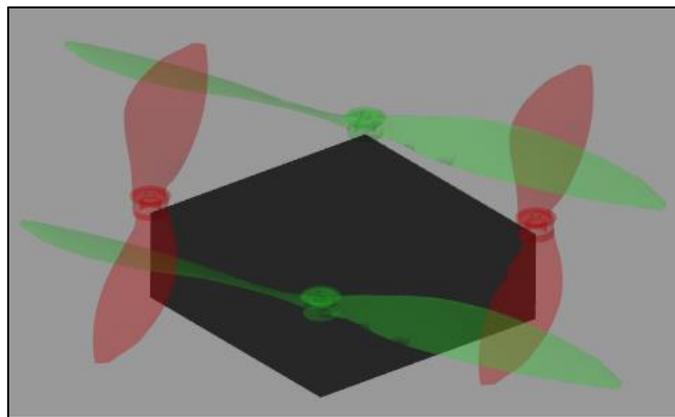


Figura 23: Simplificación cuerpo sólido del cuadricóptero

En la simulación de las fuerzas y los momentos, los bloques *Joints* correspondientes a los pares de revolución serán los responsables del momento de fuerza de cada una de las hélices y, a su vez, generan el momento resultante que controlará el giro del mecanismo. Respecto a las fuerzas lineales, se definirán puntos en cada uno de los pares de articulación y se aplicarán fuerzas en función de la velocidad angular haciendo una estimación aplicando la fórmula de la fuerza de empuje.

Para la implementación de lo expuesto en el entorno de Simulink se realiza tal y como se muestra en la siguiente figura:

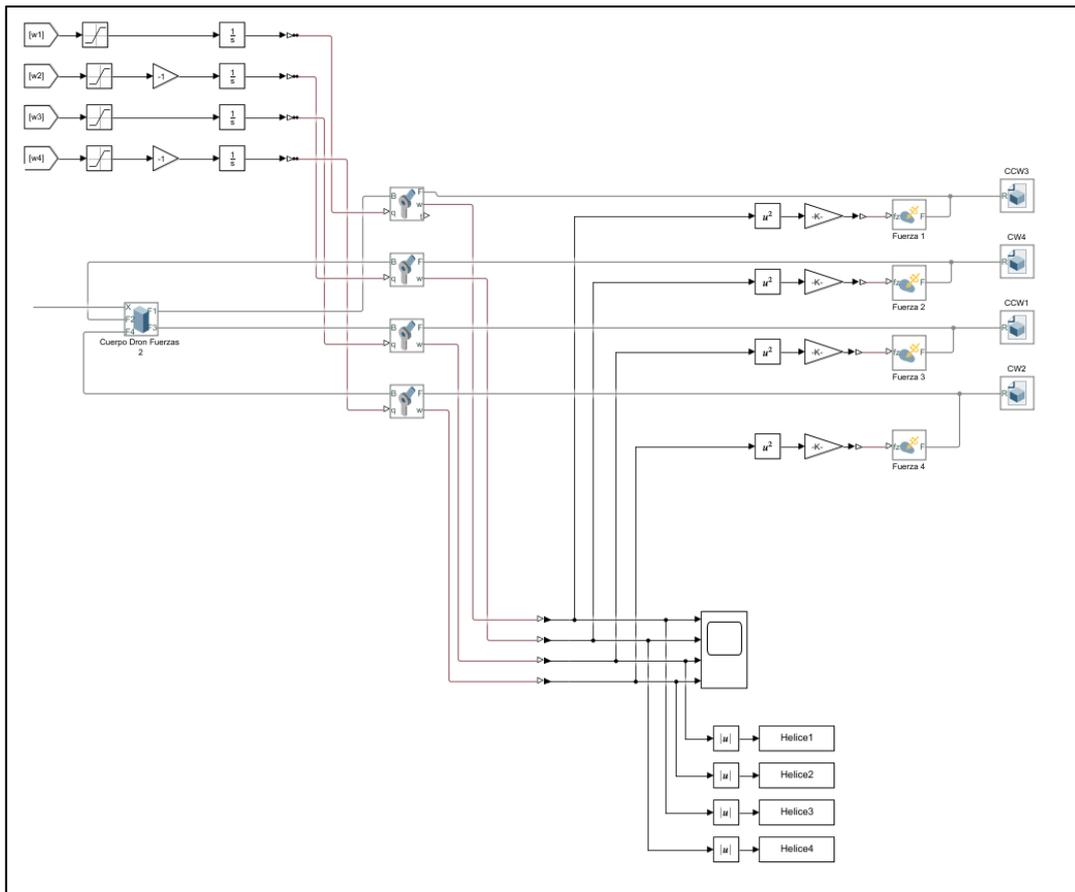


Figura 24: Simulink de la implementación del cuadricóptero y sus fuerzas

En primer lugar, se establece las conexiones físicas entre el modelo simplificado del cuadricóptero sobre la plataforma y sus correspondientes hélices por medio del bloque de par de revolución.

Una vez dadas las velocidades angulares (explicadas en el siguiente punto), se establecen unos valores mínimos y máximos y se invierten el sentido de giro de las hélices opuestas (hélices 2 y 4 de color verde que giran en sentido horario). La inversión se realiza para compensar el momento resultante en el centro de masa y anularlo en caso de que la velocidad de las palas enfrentadas sea igual ya que, en caso contrario, el momento resultante tomaría el sentido de giro opuesto al de las hélices y giraría de forma descontrolada al igual que lo haría un cuadricóptero. Además, se hace uso de un integrador para conseguir las unidades de movimiento a través de las velocidades dadas y ofrecer dichos datos de entrada al bloque *Joint* como consigan del movimiento.

De esta forma, los momentos generados en el centro de masa de la base de cuadricópteros son calculados automáticamente a través del movimiento de las articulaciones y los momentos de inercia de las hélices.

Por otra parte, la salida de cada bloque *Joint* ofrece la posibilidad de leer la velocidad angular generada entre el cuerpo del dron y la cada una de las hélices. Para conocer las fuerzas lineales de cada hélice se recurre a la siguiente fórmula, donde se determina que dicha fuerza es proporcional al cuadrado de la velocidad de las hélices.

$$\text{Fuerza de empuje} = \frac{1}{2} \cdot C_D \cdot \rho \cdot A \cdot w^2$$

Donde:

C_D = Coeficiente de arrastre

ρ = Densidad del aire

A = Área de la hélice

w = velocidad angular de las palas

Gracias a dicha fórmula, se hace una estimación de la fuerza de reacción que se generan en las hélices al paso del aire que hace que el cuadricóptero se eleve.

Para su simplificación se tendrá en cuenta la velocidad directamente del par de revolución y se elevará al cuadrado y se englobará todos los coeficientes en uno único con un valor de $1 \cdot 10^{-5}$.

Los puntos de aplicación de las fuerzas lineales de cada una de las palas serán en el centro de sus respectivas articulaciones y equidistantes del centro de masa tanto del cuerpo del dron como de la plataforma, tal y como se indican en la figura siguiente:

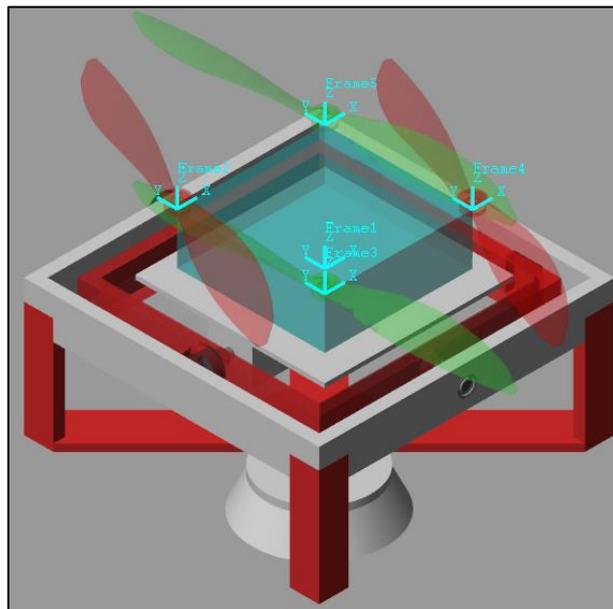


Figura 25: Puntos de aplicación de las fuerzas lineales

De esta forma, se implementa el momento de fuerza de las hélices de forma automática por medio de cada uno de los pares de revolución y, por otro lado, se calculan y se introducen en tiempo real las fuerzas lineales directamente proporcionales a la velocidad de giro de las hélices.

5.4 Condiciones de velocidad de las hélices

Para determinar las condiciones de velocidad angular de cada una de las hélices que responsables de las fuerzas que mueven el mecanismo en la simulación, hay que tener en cuenta las consideraciones hechas en el punto 2.3 *Relación de las velocidades angulares de las hélices y el movimiento del cuadricóptero*. Donde las hélices y los movimientos fueron identificados de la siguiente forma:

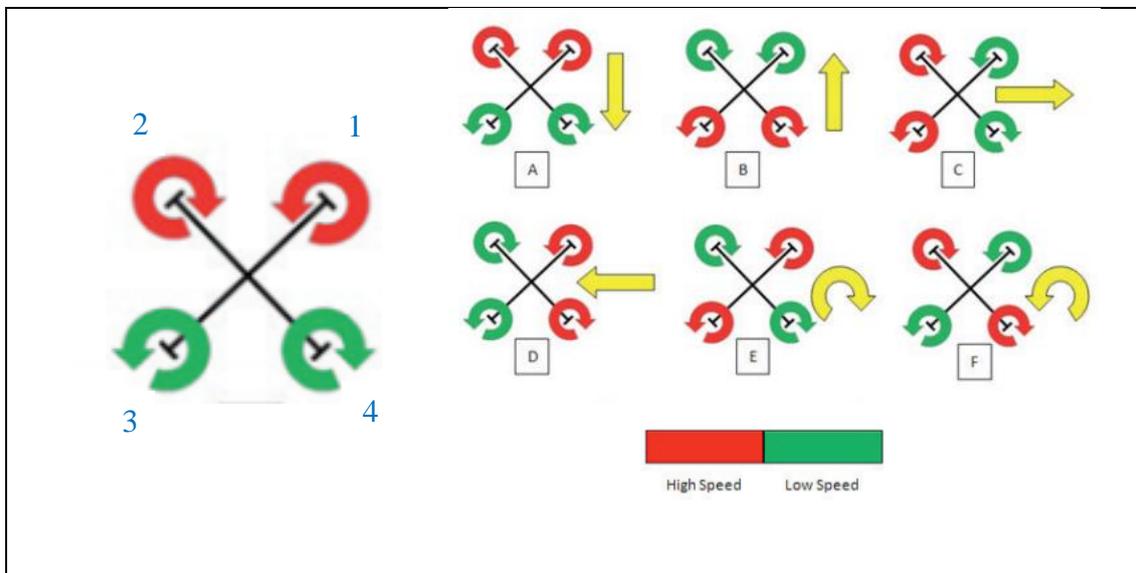


Figura 26: Identificación de hélices y movimientos del cuadricóptero en la simulación

En primer lugar, se tiene en cuenta una velocidad común en cada una de las hélices que será la velocidad que genera una fuerza de sustentación que es igual al peso del dron.

Para determinar el valor de esta, se calcula la fuerza peso del dron (F_{Pd}) de la siguiente forma:

$$F_{Pd} = m_d \cdot g = 0.12 \cdot 9.81 = 1,1772 \text{ N}$$

Dicha fuerza, será la resultante de las 4 hélices por lo que la necesaria en cada una de ellas es:

$$F_{Pdh} = \frac{F_{Pd}}{4} = \frac{1,1772}{4} = 0.2943 \text{ N}$$

La relación entre la velocidad angular y fuerza viene dada por la expresión de la fuerza de sustentación explicada en el punto anterior, donde de forma simplificada aplicada a cada hélice:

$$F_{Pdh} = \frac{1}{2} \cdot C_D \cdot \rho \cdot A \cdot w^2 = 1 \cdot 10^{-5} \cdot w^2$$

Despejando la incógnita de la velocidad angular (w):

$$w = \sqrt{\frac{F_{Pdh}}{1 \cdot 10^{-5}}} = \sqrt{\frac{0.2943}{1 \cdot 10^{-5}}} = 171.56 \text{ rad/s}$$

Se obtiene una velocidad mínima de **171.56 rad/s** en cada hélice para que genere una fuerza mínima que sea capaz de anular la componente del peso del cuadricóptero.

Respecto a las velocidades responsables del movimiento se suman y se restan dos a dos en las hélices el valor introducido para cada tipo de movimiento y así generar la descompensación de velocidades. De esta forma, respetando las relaciones explicadas en el punto 2.3, se determina que para generar cada uno se debe actuar en las hélices correspondientes dos a dos de la siguiente forma:

- Cabeceo (Pitch): Dependiente de la velocidad angular de cabeceo (w_c). Actúa de igual manera (signo +) sobre las hélices 1 y 2 y de forma contraria (signo -) en la 3 y 4. Un valor positivo de w_c hará que las hélices 1 y 2 giren a mayor velocidad produciendo un desplazamiento hacia atrás tal y como se muestra en la imagen A de la Figura 26. Del mismo modo, una w_c negativa producirá el efecto contrario donde las hélices 3 y 4 tendrán más velocidad y se produzca un movimiento hacia delante (B).
- Alabeo (Roll): Dependiente de la velocidad angular de alabeo (w_a). Actúa de igual manera (signo +) sobre las hélices 1 y 4 y de forma contraria (signo -) en la 2 y 3. Un valor positivo de w_c hará que las hélices 1 y 4 giren a mayor velocidad produciendo un desplazamiento lateral hacia la izquierda tal y como se muestra en la imagen D de la Figura 26. Del mismo modo, una w_a negativa producirá el efecto contrario donde las hélices 2 y 3 tendrán más velocidad y se produzca un movimiento lateral hacia la derecha (C).
- Guiñada (Yaw): Dependiente de la velocidad angular de alabeo (w_g). Actúa de igual manera (signo +) sobre las hélices 1 y 3 con sentido de giro antihorario y de

forma contraria (signo -) en la 2 y 4 con sentido de giro horario. Un valor positivo de w_c hará que las hélices 1 y 3 giren a mayor velocidad generando una descompensación en el momento haciendo que se produzca el giro en sentido horario tal y como se muestra en la imagen E de la Figura 26. Del mismo modo, una w_g negativa producirá el efecto contrario donde las hélices 2 y 4 tendrán más velocidad generando una descompensación en el momento haciendo que se produzca el giro en sentido antihorario (F). Como se observa en este movimiento, cuando la velocidad de las palas en sentido horario (2 y 4) es superior a las que giran en sentido antihorario (1 y 3) y por tanto su momento también lo es, por la ley de equilibrio de momentos ($\Sigma M_{ext} = 0$), en el centro de gravedad del cuadricóptero se produce un momento resultante para compensar el desequilibrio y hace que gire en sentido opuesto.

Trasladando lo expuesto anteriormente, se concluye que la velocidad angular de cada hélice viene determinada por la velocidad mínima que sustenta el peso del cuadricóptero y las velocidades de cada tipo de movimiento. Esta relación viene dada por las siguientes expresiones correspondientes a cada una de las hélices:

$$w_{h1} = w_0 + w_c + w_a + w_g$$

$$w_{h2} = w_0 + w_c - w_a - w_g$$

$$w_{h3} = w_0 - w_c - w_a + w_g$$

$$w_{h4} = w_0 - w_c + w_a - w_g$$

Donde:

$w_0 = w$ mínima para soportar el peso del dron

$w_c = w$ de cabeceo

$w_a = w$ de alabeo

$w_g = w$ de guiñada

La forma de implementación viene dada por los siguientes bloques de Simulink:

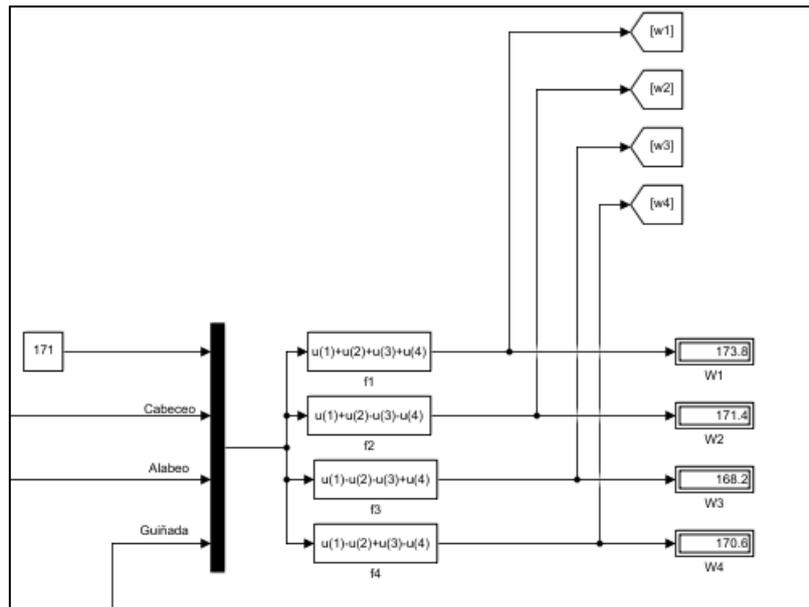


Figura 27: Simulink de la implementación de las expresiones de velocidad angular

Donde las expresiones se introducen mediante el bloque “Fcn” de Simulink y además se tienen acceso al valor de velocidad de cada una de las palas. Los valores de entrada a

5.5 Introducción de velocidades

Los valores de velocidad angular correspondientes al cabeceo, alabeo y guiñada se introducen de forma manual para el control del movimiento del mecanismo durante la simulación en tiempo real.

Para ofrecer una experiencia de control sobre el movimiento más real se ha implementado un bloque de Simulink capaz de hacer una lectura de los botones y joysticks de un mando físico. En este caso, se ha usado un mando de la *PlayStation 3* cuyos botones han sido seleccionados por no ser botones todo o nada, ya son capaces de ofrecer valores progresivos dependiendo de su posición.

De esta forma, los botones usados son:

- Joystick izquierdo: ofrece valores entre -1 y 1, con el que se controla las velocidades angulares de cabeceo
- Joystick izquierdo: ofrece valores entre -1 y 1, con el que se controla las velocidades angulares de alabeo
- Botón L2 y R2: en este caso el botón L2 ofrece el recorrido entre 0 y 1 y el R2 entre 0 y -1 actuando como un único botón, con el que se controla las velocidades angulares de guiñada.

De esta forma, los controles usados y las direcciones con los signos viene representada por la siguiente figura:



Figura 28: Controles usados del mando

Para su implementación, se ha usado el siguiente esquema de *Simulink*:

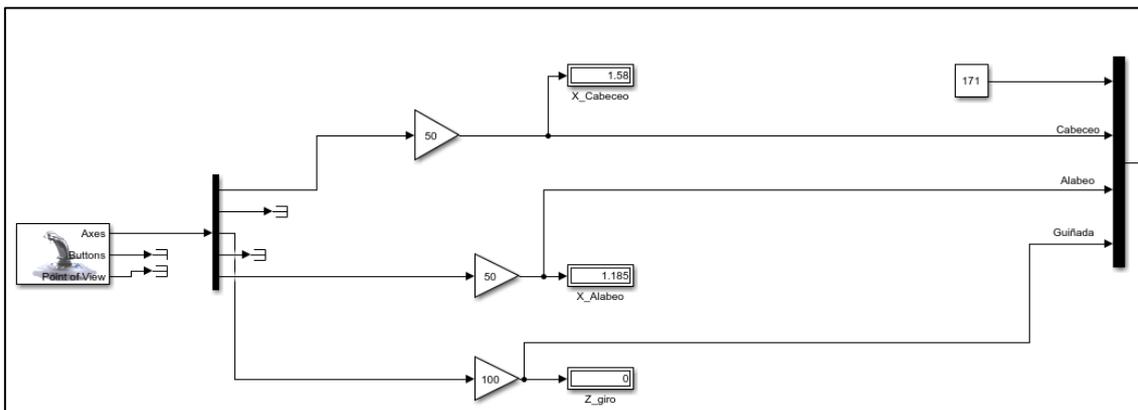


Figura 29: Simulink para la lectura y consigna de las velocidades

En este se observa como los valores que es capaz de ofrecer el mando son amplificados por una ganancia de 50 para el cabeceo y alabeo y 100 para la guiñada. De esta forma, se obtiene un rango de valores entre $[-50 \text{ y } +50]$ y $[-100 \text{ y } +100]$ para introducir como velocidades angulares para cada uno de los tipos de movimiento.

Con todo ello, el control sobre el valor de las velocidades durante la simulación es absoluta y se puede interactuar con el mecanismo en tiempo real para controlar su movimiento.

5.6 Resultados de la simulación

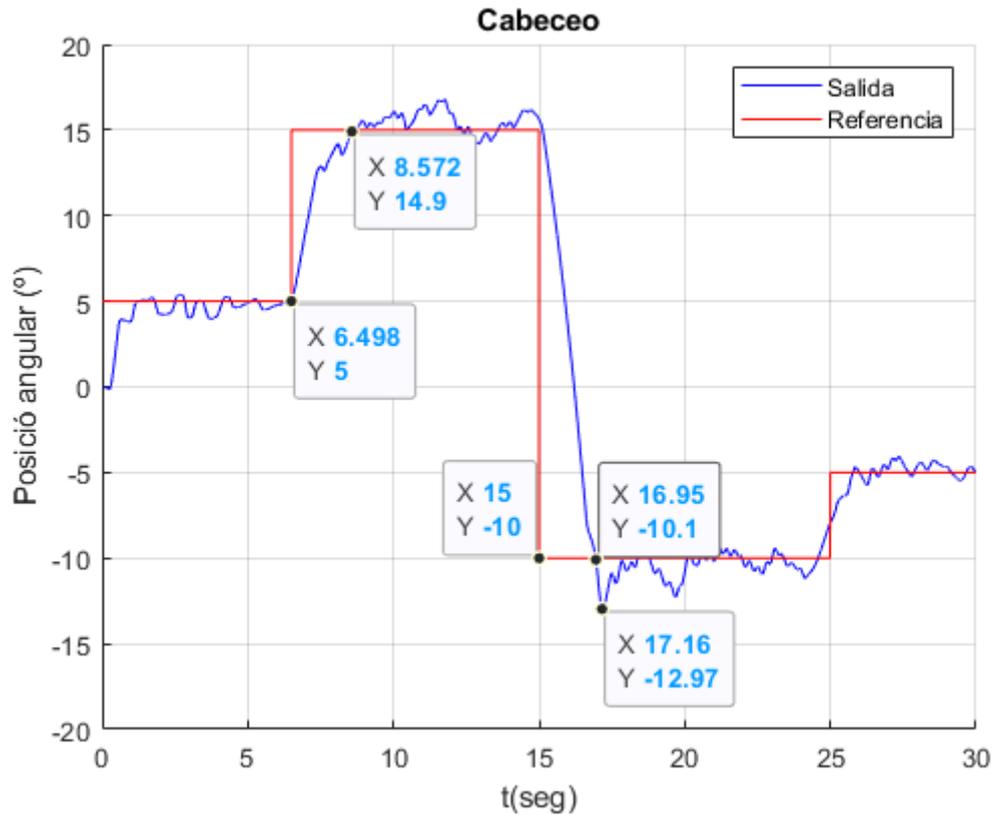
La simulación del movimiento del mecanismo viene dada por las condiciones de fuerza y velocidad explicadas en los puntos anteriores. Además, como se ha visto los valores de las velocidades angulares de cabeceo, alabeo y guiñada serán introducidas en tiempo real mediante un mando físico.

El objetivo de la simulación del mecanismo en movimiento es observar cómo se comporta ante los comandos de velocidad introducidos. Teniendo en cuenta, que el ángulo de cabeceo y alabeo son aproximadamente de $\pm 15^\circ$ en un cuadricóptero real, lo que se pretende con dicha simulación es mover el mecanismo entre ese rango de movimiento.

Para comprobar el control manual se pondrá como referencia ángulos comprendidos entre $\pm 15^\circ$ en el alabeo y cabeceo, y 360° para la guiñada. Manteniendo de forma manual con el joystick una referencia

Cabeceo

Tomando como referencia valores de 5, 15, -10 y -5° la salida en la lectura de la articulación correspondiente al Cabeceo es la siguiente:



Gráfica 1: Seguimiento Cabeceo

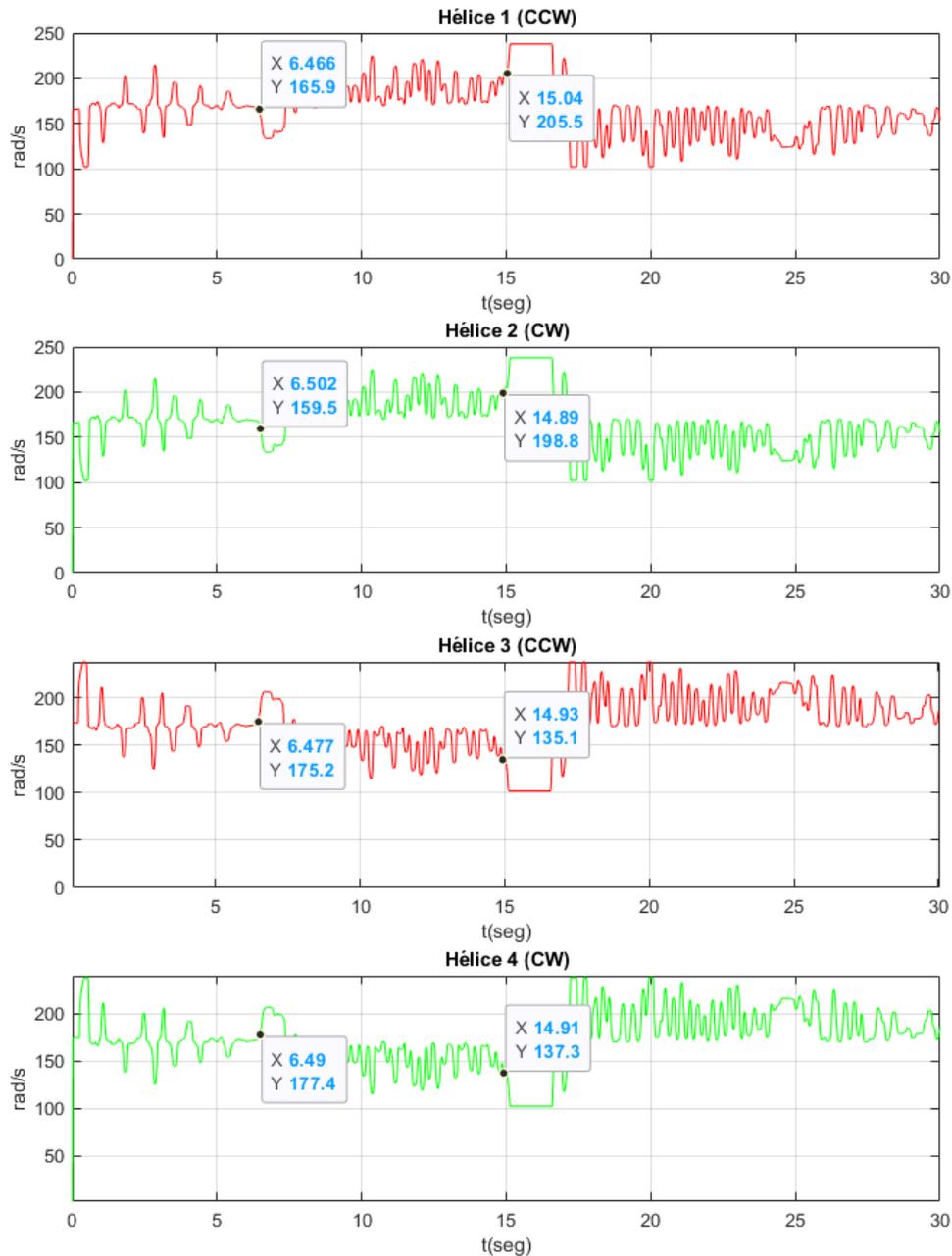
Como se puede observar al llegar a la referencia deseada existen oscilaciones debido al control manual. La de la respuesta del comportamiento se obtiene:

$$\text{Tiempo de establecimiento } (T_s) = 2s$$

$$\text{Sobreoscilación } (\xi) = 30\%$$

Respecto a las velocidades angulares, para facilitar su comparación se han invertido los valores de las hélices 2 y 4 en sentido horario (CW), ya que sus velocidades son negativas.

Las w obtenidas en cada una de las hélices para la respuesta indicada son:

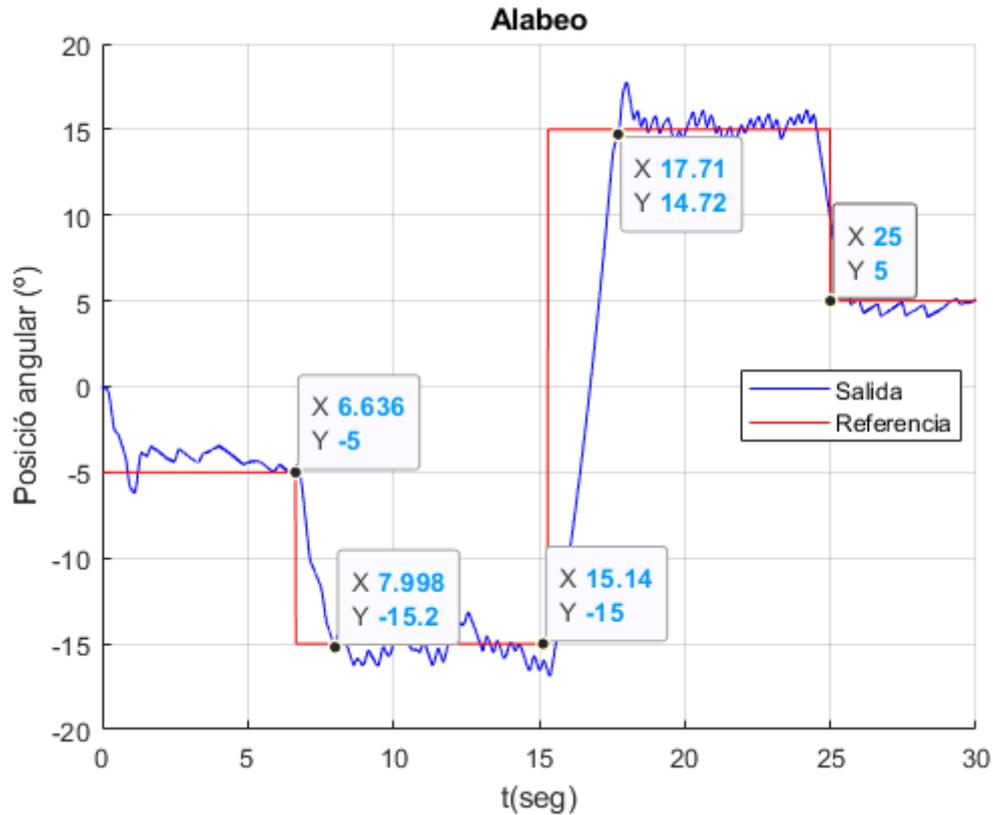


Gráfica 2: Velocidades angulares Cabeceo

Se observa claramente como en los instantes 6,5s y 15s (instantes de máxima diferencia entre referencias de ángulos) las velocidades angulares de las hélices difieren para conseguir el cabeceo. De esta forma, cuando se pretende obtener valores de ángulos + el valor de las w de las hélices 1 y 2 son mayores en el instante 6,5s, y cuando se pretende conseguir ángulos negativos de cabeceo el valor de las w de las hélices 3 y 4 son mayores en el instante 15s.

Alabeo

Tomando como referencia valores de -5, -15, +15 y 5° la salida en la lectura de la articulación correspondiente al Alabeo es la siguiente:



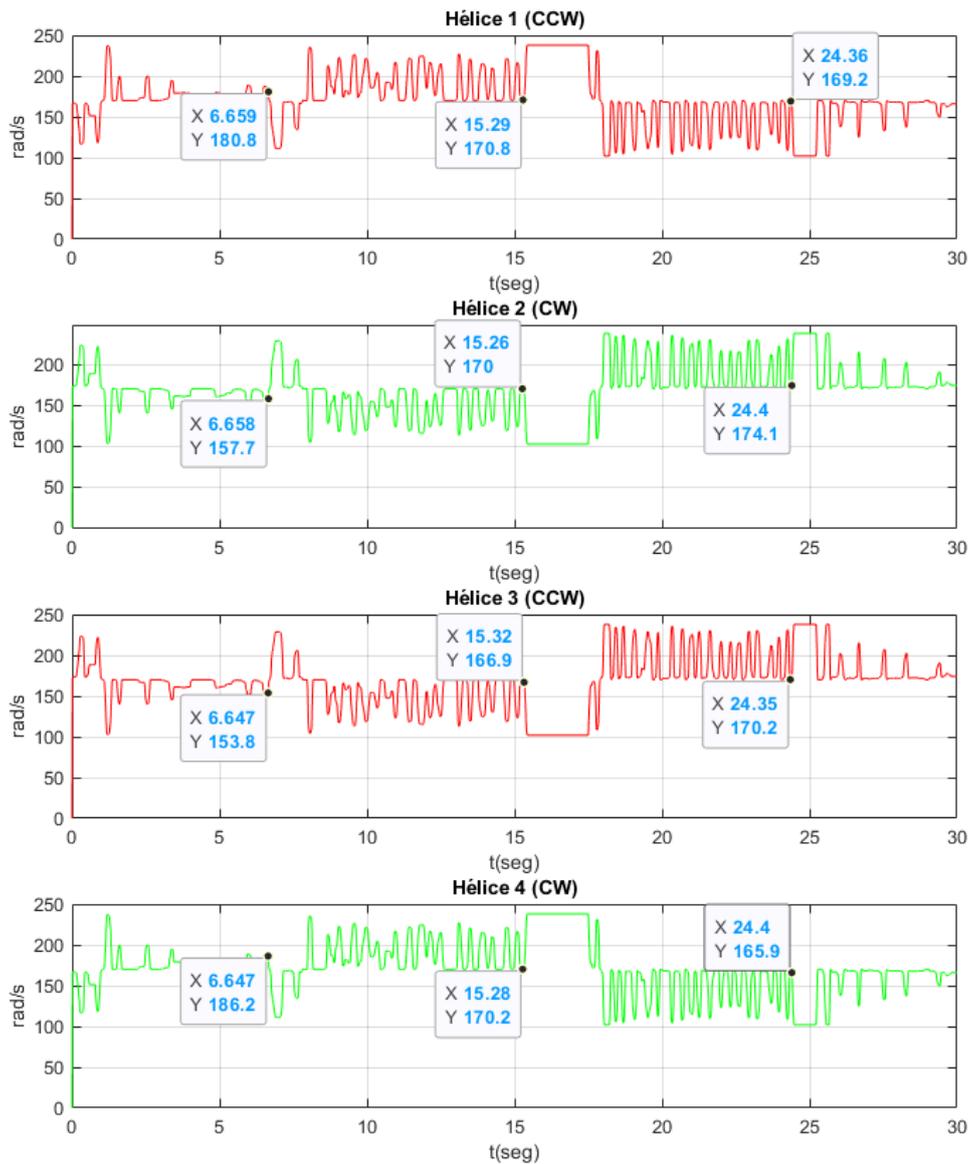
Gráfica 3: Seguimiento Alabeo

Como se puede observar al llegar a la referencia deseada existen oscilaciones debido al control manual. La de la respuesta del comportamiento se obtiene:

$$\text{Tiempo de establecimiento } (T_s) = 2s$$

$$\text{Sobreoscilación } (\xi) = 30\%$$

Las velocidades angulares (ω) obtenidas en cada una de las hélices para la respuesta indicada son:

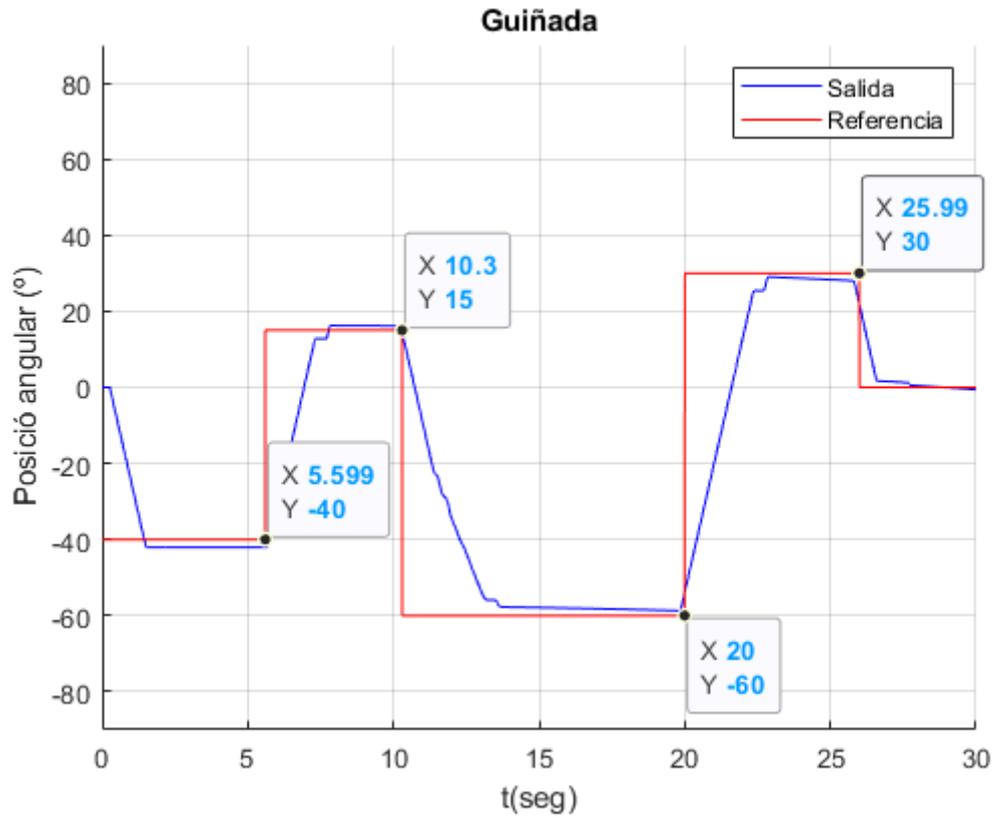


Gráfica 4: Velocidades angulares Alabeo

Se observa como en los instantes 6,6s, 15s y 25s (instantes de máxima diferencia entre referencias de ángulos) las velocidades angulares de las hélices difieren para conseguir el alabeo. De esta forma, cuando se pretende obtener valores de ángulos negativos (-) el valor de las w de las hélices 1 y 4 son menores en el instante 6,6 y 25s, y cuando se pretende conseguir ángulos positivos (+) de alabeo el valor de las w de las hélices 1 y 4 son mayores en el instante 15s, respecto a las hélices 1 y 2.

Guiñada.

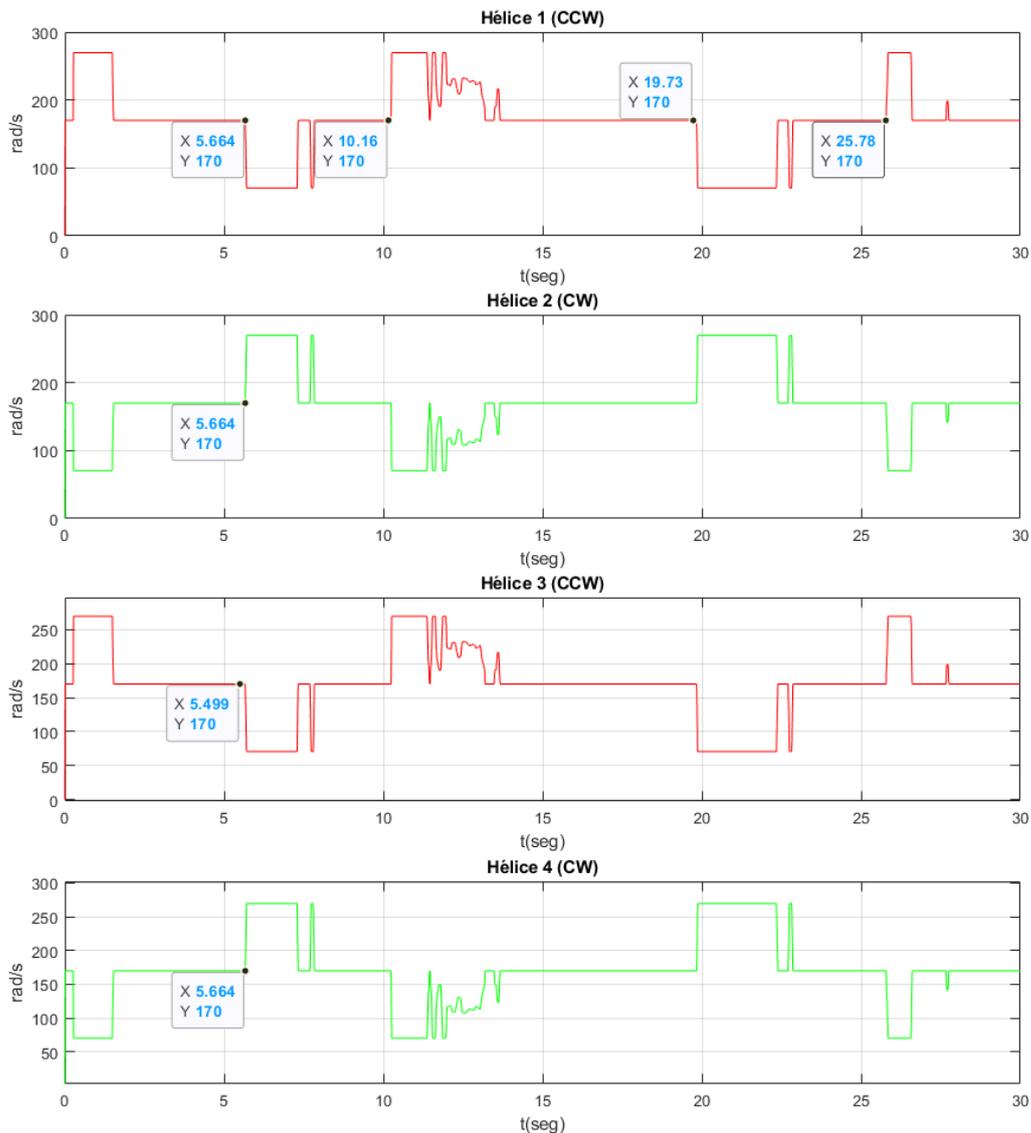
Tomando como ángulos de referencia -40, 15, -60 y 30, la respuesta de la guiñada es la siguiente:



Gráfica 5: Seguimiento Guiñada

En este caso la respuesta no presenta oscilaciones debido a que para generar el control de esta articulación se han bloqueado las articulaciones del alabeo y cabeceo para facilitar el alcance a las referencias impuestas y observar adecuadamente la respuesta de las velocidades angulares de las hélices sin que influyan dichos pares de revolución en esta.

Las velocidades angulares (ω) obtenidas en cada una de las hélices para la respuesta indicada son:



Gràfica 6: Velocidades angulares Guñada

En este caso, algunos de los giros se producen en los instantes 5,6s, 10s, 20s y 25s donde las velocidades angulares de las hélices difieren para conseguir dicho movimiento. De esta forma, si se pretende girar en sentido horario el valor de las velocidades angulares (w) que giran en sentido antihorario deben ser mayores y viceversa. Siendo en los instantes 5,6 y 20s la velocidad de las hélices 2 y 3 (CW) mayores produciendo un giro en sentido antihorario y en los instantes 10s y 25s la velocidad de las hélices 1 y 3 (CCW) mayores produciendo un giro en el sentido horario.

Como se puede apreciar, el tiempo que se tarda en alcanzar la referencia es elevado y la estabilidad del sistema es baja. Estos resultados se corresponden con la realidad, ya que

se establece un control manual en el que las referencias son alcanzados mediante la acción de los controles del mando e interviene el factor humano: coordinación de ojos y manos.

En conjunto

Como final de este apartado se realiza un video demo, adjunto en los archivos entregados, donde interactúan interactuaran los 3 tipos de movimiento a la vez mostrando el comportamiento del mecanismo en tiempo real.

Para que sea más visible se han sustituido las gráficas por indicadores de los ángulos de cabeceo, alabeo y guiñada situados en la izquierda, derecha y centro abajo, respectivamente. Estos nos indican el ángulo en ese instante de cada una de las articulaciones. A continuación, se muestran unas capturas de la simulación.

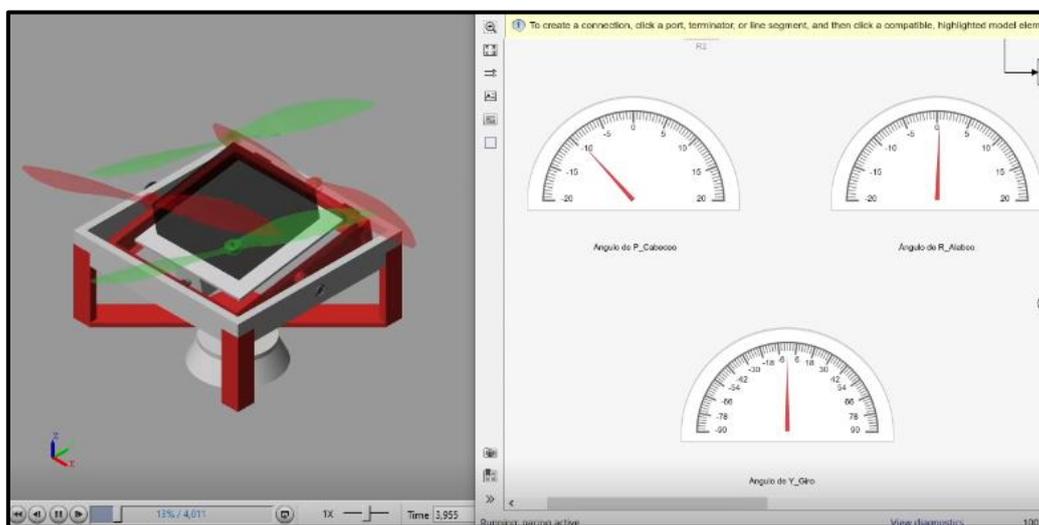


Figura 30: Captura simulación mecanismo 1 Cabeceo

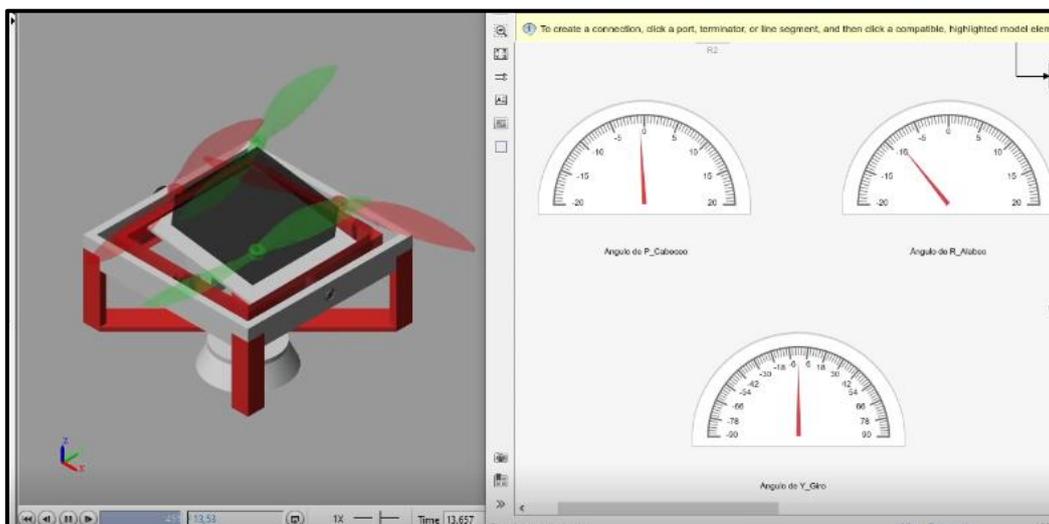


Figura 31: Captura simulación mecanismo 2 Alabeo

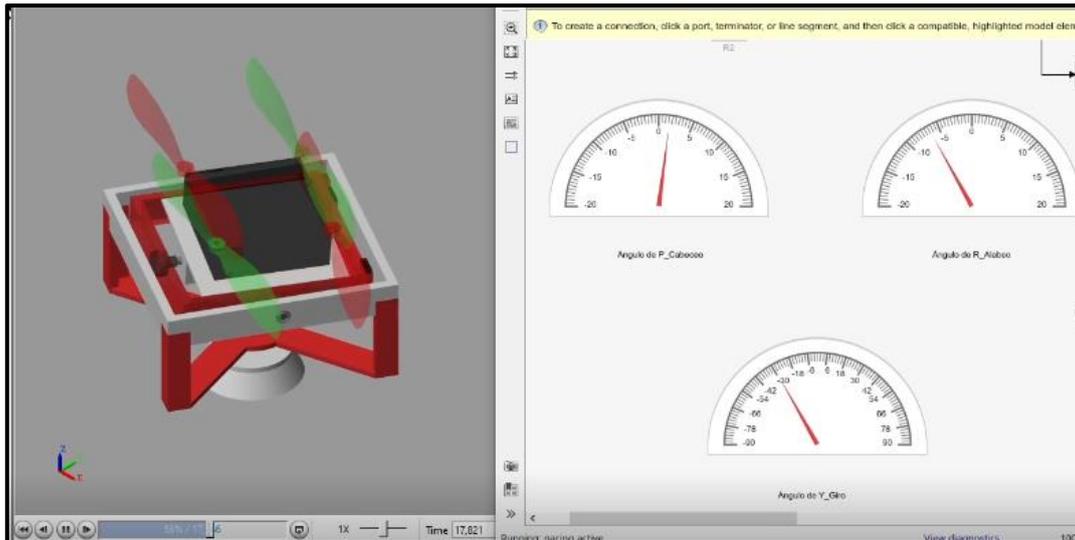


Figura 32: Captura simulación mecanismo 3 Guiñada

En este durante 10 primeros segundos de la simulación se observa un control sobre el cabeceo entre $\pm 10^\circ$ y seguidamente hasta el instante 18s se controla el alabeo. En el tiempo restante se combinan los 3 movimientos a la vez observándose en todo momento el ángulo de cada uno de ellos en los indicadores de la derecha.

Se observa como el comportamiento y la dinámica del mecanismo es el deseado, donde el sistema responde adecuadamente al movimiento de un cuadricóptero. Para que las fuerzas de cabeceo y alabeo mezcladas no tengan a penas efecto sobre la articulación de la guiñada el rozamiento de esta debe ser algo superior a las otras dos.

De esta forma, bajo estas condiciones de simulación al igual que se hará en el mecanismo físico real, mediante un mando RC se pilotará un cuadricóptero anclado sobre la plataforma lo que el movimiento de este se transmitirá a las articulaciones del mecanismo y se buscará una respuesta similar.

6. IMPLEMENTACIÓN DEL SISTEMA REAL

La implementación del proyecto se puede dividir en 4 partes principalmente: construcción mecánica, cableado eléctrico, comunicación serie y entorno de simulación de vuelo.

Una vez diseñadas las piezas del mecanismo para su fabricación se ha hecho uso de los equipos de impresión 3D de los que se dispone en la Escuela Técnica Superior de Diseño de la UPV. Dicha tecnología permite crear un mecanismo formado por piezas principalmente de plástico, funcionales, ligero y duradero de bajo coste idóneo para la creación de este prototipo.

Tras el montaje se procede al cableado eléctrico para alimentar eléctricamente a las partes eléctricas del mecanismo y tener a disposición los valores de lectura de los potenciómetros magnéticos.

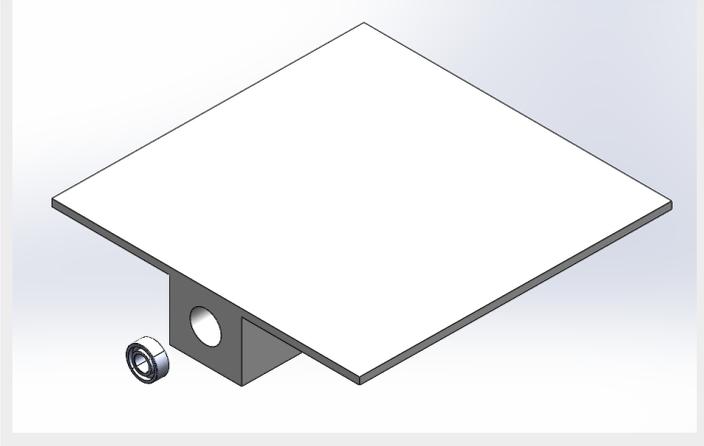
De la misma forma, para el tratamiento de los valores de lectura correspondientes a los ángulos de giro del cuadricóptero y su posterior transferencia al entorno de simulación, es necesario establecer una comunicación Serie. Dicha comunicación se realiza entre la placa de *Arduino DUE* usada y el programa de *MATLAB* que contiene el entorno de simulación.

Finalmente, el entorno de simulación creado en *Simulink* contará con un cuadricóptero de simulación que sea capaz de imitar los movimientos del cuadricóptero físico situado sobre la plataforma del mecanismo en tiempo real. Además, de reproducir sus movimientos el cuadricóptero se podrá desplazar por el entorno en el eje X, Y y girar en Z en función de los ángulos de navegación.

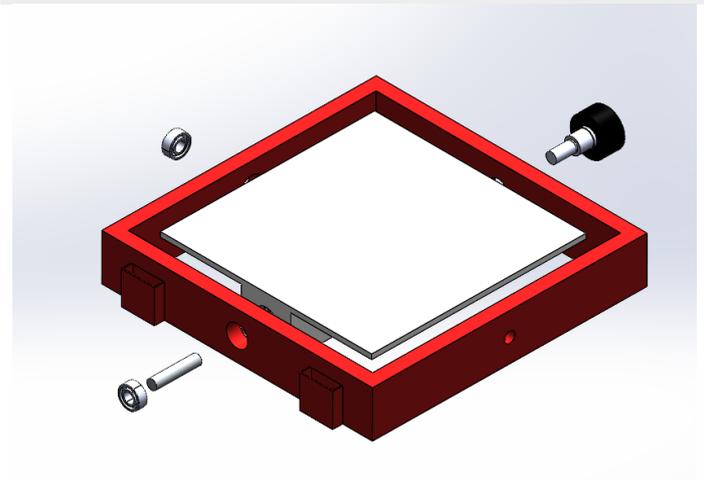
6.1 Montaje mecánico

El montaje del mecanismo se lleva a cabo de dentro hacia fuera. Siendo la secuencia de este la siguiente:

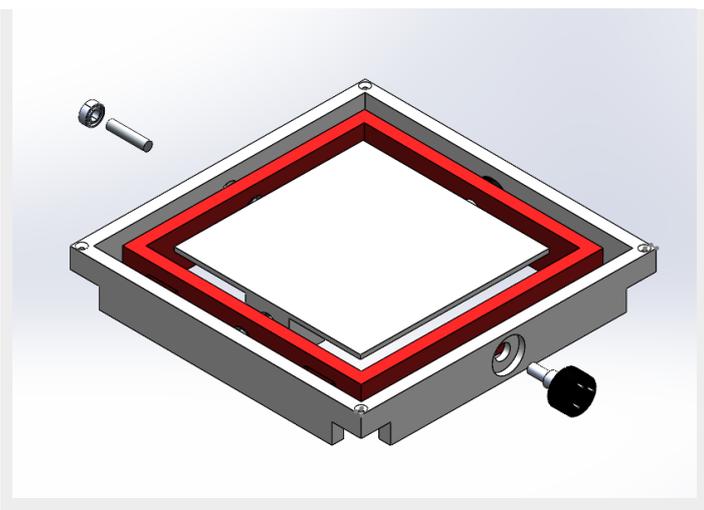
1. Se introduce el primer rodamiento dentro de la plataforma del eje longitudinal.



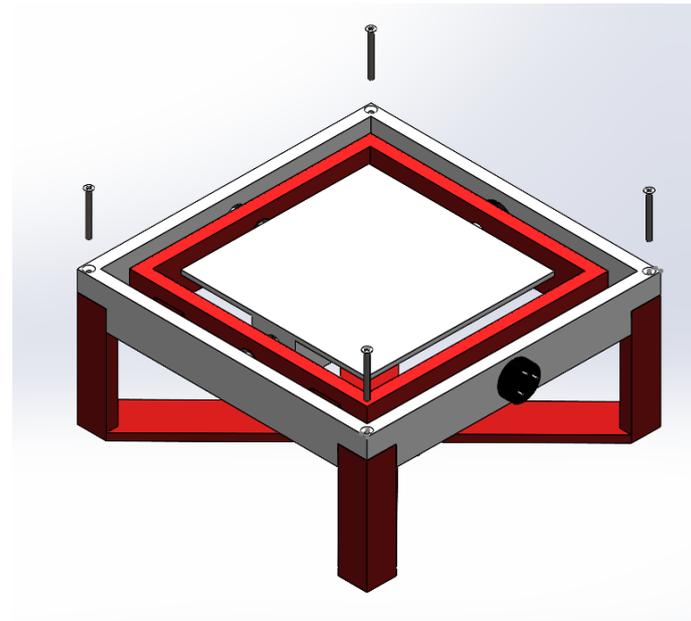
2. Se dispone la plataforma dentro del marco interior. A continuación, se disponen los rodamientos dentro de dicho marco y se introducen a presión el eje interior y el sensor de alabeo en el eje longitudinal.



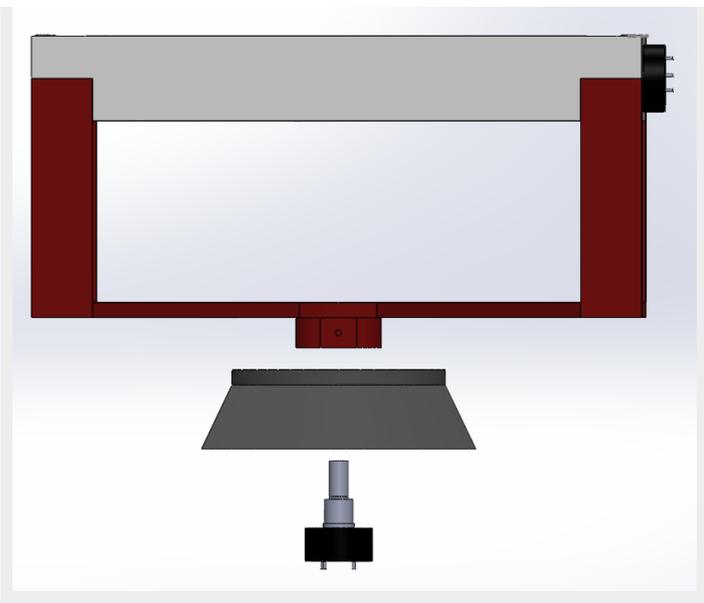
3. Se disponen los elementos anteriores dentro del marco exterior. Se dispone el rodamiento restante y se introduce a presión el eje y el sensor de cabeceo correspondiente al eje transversal.



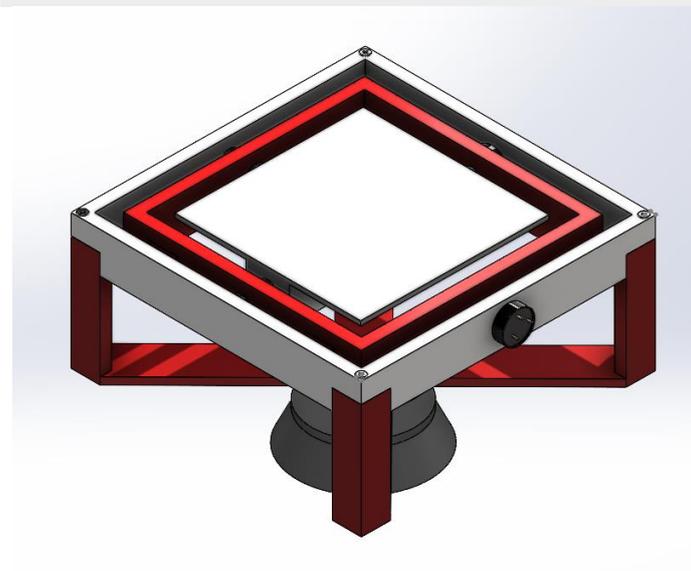
4. Se acoplan los pilares al marco exterior por encaje y se fija su posición mediante 4 tornillos M3



5. Se introduce el sensor correspondiente a la guiñada en la cavidad diseñada dentro la base para ese propósito. Se asegura su posición mediante una arandela de rosca.



6. Finalmente se junta el resto del mecanismo con la base mediante el sensor de guiñada que conecta



6.2 Esquema eléctrico

En este apartado se explicarán los elementos conectados a la placa de *Arduino DUE* y los pines asociados a estos.

6.2.1 Etapa de acondicionamiento

Antes de indicar los pines de conexión del montaje eléctrico, cabe destacar que es necesario el diseño de una etapa de acondicionamiento de la señal de salida de los potenciómetros magnéticos. Esto se debe a que su salida es de un rango de [0-5] V y los pines de entrada de entrada analógicos de la placa es de [0-3,3] V. Para ello se utiliza un divisor resistivo:

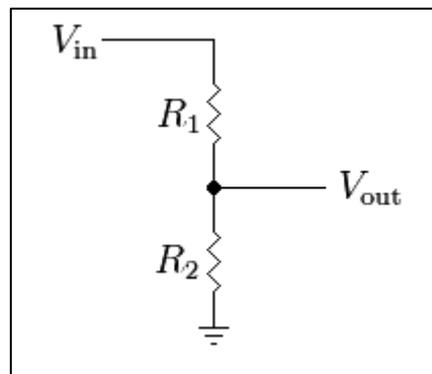


Figura 33: Divisor resistivo

Cuya salida se rige por la siguiente función de transferencia:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Siendo en este caso el valor de las resistencias $R_1 = 10k\Omega$ y $R_2 = 20k\Omega$, proporcionado los siguientes resultados:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2} = 5 \cdot \frac{20}{10 + 20} = 3,33 \text{ V}$$

Como se observa gracias a esta etapa se acondiciona la señal de salida correctamente consiguiendo una salida con una salida de [0-3,3] V.

6.2.2 Esquema de conexión

Una vez diseñada la etapa de acondicionamiento, los elementos conectados a la placa son principalmente los tres potenciómetros magnéticos que nos servirán para la lectura de los ángulos del mecanismo.

La alimentación de sensores se realiza a través de la alimentación desde la placa de *Arduino* gracias a la extensión de los carriles de alimentación de una placa *Board*. Y sus salidas se conectan directamente a los pines analógicos indicados en el esquema.

Además, cuenta con un LED para controlar el tiempo de ejecución del código implementado. De esta forma, si supera los 0.1s se encenderá indicando que se ha superado el tiempo de ejecución establecido.

La carga del código de *Arduino* y la comunicación serie con el entorno de *Simulink* se realiza a través de dos cables independientes de tipo *USB* que se conectarán al puerto *Programming* y *Native USB*.

Además, la alimentación de la placa *Arduino* vendrá dada por uno de los puertos USB sin necesidad de una fuente externa de alimentación, ya que esta deberá siempre estar conectada para llevar a cabo la comunicación serie.

De esta forma, el esquema del mecanismo es representado en la siguiente figura:

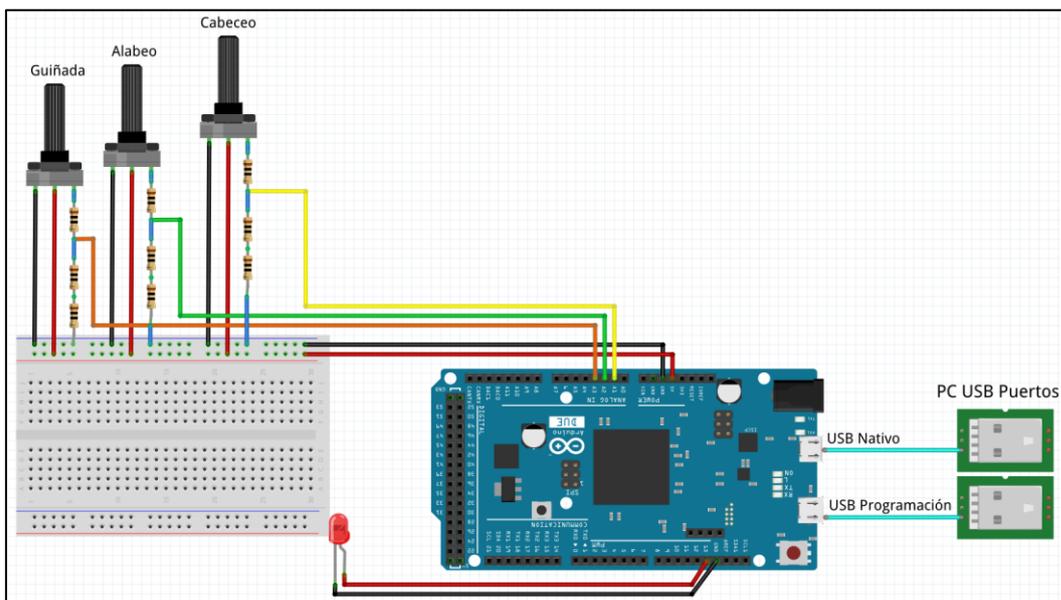


Figura 34: Esquema eléctrico de conexión

Los pines de conexión usados son los siguientes:

esta forma, los botones usados son:

- **Potenciómetro Cabeceo**
Positivo: 5V
Negativo: GND
Salida: Pin analógico A1 (cable amarillo)

- **Potenciómetro Alabeo**
Positivo: 5V
Negativo: GND
Salida: Pin analógico A2 (Cable verde)

- **Potenciómetro Guiñada**
Positivo: 5V
Negativo: GND
Salida: Pin analógico A3 (Cable naranja)

- **Led de temporización**
Positivo: Pin digital 13
Negativo: GND

- **USB de programación**
Puerto de programación

- **USB comunicación serie**
Puerto nativo

6.3 Prototipo de implementación

A continuación, se muestran el mecanismo después del montaje y con las conexiones explicadas en el punto anterior y sus distintas partes.



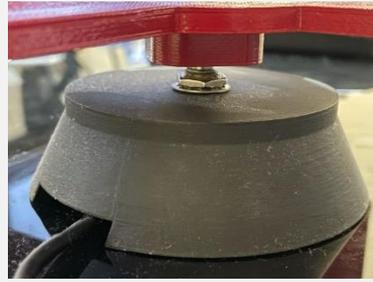
Figura 35: Mecanismo real con 3 ejes de libertad

Se comprueba que el movimiento de las articulaciones es el adecuado y que no chocan partes entre sí.

En la siguiente tabla se indican los sensores y su correspondencia con los ángulos de navegación a leer.

<p>SENSOR DE CABECEO</p>	
<p>SENSOR DE ALABEO</p>	

SENSOR DE GUIÑADA



Finalmente, con respecto al conexionado electrónico el resultado es el siguiente:

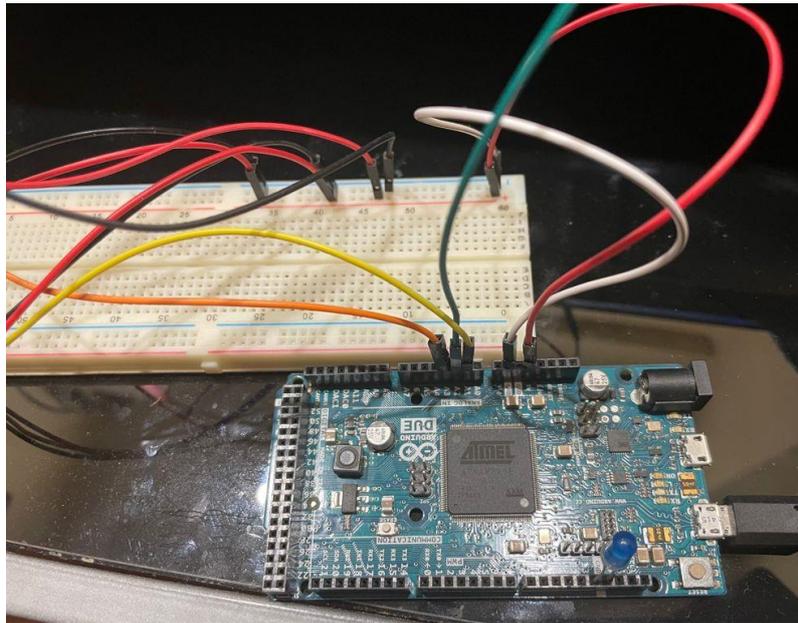


Figura 36: Conexiones implementadas

Los divisores resistivos se encuentran alojados dentro del cableado de cada uno de los sensores para ofrecer una señal de lectura adecuada (0-3'3V) para las entradas analógicas de nuestro *Arduino Due*.

6.4 Comunicación serie

Para establecer la comunicación serie entre el entorno de *Arduino* y *Simulink* se implementa por código y por medio de bloques programables, respectivamente.

El código implementado en *Arduino DUE* se divide en la lectura de los datos de los sensores del mecanismo físico y una función que permite establecer una comunicación serie entre el entorno de dicha plataforma y *Simulink*. El código se encuentra adjunto en los anexos de esta memoria.

En primer lugar, para la lectura de los sensores se realiza directamente a través de la función *analogRead()*. La resolución establecida es de 8 bits, ya que corresponde con el tamaño de paquete de datos que se envían a través del puerto serie.

Para la recepción de los datos leídos del mecanismo, se implementa una función comunicación serie entre *Arduino* y *Simulink*. Además, se incluye la posibilidad de realizar un envío de datos de forma bidireccional, es decir, de *Simulink* a *Arduino* también.

Estas órdenes se implementan en un bucle infinito que finaliza con una función de temporizado que asegura que cada ciclo del bucle sea del periodo establecido. En caso contrario, se encenderá un LED situado en el pin 13 para indicar que dicho periodo no se cumple.

En la figura siguiente, se muestra un diagrama de flujo del código implementado en *Arduino* que interactúa con el entorno de *Simulink*:

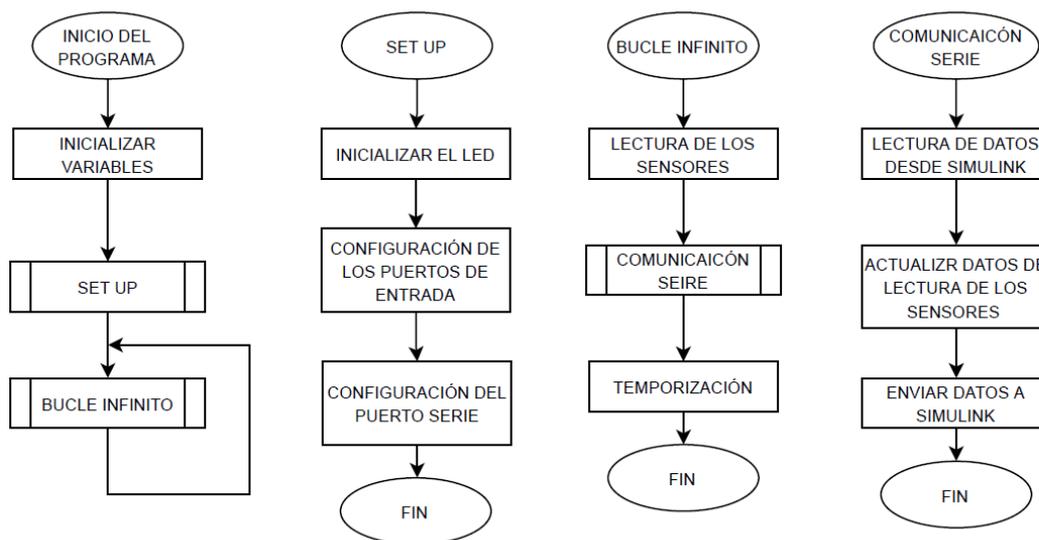


Figura 37: Diagrama de flujo del código de implementación

A continuación, se entrará más en detalle de la implementación de la comunicación serie vista desde ambos elementos, así como su configuración.

6.4.1 Entorno Arduino

Para establecer la comunicación serie en *Arduino* nos valdremos de las siguientes variables:

- Tiempo de muestreo
- Vector de almacenamiento para cada variable

- Variables de referencia para consignar las acciones

Respecto al tiempo de muestreo, se programa con un valor de 50 ms correspondiente al valor que se asignará en el entorno de simulación. De esa forma, cada paso del programa de simulación es equivalente al tiempo de cada bucle y no se perderán datos durante la transmisión.

Con ello, si el puerto de conexión USB está disponible se lleva a cabo la transmisión y recepción de datos tal como se representa en a la siguiente figura. En caso contrario, termina la comunicación serie enviando un mensaje de error “---XXX---” y comprobará de nuevo su estado en el siguiente ciclo.

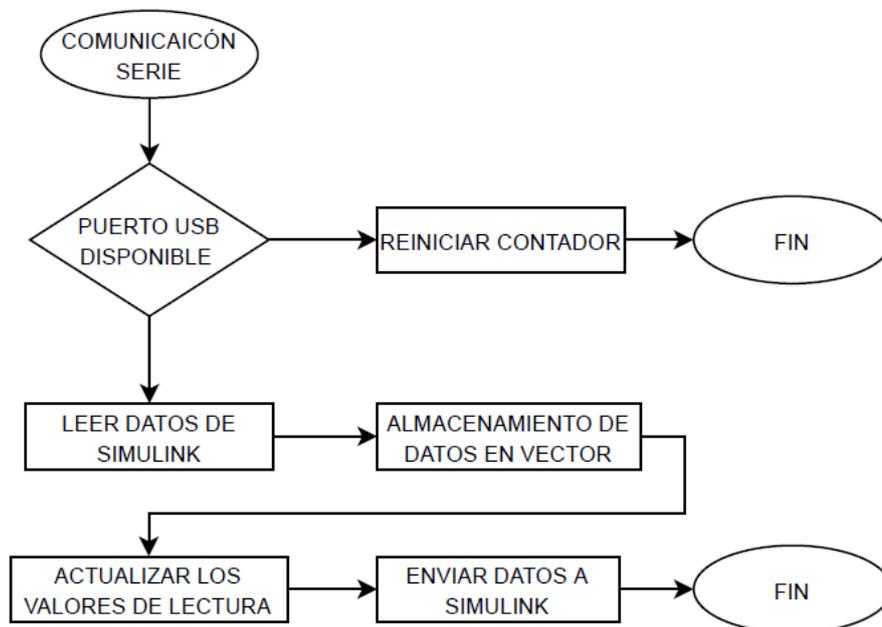


Figura 38: Diagrama de flujo de la Comunicación serie

En primer lugar, se da la recepción y luego la transmisión de datos. A continuación, se explican dichos procesos.

Recepción (Simulink-Arduino): se establece el número de bytes totales de la lectura y se agrupan en bloques de 1 byte (8 bits) correspondiente a cada variable definida. Para llevar a cabo la lectura, se usa la función *SerialUSB.read()*.

Transmisión (Arduino-Simulink): se envían las variables de 8 bits recogidas en la función de lectura. Al ser, de 8 bis no hace falta ninguna conversión ya que es el tamaño

de los paquetes de recepción y envío. La transmisión de los datos a Simulink se realiza mediante la función *SerialUSB.write()*.

6.4.2 Entorno Simulink

Desde el punto de vista de *MATLAB* es necesario implementar un esquema de *Simulink* para establecer la comunicación con *Arduino*. En este caso, el esquema es el siguiente:

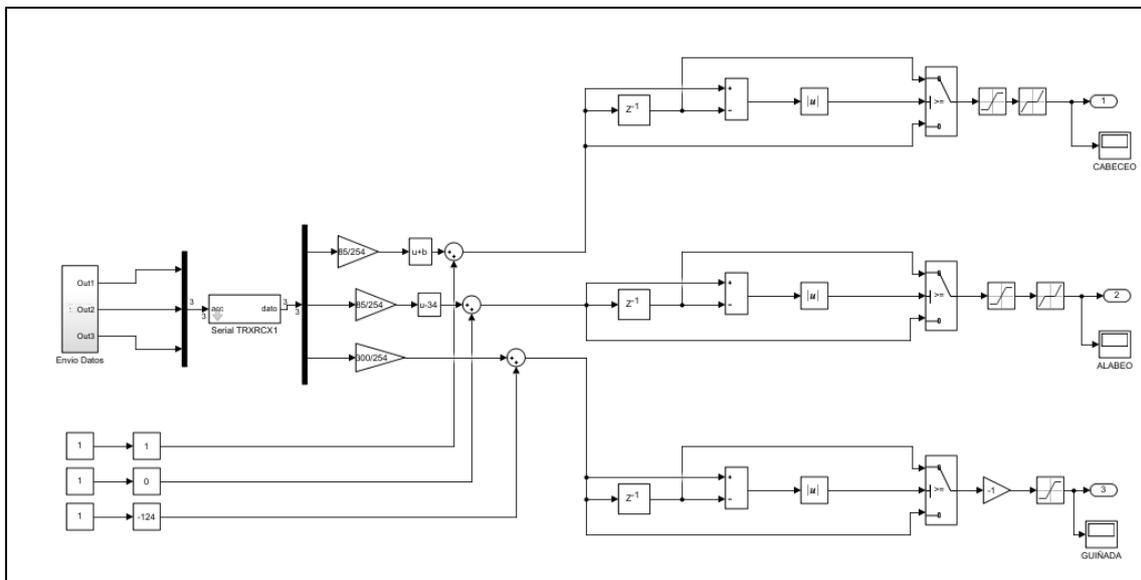


Figura 39: Simulink de la comunicación serie

El bloque que incluye la programación donde se configuran los parámetros de la comunicación serie es el bloque *S-Function* (dentro del bloque *Serial TRXRCX*). Siendo los parámetros de comunicación a configurar:

- Puerto de conexión: **COM4**
- Número de variables transferidas: **3**
- El periodo de comunicación: **0.1 s**

La transmisión y recepción de datos se produce de la siguiente manera:

Transmisión (Simulink-Arduino): Se introduce a la entrada los datos que se quieren enviar y posteriormente se aplican un escalado. Dicho escalado es necesario para adaptar los valores introducidos en una escala de 8 bits (255 valores) al igual que el tamaño de los paquetes que lee Arduino.

Recepción (Arduino-Simulink): en este caso los valores recibidos tienen una resolución de 8 bits por lo que a su salida se realiza una conversión. Gracias a la conversión, se adapta los datos recibidos de los sensores y se pueden expresar en grados. Además, se

dispone de un offset por medio de Sliders para determinar el punto de origen o referencia 0 de los ángulos. A la salida cuenta con un posprocesado de los ángulos por el cual se evitarán falsas lecturas. Este se basa en el que si se da una diferencia de ángulo entre el instante actual y el anterior demasiado grande (a elección) tomará como valor válido el anterior. Estos valores directamente en grados serán los que den la posición angular del mecanismo y se transfiera a las articulaciones de la simulación y den origen a la reproducción del movimiento real.

6.5 Simulador de vuelo

Para la construcción del simulador de vuelo nos valdremos de *Simulink* y los bloques tratados en el apartado 5.2.1 *Bloques de Simscape Multibody* de este proyecto.

Para su explicación se ha dividido en 4 subapartados correspondientes al modelo del cuadricóptero, ejes de libertad para su desplazamiento, implementación de las fuerzas y el entorno en el que se moverá el cuadricóptero.

6.5.1 Modelo del cuadricóptero implementado

En primer lugar, se ha importado un modelo de un cuadricóptero (proporcionado por el tutor) por partes parecido al real. Los bloques de *Solid* se han conectado guardando las relaciones de posición, siendo los elementos que lo componen mostrados en la siguiente figura:

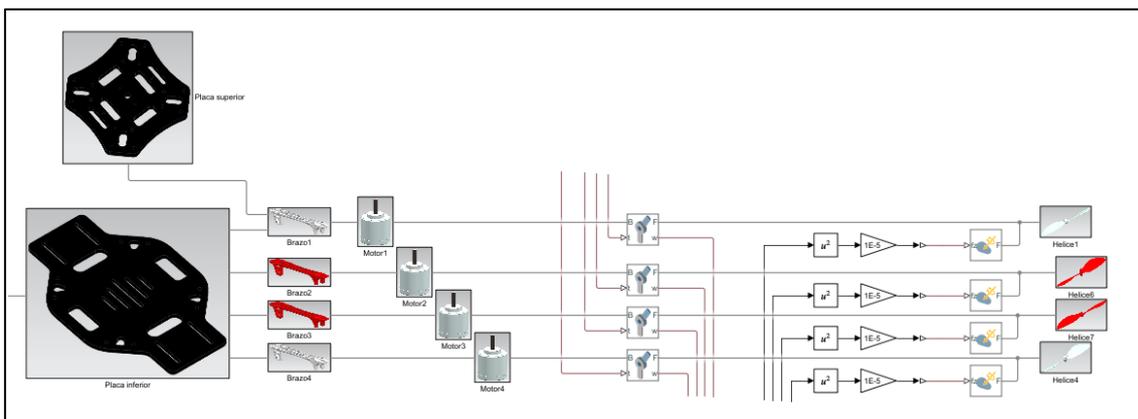


Figura 40: Simulink del modelo del cuadricóptero

Dicho modelo será el que se visualice en la simulación y reproduzca los movimientos del cuadricóptero real sobre la plataforma. Este modelo al igual que el real dispone de 4 motores y hélices con una configuración en X, al que se le han atribuido características

similares al real como el peso para que la simulación sea lo más fidedigna posible. Dando como resultado el siguiente modelo:

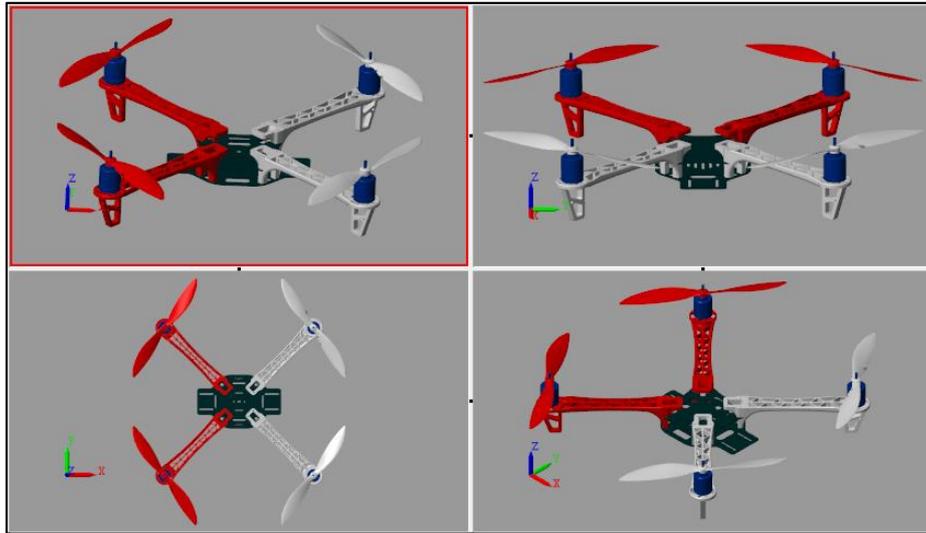


Figura 41: Cuadricóptero de simulación

6.5.2 ejes de desplazamiento y rotación

Tanto para la rotación como para el desplazamiento en X e Y se han implementado articulaciones en serie asignadas todas al mismo punto: el centro de la base inferior del cuadricóptero con respecto al mundo.

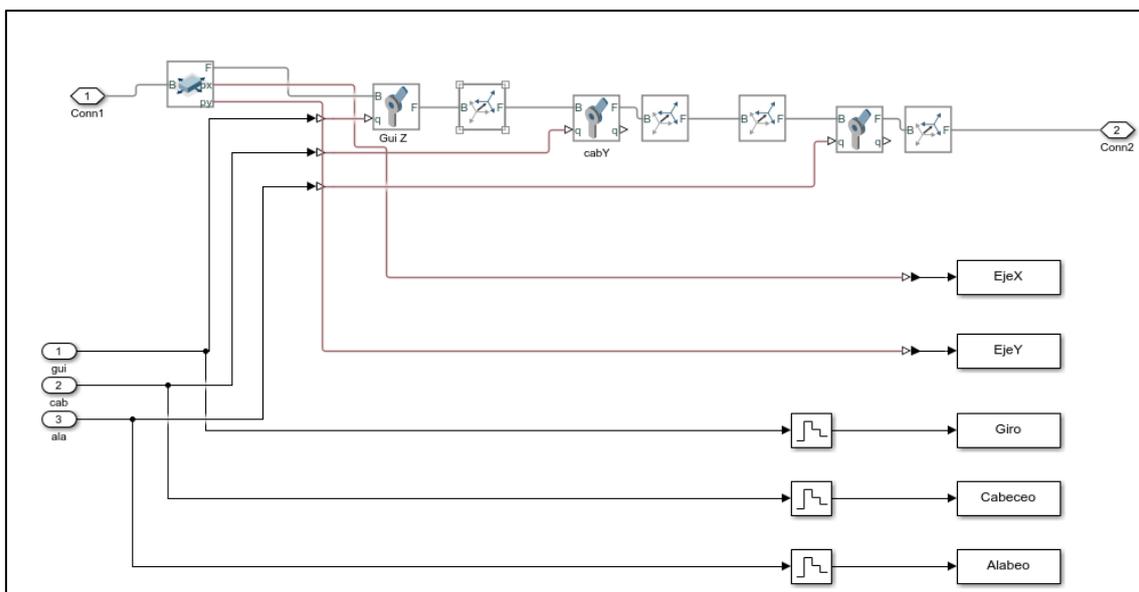


Figura 42: Simulink de las articulaciones

De esta forma, se establece una articulación rectangular que permite dotarle de dos grados de libertad extra que permiten el desplazamiento del cuadricóptero por medio del entorno en los ejes x e y .

Para los ejes de rotación correspondientes al alabeo, cabeceo y guiñada se asignan articulaciones de revolución que permiten la rotación del cuadricóptero en los ejes X , Y y Z respectivamente. Los pares de rotación de forma predefinida solo se pueden establecer en el eje Z del marco de coordenadas al que está conectado, por ello, para su implementación es necesario el uso del bloque *Regid Transform* y aplicar una rotación entre los pares de revolución y redefinir dicho marco. Asimismo, dichas articulaciones serán forzadas a girar con los valores que se reciban a través de la comunicación serie y así reproducir el movimiento del cuadricóptero real.

Además, se disponen de estas a través del bloque *To workSpace* para poder graficarlas y obtener los resultados de la simulación.

6.5.3 Fuerzas que intervienen en el movimiento

La implementación de las fuerzas en el cuadricóptero de simulación es análoga a la implementación que se usó con la simulación del mecanismo. Siendo el resultado de su implementación la siguiente:

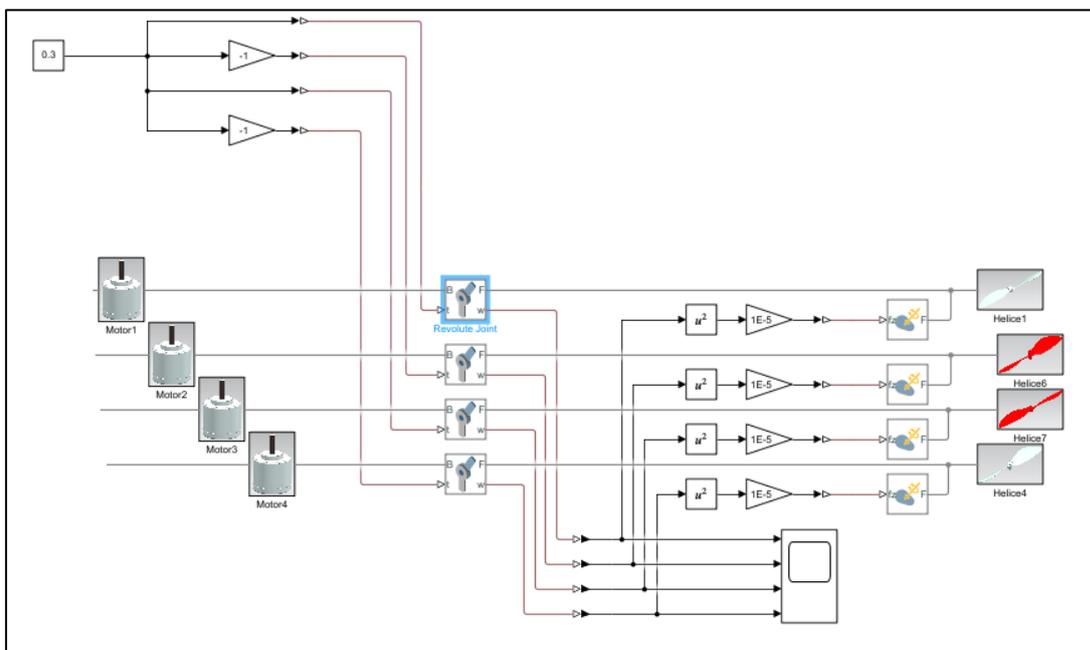


Figura 43: Simulink en la implementación de fuerzas

En este caso concretamente se introducirá un par de fuerzas constante en las articulaciones de revolución que unen los motores con las hélices. Estos producirán el giro y un par de fuerza para cada una de las hélices, que a su vez generarán un momento resultante en el centro de masas del cuadricóptero.

En este punto es importante establecer un coeficiente de rozamiento mínimo en los pares de revolución entre las hélices y los motores, siendo de $0.00001 N \cdot m$. En caso contrario, al no oponerse ningún par de fuerzas al par constante introducido se produciría una velocidad con una aceleración contante que tendería al infinito. De esta forma, se establecen velocidades contantes durante toda la simulación.

Por otra parte, el cálculo de las fuerzas lineales se realiza a través de las velocidades angulares que proporcionan las salidas de las articulaciones. Al igual que en el punto 5.3 “*Condiciones de fuerza*” nos valemos de la misma conversión para el cálculo de dichas fuerzas.

Partiendo de estas condiciones de fuerza y velocidades constantes en las hélices de nuestro cuadricóptero, el desplazamiento del cuadricóptero por el entorno de simulación en los ejes X e Y se producirá por un desequilibrio entre estas provocado por el ángulo de alabeo y/o cabeceo de lectura cuando sea distinto de 0° .

6.5.4 Entorno del simulador de vuelo

Para dar más realismo a la simulación del cuadricóptero, se ha construido un entorno de simulación. Esta cuenta principalmente por una base sobre la cual se desplazará el cuadricóptero, límites y una cámara que ofrecerán un punto de vista en primera persona.

La base está compuesta por 4 bloques de distinto color que permiten distinguir visualmente el desplazamiento del cuadricóptero y 4 paredes, delimitando así la zona de vuelo. Dichos bloques se han implementado por medio del bloque *Solid* con unas medidas de $4 \times 4 m$. Siendo el total de la superficie de desplazamiento de $16 m^2$.

En este caso, se han establecido los límites del entorno mediante el bloque “*Rectangular Joint*” que permite el desplazamiento del cuadricóptero con unos límites coincidentes las medidas de la superficie asignadas. De esta forma, el cuadricóptero solo se podrá desplazar sobre la base.

De esta forma, el entorno de simulación se visualizará de la siguiente manera:

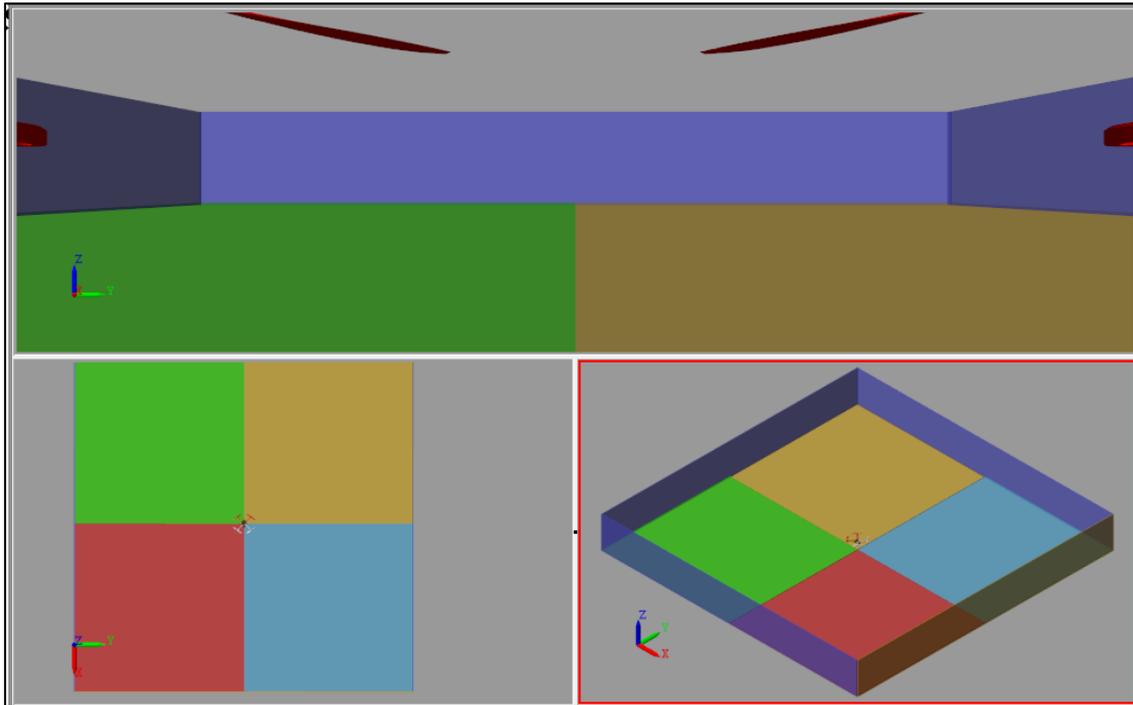


Figura 44: Entorno de simulación y perspectivas

En la zona superior cuenta con una cámara en primera persona y la zona inferior se subdivide en dos: a la izquierda la vista superior y a la derecha vista isométrica.

Asimismo, la configuración de los parámetros para la simulación establecidos son los siguientes:

- **Configuración del tiempo de simulación**
 Inicio: 0 segundo
 Final: infinito
- **Configuración del método de resolución (solver)**
 Tipos de resolución: Paso fijo
 Método de resolución: discreto
 Tiempo de paso: 0,1 segundo
- **Configuración del tiempo de ejecución**
 Habilitar el modo de “Run” para la simulación
 Tiempo de simulación por segundo del reloj: 1 (simulación en tiempo real)

7. Comprobaciones de la implementación

Un proceso paralelo a la implementación del mecanismo se realizaron comprobaciones del mecanismo y aplicación para determinar su alcance. Se destacan principalmente: lectura de los sensores, movimiento del mecanismo real y simulado.

7.1 Lectura de los sensores

En el primer caso, se implementó en *Arduino* un programa previo al de comunicación serie para comprobar la lectura de los sensores mediante las entradas analógicas correspondientes. A partir de este se concluyó que el rango de tensión de la señal de salida, tras pasar por la etapa de acondicionamiento, que ofrecían abarcaba un 94% de los 3,3V posibles, lo que se traducía en una lectura inferior a la total de los sensores. Considerando que se escogió una resolución de 8 bits para las entradas analógicas se dispone de unas 255 cuentas, de las cuales se pierden el 6% restante debido a la tensión de salida de los sensores, siendo las cuentas disponibles 238.

Para el cálculo de la resolución en ángulos basta con dividir el rango en ángulos que abarca el sensor entre el número de cuentas disponibles.

De esta manera el rango que abarcan los sensores y su resolución en ángulos disponibles son:

Sensor	Rango (°)	Resolución (°)
Cabeceo /Alabeo	85°	0,35°
Guiñada	338°	1,42°

Estos rangos no presentan ningún inconveniente para nuestra aplicación, puesto que para el alabeo y cabeceo como máximo un rango de 30° correspondientes a los $\pm 15^\circ$ que aproximadamente necesita un cuadricóptero para desplazarse. Respecto a la guiñada por la construcción física y el cableado tampoco se pretenderá realizar giros de más de $\pm 90^\circ$.

Cabe recordar que se escogió una resolución de 8 bits para las entradas analógicas, ya que en la comunicación serie de *Arduino* se envían tramas en paquetes de bytes. Por lo que aumentar la resolución suponían duplicar el número de bytes a enviar y por ende supondría un problema a la hora de poder ejecutar de forma mínimamente fluida la simulación en tiempo real.

Durante las comprobaciones de lectura se observó o como a veces la señal de entrada variaba para un mismo movimiento, esto se debía a la alimentación de los sensores. Por ello, se usó una fuente externa independiente que alimentase los 3 sensores y así solventar el problema.

7.2 Movimiento mecánico del mecanismo

La segunda parte de las comprobaciones hace referencia a la capacidad del mecanismo real de moverse de forma adecuada para el propósito para la que fue diseñada.

Se acopló el cuadricóptero a usar y se realizaron pruebas de vuelo sobre la plataforma para verificar los resultados de dinámica del modelo simulado y que el cuadricóptero pueda volar de forma natural sobre la plataforma y a su vez transmitir este movimiento a las articulaciones del mecanismo diseñado.

Debido a los brazos del cuadricóptero y el bastidor exterior de la plataforma el rango de movimiento para el alabeo y cabeceo se establece en $\pm 15^\circ$. Siendo un rango aceptable para estos movimientos, ya que en un vuelo real se usan pequeños ángulos de alabeo y cabeceo para desplazarse.

Asimismo, un dato a recalcar es que, dependiendo de las fuerzas empleadas en la elevación del cuadricóptero, los movimientos de alabeo y cabeceo producen guiñada en la plataforma. Es importante ajustar adecuadamente la velocidad de las hélices para evitar este problema.

Cabe señalar que controlar un cuadricóptero no es tarea sencilla por lo que requiere cierta habilidad, puesto que entran en juego las dinámicas del mecanismo y el cuadricóptero con un control manual y coordinación mano-ojos. Por ello, tras las pruebas de movimiento se determinó que la dinámica del mecanismo y el cuadricóptero era el adecuado, sin producirse un movimiento errático durante el control del mismo.

7.3 Movimiento real y simulado

Una vez comprobada la lectura de los sensores y que el movimiento del mecanismo era el adecuado para la aplicación, la última comprobación consistía en la relación entre ellos y el movimiento simulado que se podía generar a partir del real.

Para este propósito, se incorporó el modelo del cuadricóptero de la simulación y se implementaron indicadores de los ángulos de cabeceo, alabeo y guiñada situados en la izquierda, derecha y centro abajo, respectivamente.

Evidencias de estas se encuentran en videos adjuntos a este proyecto que verifican lo expuesto en este punto. A continuación, se muestran capturas de las pruebas realizadas correspondientes al alabeo, cabeceo y guiñada

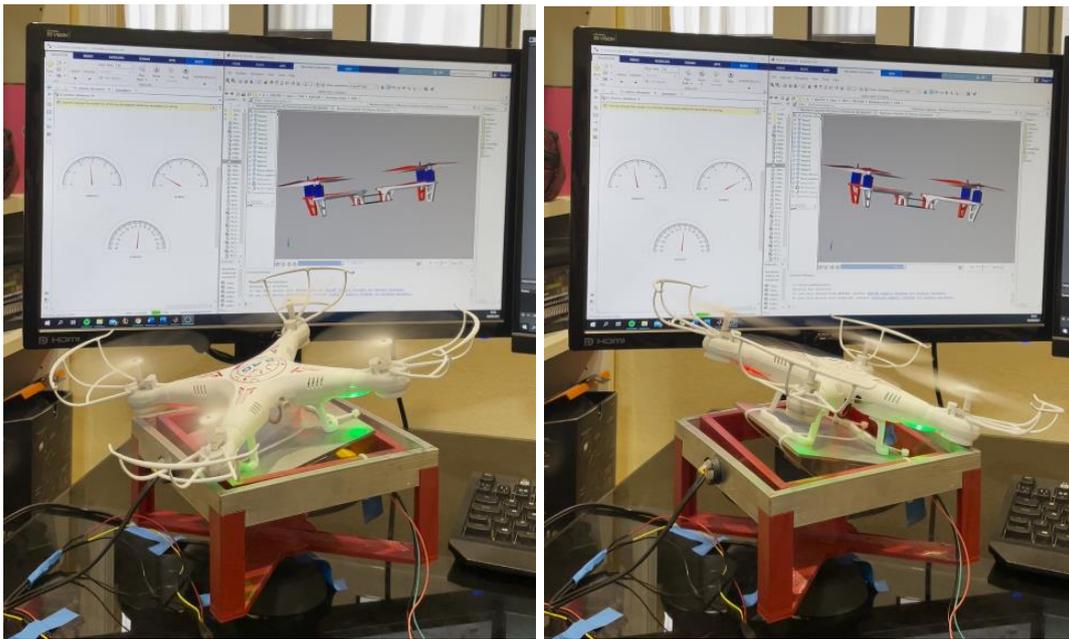


Figura 45: Movimiento de Alabeo

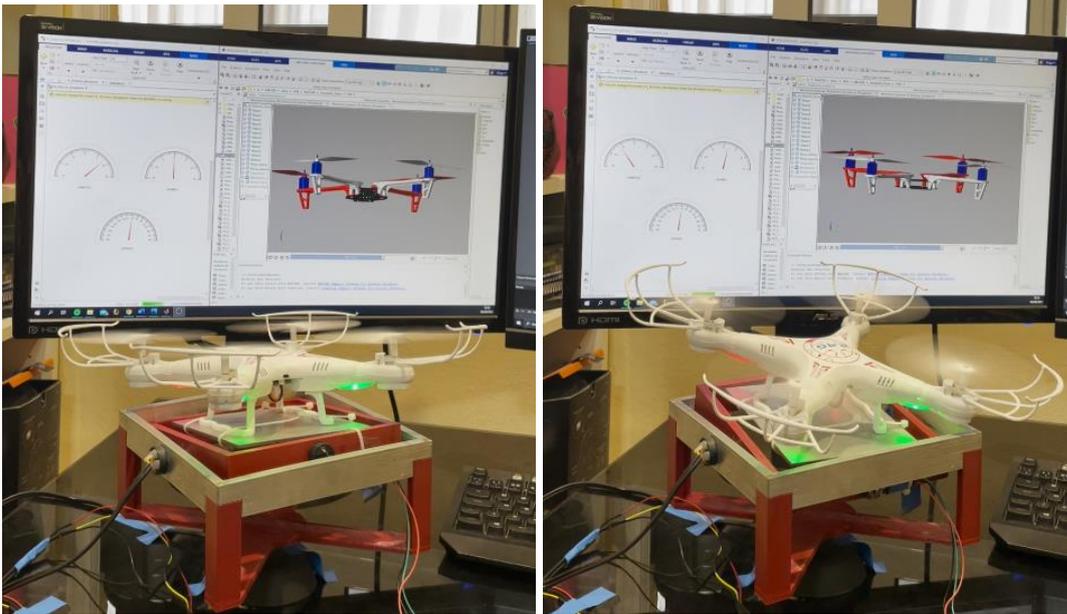


Figura 46: Movimiento de Cabeceo

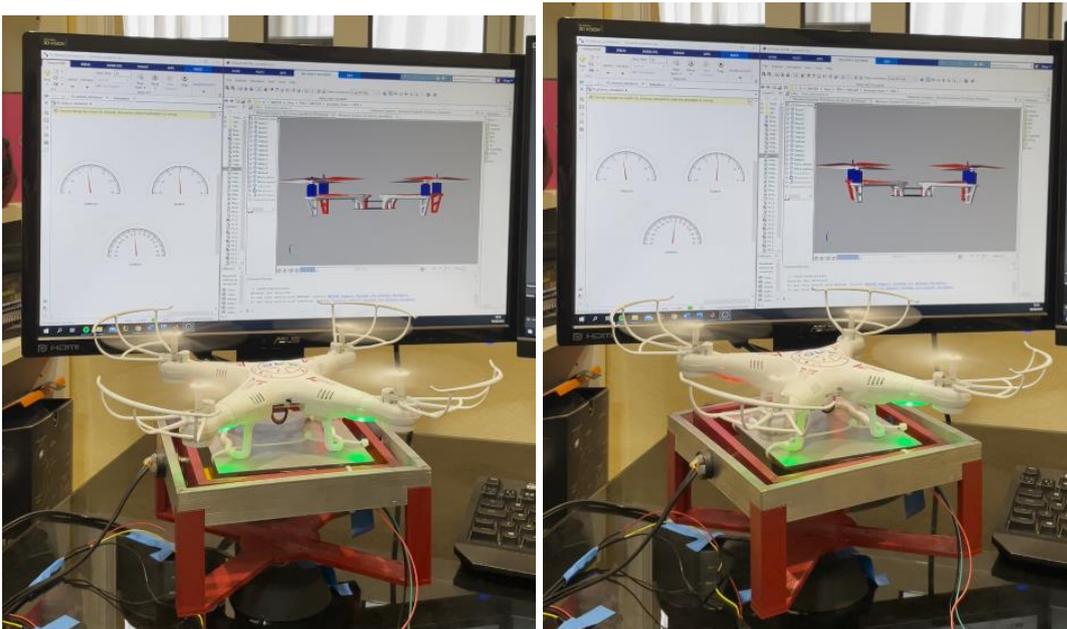


Figura 47: Movimiento de Guiñada

Se puede comprobar cómo tanto el movimiento como la lectura es adecuado y se puede reproducir los movimientos del cuadricóptero en tiempo real. Asimismo, cabe añadir que la fluidez de la simulación depende de la potencia del ordenador usado, ya que se observó cambios de fluidez en dos equipos distintos.

8. RESULTADOS

Tras realizar las comprobaciones de la implementación del mecanismo real y determinar la viabilidad de este, se realiza una demo de vuelo. Esta demo consiste en realizar una simulación del cuadricóptero desplazándose sobre el entorno de forma controlada y realizar una trayectoria en el que se evidencie los distintos tipos de movimientos y así, poner en manifiesto el cumplimiento del objetivo inicialmente propuesto.

Antes cabe comentar que, los problemas surgidos durante su realización fueron solventados durante su implementación y han sido comentados en sus respectivos apartados del punto *6 Implementación del sistema real* y *7 Comprobaciones*.

A continuación, se presenta una secuencia correspondiente al video adjunto en la presentación de este proyecto:

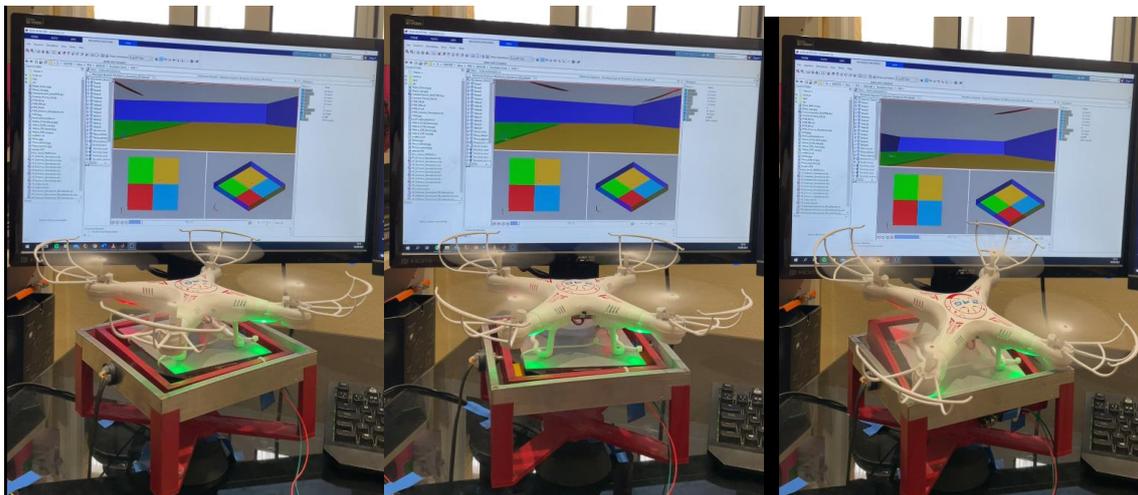


Figura 48: Video demo Secuencia 1

En la primera secuencia se realiza una guiñada en ambos sentidos y un movimiento de cabeceo negativo (hacia atrás).

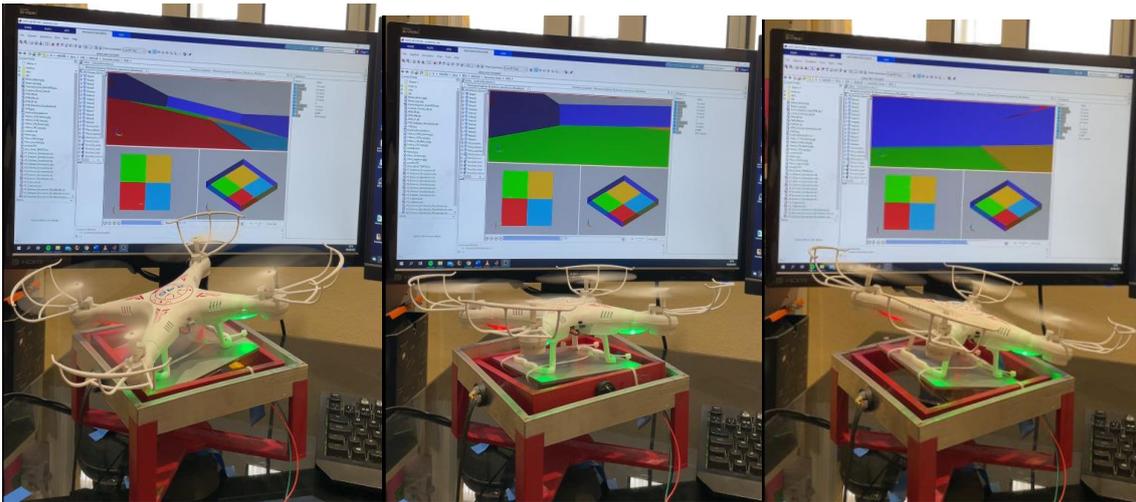


Figura 49: Video demo Secuencia 2

En la secuencia 2 se realiza un movimiento de alabeo negativo (izquierda) y un movimiento de cabeceo positivo (delante) combinado con un movimiento de alabeo positivo (derecha).

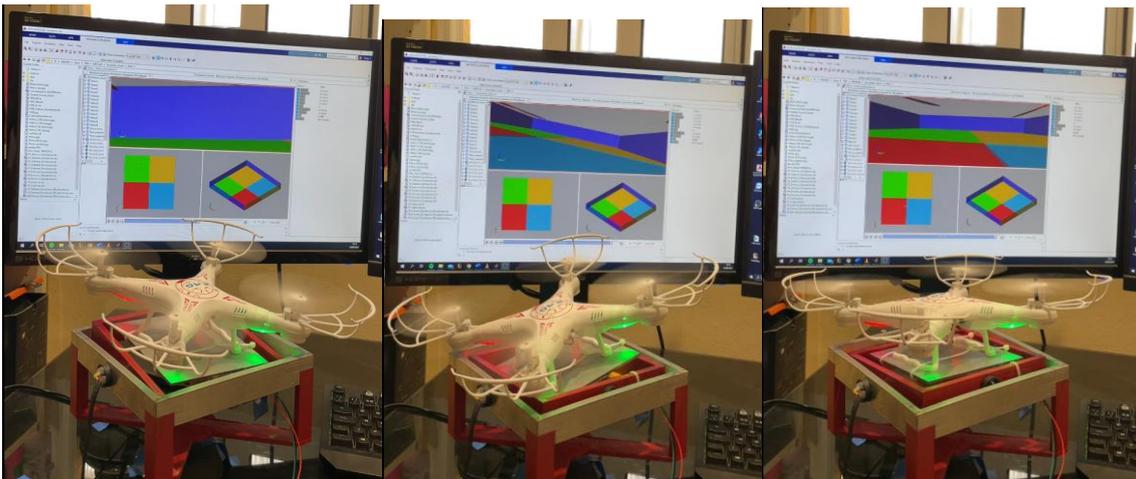
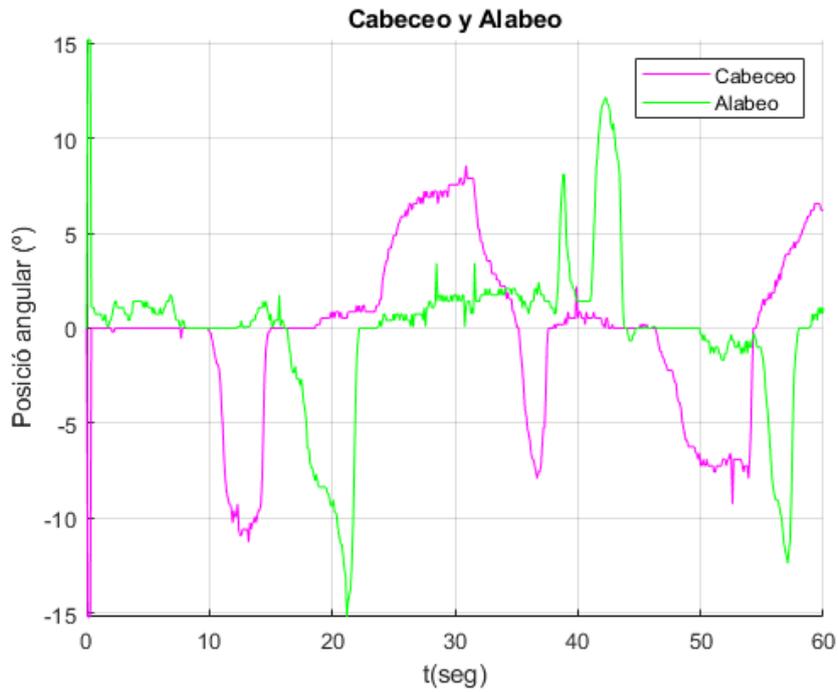


Figura 50: Video demo Secuencia 3

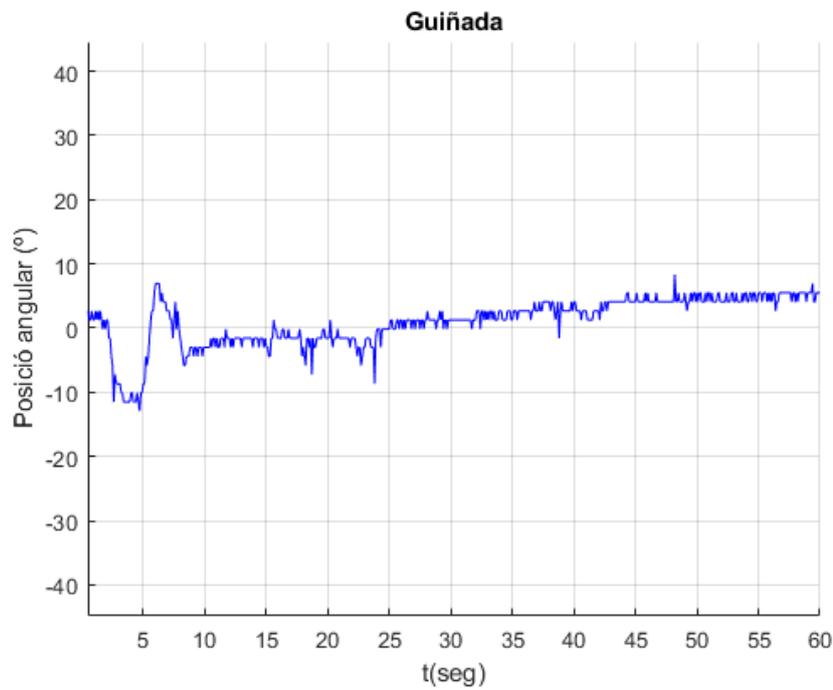
Finalmente, en la secuencia 3 se vuelve al punto de retorno con un movimiento de cabeceo negativo, alabeo negativo y cabeceo positivo, respectivamente.

Asimismo, gracias al entorno de simulación se pueden extraer los datos del vuelo y representar tanto los ángulos de navegación como el desplazamiento realizado y poder almacenar estos datos.

De esta forma, los ángulos de navegación obtenidos para la simulación son:

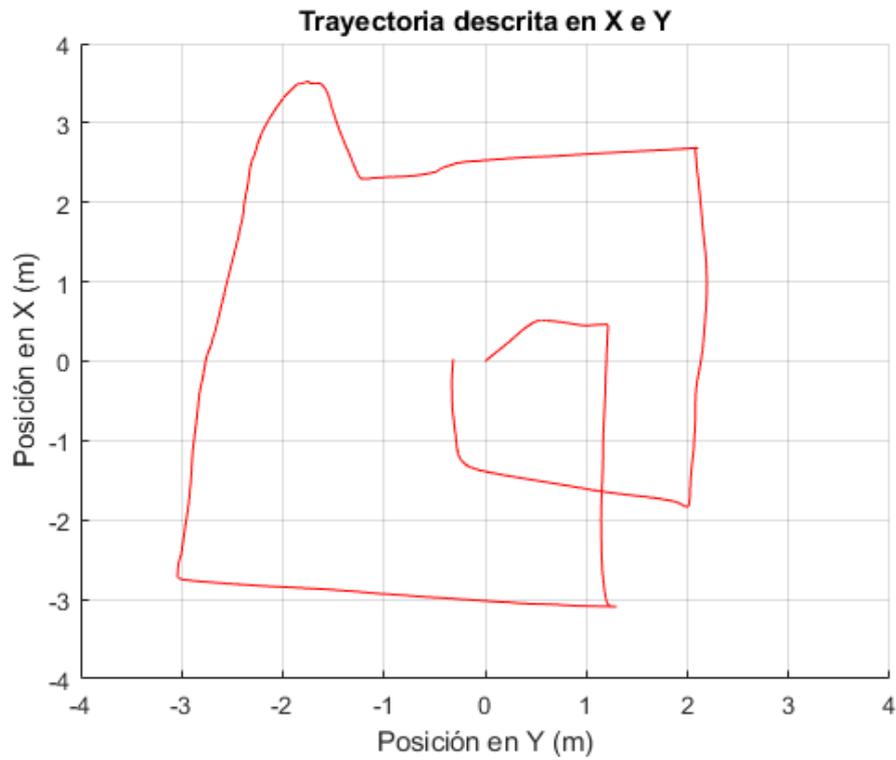


Gráfica 7: Video demo ángulos de Alabeo y Cabeceo



Gráfica 8: Video demo ángulo de guiñada

La correspondencia de los ángulos de navegación trasladados al desplazamiento y a la trayectoria del cuadricóptero viene indicada en la siguiente gráfica:



Gráfica 9: Video demo trayectoria descrita

De esta forma, la trayectoria descrita se asemeja a una forma poligonal donde se recorre el entorno en sentido antihorario y donde se han realizado los 3 tipos de movimientos implementados: cabeceo, alabeo y guiñada.

9. CONCLUSIONES Y RECOMENDACIONES

Tras la realización del presente Trabajo Final de Máster se puede dar por cumplido el objetivo principal de este: la construcción de un mecanismo capaz de medir la actitud de un cuadricóptero. Asimismo, se han añadido elementos útiles como es la incorporación de un entorno para la simulación donde poder visualizar en tiempo real el movimiento y la trayectoria del cuadricóptero de forma visual. Además, permite almacenar y graficar los datos del vuelo como son los ángulos de navegación y su trayectoria.

Durante su realización, se ha podido comprender la dificultad que representa materializar una idea. Es un proceso laborioso que requiere de planificación y marcar unos objetivos iniciales o lo que es lo mismo, el mínimo producto viable. Así se partió de la idea inicial de sensorizar una plataforma con 2 articulaciones correspondientes al alabeo y cabeceo. Por ello, tras estudiar la situación y a través de prueba y error se determinó que era viable añadir el 3 grado de libertad correspondiente a la guiñada. A esto se decidió incluir un entorno de simulación para hacer más visibles los resultados.

Por un lado, respecto al diseño mecánico se destaca la parte de diseño CAD e impresión 3D, donde se encontraron dificultades a la hora de obtener las piezas deseadas. Por ejemplo, en la impresión de los pilares, debido a que su estructura no era muy compatible con el proceso de impresión 3D, se tuvo que rediseñar estos y añadirles soportes para que durante el proceso de impresión los pilares no se deformaran.

En cuanto a la simulación, se comprendió las limitaciones de los equipos a lo que a hardware se refiere. La resolución de los sensores se estableció en 8 bits ya que la comunicación serie en Arduino se produce por paquetes de bytes. Por ello, aumentar la resolución significaría añadir un byte más, es decir, duplicar la información, lo que comprometería la fluidez de la simulación.

Antes de finalizar este apartado, echando la vista atrás se proponen las siguientes mejoras:

- La posibilidad de añadir a la articulación de guiñada un desacople que permita medir la altitud del cuadricóptero y así tener a disposición una lectura total y poder realizar una simulación en un entorno en 3D

- Buscar un chasis vez más resistente. Esto se debe a que las piezas de plástico presentan problemas a la hora de mecanizarlas. El hecho de atornillar algunas de las partes ha generado pequeñas grietas.
- Realizar una modificación en el diseño para permitir el cambio de algunos de sus elementos de forma más sencilla. Es el caso de los sensores de alabeo y cabeceo, ya que si sufren daños hay que desmontar casi el mecanismo en su totalidad.

En general, ha sido un proyecto multidisciplinar que abarca todas las fases de diseño de un nuevo producto: concepción de la idea, diseño, simulación, construcción, implementación y comprobaciones para determinar su alcance. Donde se han puesto en práctica conocimientos adquiridos en Dinámica de Sistemas Mecánicos para su diseño en un programa de CAD; Sistemas de Medición y Actuación para la sensorización; y Control Aplicado a Sistemas Mecatrónicos para la simulación del modelo y la implementación en su conjunto. Otorgando una experiencia en

10. BIBLIOGRAFÍA

Documentación

[1] [MathWorks](#)

- ✚ MATLAB
- ✚ Simulink

[2] [Arduino](#)

- ✚ Especificaciones técnicas
- ✚ Programación

Páginas WEB (accedido el 06/09/2021)

[3] One Air, *Normativa de drones en España 2021*
<https://www.oneair.es/normativa-drones-espana-aea/>

[4] Mancloc, Estefania (2017). *Física de un quadróptero*
http://documentacio.vedrunacatalunya.cat/recerques//treballs/2017/tarrega/TR_estefania_mancioc.pdf

[5] García Muñoz, Francisco Javier (2015). *Técnicas de control de vuelo sobre cuadricóptero*.
https://e-archivo.uc3m.es/bitstream/handle/10016/23041/TFG_Francisco_Javier_Garcia_Munoz_2015.pdf?sequence=1&isAllowed=y

[6] Embry-Riddle Aeronautical University (2019). A thrust equation treats propellers and rotors as aerodynamic cycles and calculates their thrust without resorting to the blade element method
<https://commons.erau.edu/cgi/viewcontent.cgi?article=1427&context=ijaaa>

Apuntes

[7] Apuntes de la asignatura de *Control Aplicado a Sistemas Mecatrónicos* impartida por el departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Valencia.

11. ANEXOS

11.1 Código de implementación Arduino

```

//////////DEFINICIÓN DE CONSTANTES//////////
#define RT 13 //Puerto del Led de aviso
#define MIN -45
#define MAX 45

//////////VARIABLES//////////

//Temporización
long t1 = 0 ;
long t2 = 0 ;
int T = 100 ; // Periodo de muestreo
milisegundos

//Comunicación Simulink
byte datos[4]={0,0,0,0} ;
float d1, d2, d3, d4 ;

//Sensores
int Sensor_Cabeceo = 0;

int Sensor_Alabeo = 0;

int Sensor_Giro = 0;

//////////SET UP//////////
void setup() {
    Serial.begin(115200) ;
    SerialUSB.begin(2000000) ;
    analogReadResolution(8); // Entrada analógica 8 bits (0-255)
    pinMode(RT,OUTPUT) ; // Inicializa el pin 13 del LED de
prueba como salida
} //Fin SET UP

//////////BUCLE INFINIT//////////
void loop() {

    //////////Lectura de los sensores
    Sensor_Cabeceo = analogRead(A1); //Lectura del
sensor en el puerto A0 para el Cabeceo
    Sensor_Alabeo = analogRead(A2); //Lectura del
sensor en el puerto A1 para el Alabeo
    Sensor_Giro = analogRead(A3); //Lectura del
sensor en el puerto A2 para el Alabeo

    //////////Comunicación serie
    if (SerialUSB.available()>=1) { //Comprueba si la conexión serie está
disponible
        int bb=0 ; //variable de referencia para comprobar los bits de
información

        //LECTURA DE LOS DATOS
    }
}

```

```

while (bb!=255 && bb!=-1) { // Lee el puerto serie de simulink
hasta que en el dato se recibe un 255 y recibe parada de lectura
  bb=SerialUSB.read() ;}

  if (SerialUSB.available()>=4) { //Se lee el puerto serie 4 veces y
se almacenan en vectores de 8 bits
    datos[0]=SerialUSB.read() ;
    datos[1]=SerialUSB.read() ;
    datos[2]=SerialUSB.read() ;
    datos[3]=SerialUSB.read() ;
  }

while (bb!=-1) {
  bb=SerialUSB.read() ;}

//Se produce el escalado de los datos recibidos de [0-254] a lo
que se desea y se pone a disposición para poder trabajar con ellos
d1=map(datos[0],0,254,MIN*100,MAX*100)/100.0 ;
d2=map(datos[1],0,254,MIN*100,MAX*100)/100.0 ;
d3=map(datos[2],0,254,MIN*100,MAX*100)/100.0 ;
d4=map(datos[3],0,254,MIN*100,MAX*100)/100.0 ;

//ARDUINO-SIMULINK (Datos que se envían desde los sensores de
forma directa)
//Manda los resultados y se revierte la escala aplicada
anteriormentete a [0 254], ya que se lee y se envía en 8 bits

SerialUSB.write(255) ; //Aviso del flag para empezar a escribir

//SerialUSB.write(map(100*d1,MIN*100,MAX*100,0,254));
//SerialUSB.write(map(100*d2,MIN*100,MAX*100,0,254)) ;
//SerialUSB.write(map(100*d3,MIN*100,MAX*100,0,254)) ;
//SerialUSB.write(map(100*d4,MIN*100,MAX*100,0,254)) ;

SerialUSB.write(Sensor_Cabeceo);
SerialUSB.write(Sensor_Alabeo);
SerialUSB.write(Sensor_Giro);
SerialUSB.write(map(100*d4,MIN*100,MAX*100,0,254)) ;

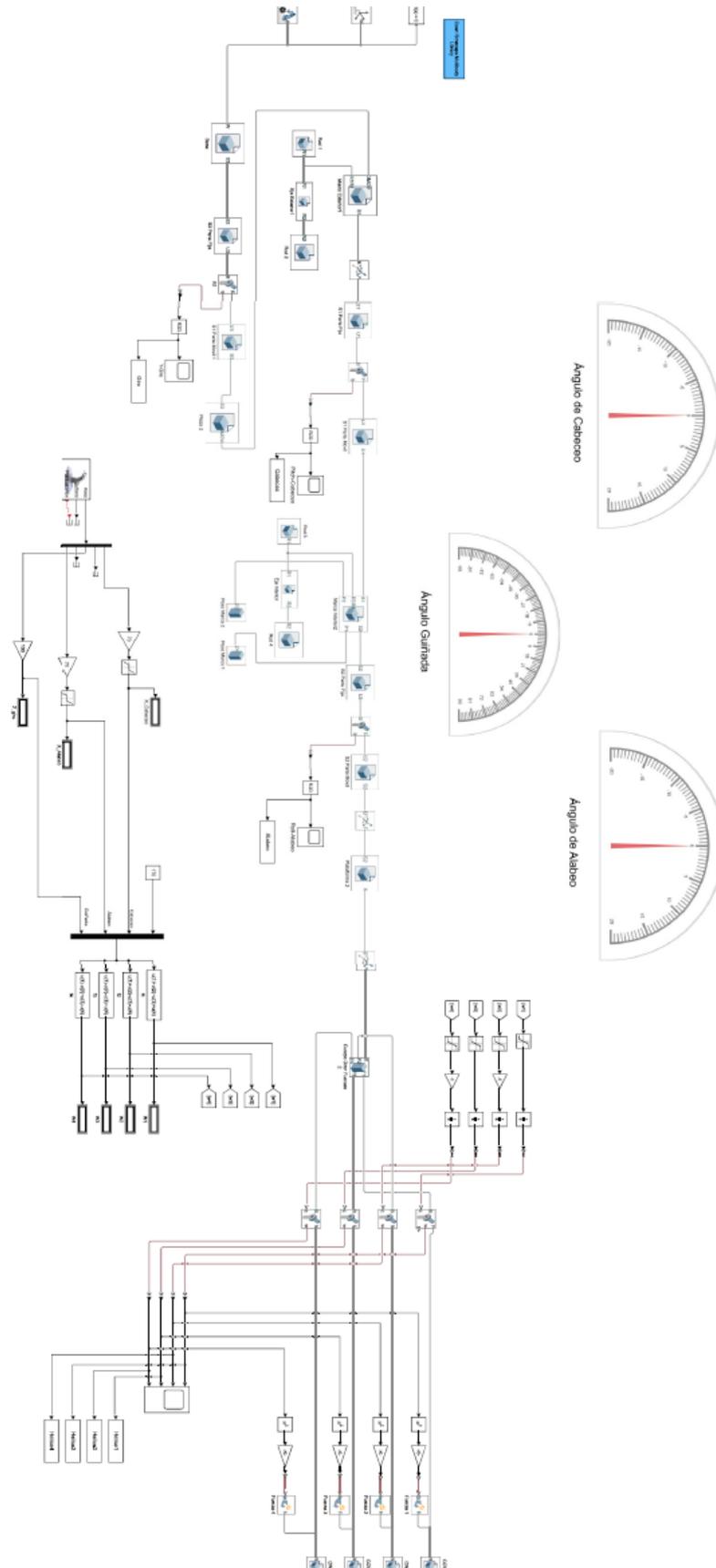
} else {Serial.println("---XXX---") ;}

//////////Temporización del ciclo
t2=millis() ;
if (t2-t1>T) { //Condicional para determinar si se supera el timepo,
si es igual al T programado avanza
  digitalWrite(RT,HIGH) ; Serial.print(t2-t1) ;}
else {digitalWrite(RT,LOW) ;}
while (t2-t1<T) { //Bucle que hace que se cumpla el timepo
  t2=millis() ;}
t1=millis() ; // se guarda el valor de t1 para el siguiente ciclo
}

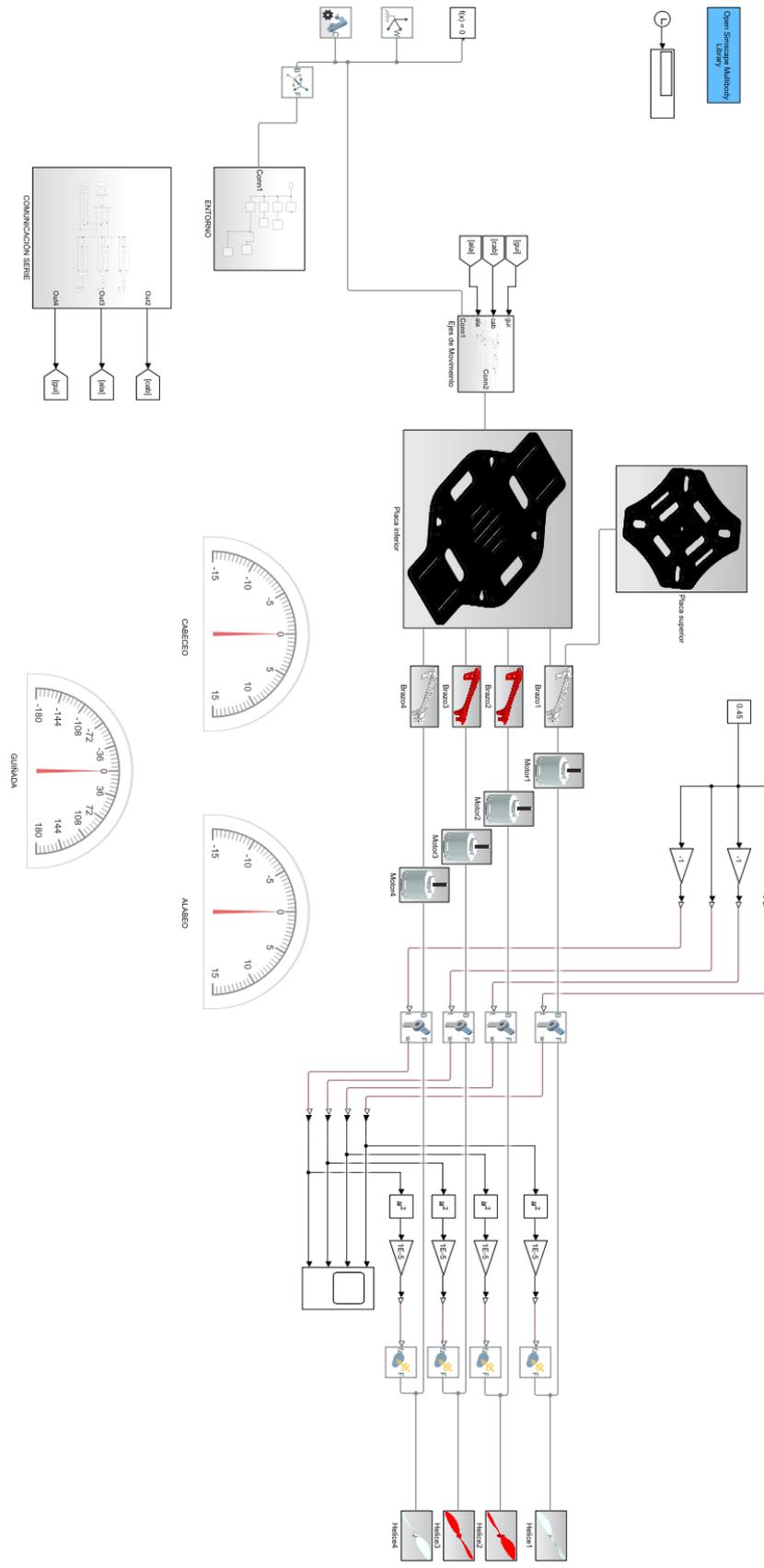
```

11.2 Vista general Simulinks

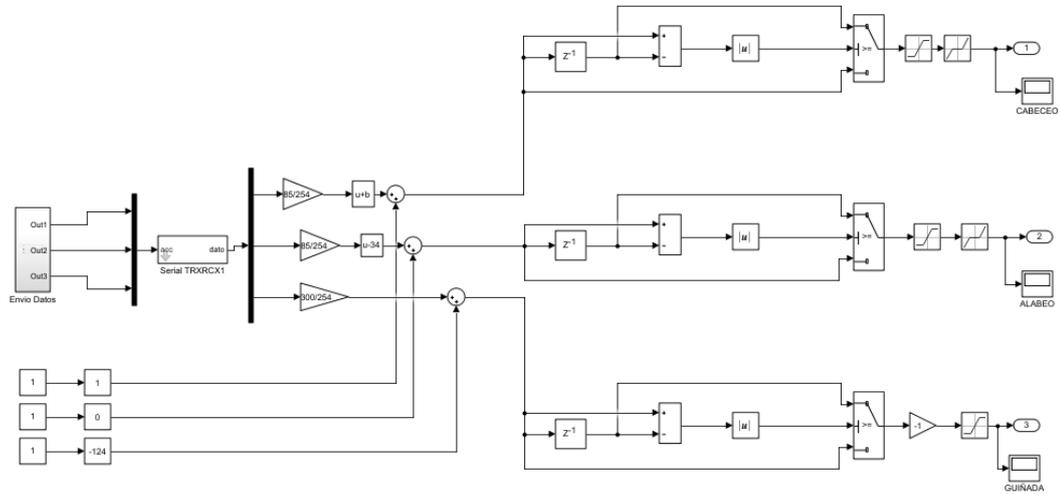
SIMULINK DE SIMULACIÓN DEL MECANISMO



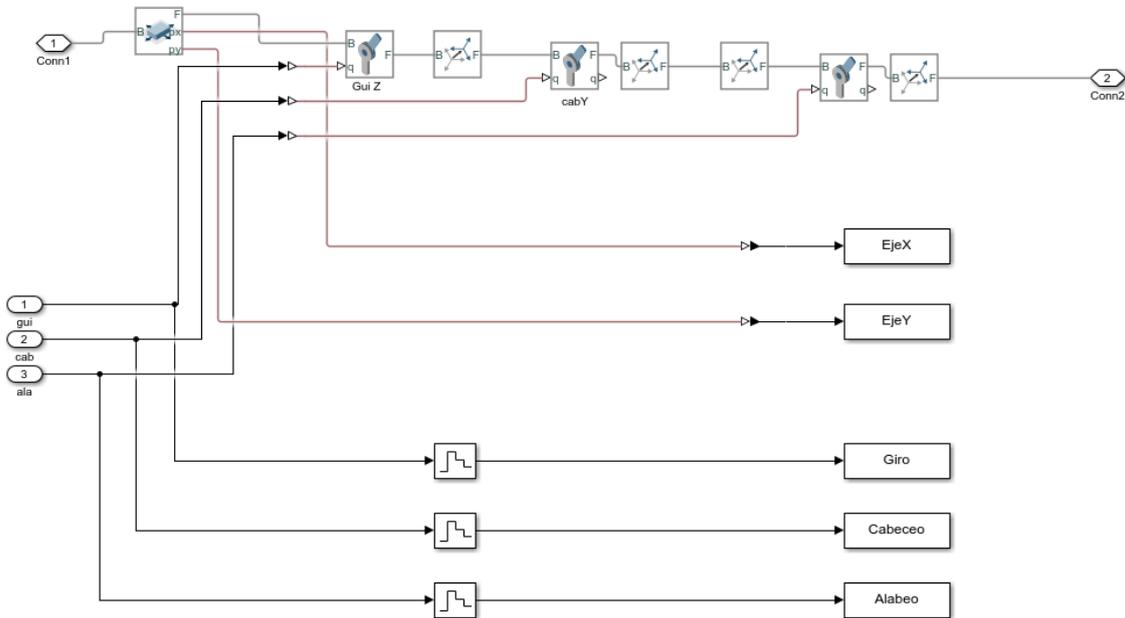
SIMULINK DE LA SIMULACIÓN DE VUELO



- Bloque comunicación Serie



- Articulaciones



11.3 Ficha técnica sensor 6120 series

Non-Contacting Hall Effect Single Turn Position Sensor



Model 6120 Series

Features:

- 7/8" diameter
- Non-contacting
- Hall-effect technology
- Single turn
- Multiple styles available
- Custom models available



Model Styles Available

6121	1/8" Shaft, 1/4" Bushing
6124	6 mm Shaft, 3/8" Bushing
6126	1/8" Shaft, 3/8" Bushing
6127	1/4" Shaft, 3/8" Bushing
612x-XXXX	Custom models are available; Contact Customer Service for special features

Electrical

Output Voltage	0.2 Vdc (4%) to 4.8 Vdc (96%) Typical (see Feature Codes table)
Output Overvoltage Limits	10 Vdc to -0.3 Vdc; output may be shorted to ground or supply without damage
Output Current	±8 mA Max.
Output Load	1 kΩ Min., 10 kΩ Typical
Operating Input Voltage Range	4.5 to 5.5 Vdc
Supply Voltage Absolute Limits	20 Vdc Max., -10 Vdc Min.
Independent Linearity ²	±0.5% (0.25% Available)
Hysteresis	0.2% Max.
Resolution	0.088° for 360° travel, 0.011° for 45° travel
Supply Current	8.5 mA Typical, 12 mA Max.
Dielectric Strength	750 V rms
Insulation Resistance	1,000 MegΩ Min.
Electrostatic Discharge (ESD)	Passes 2 kV human body model and 15 kV air discharge
Bulk Current Injection (BCI)	Passes 2-500 MHz at 200 mA
Actual Electrical Travel	360° Typical (see Ordering Information)
Temperature Coefficient of Output Voltage	± 20 ppm/°C

Mechanical

Total Mechanical Travel	360° Continuous (320° with stop feature)
Bearing	Bearing Bronze Bushing
Weight	0.6 oz. Typical
Static Stop Strength	40 in. oz.
Panel Nut Tightening Torque	25 in. lb. Max.
Rotational Speed	500 RPM Max.

¹ Specifications subject to change without notice.

² Linearity is measured between 1% and 99% of input voltage.



General Note
TT Electronics reserves the right to make changes in product specification without notice or liability. All information is subject to TT Electronics' own data and is considered accurate at time of going to print.

TT Electronics | BI Technologies
4200 Bonita Place, Fullerton, CA USA 92835 | Ph: +1 714 447 2343
www.ttelectronics.com | sensors@ttelectronics.com

© TT electronics plc

Issue A 11/2016 Page 1

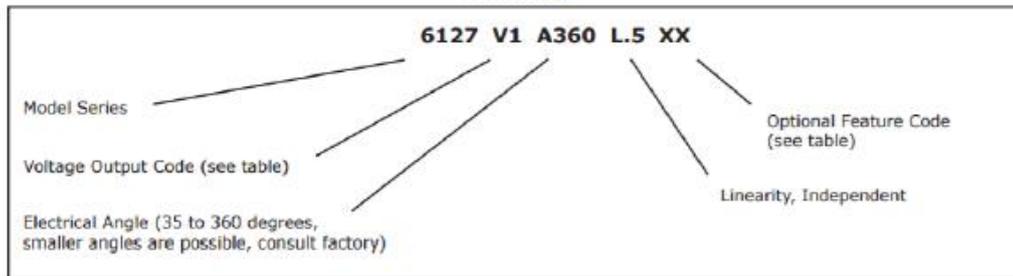
Non-Contacting Hall Effect Single Turn Position Sensor



Environmental

Operating Temperature Range	-40°C to +125°C
Shock	Per MIL R-39023, 6 ms Saw-tooth 100 G's
Vibration	Per MIL R-39023, 10 G's, 100 to 500 Hz
Moisture Resistance, Powered	Per MIL 202G, Method 106G
Rotational Life	10 million shaft revolutions with up to 8 oz sideload
Storage Temperature Range	-55°C to +125°C
Ingress Protection Rating (IP Code)	IP50, IP66 available as option (feature code ES)

Ordering



Feature Codes

Voltage Output Codes	
V0	≤ 3% to ≥ 96%
V1	4% to 96%
V2	5% to 95%
V3	10% to 90%
V4	15% to 85%
V5	20% to 80%

When V0 is used the angle specified is the theoretical angle over which the output would vary if the output could actually reach 0% and 100% of Vcc.

Optional Feature Codes	
ST	Stop (320°)
FS	Flatted Shaft (slot standard)
LT	Linearity Data
SL	Shaft Lock 6127 only
CW	Reverse Direction
ES	Seal (IP66) 6124 or 6127 only

When multiple Optional Feature codes are used the P/N shall be in the same sequence as listed in this table (top to bottom).

General Note
TT Electronics reserves the right to make changes in product specification without notice or liability. All information is subject to TT Electronics' own data and is considered accurate at time of going to print.

© TT electronics plc

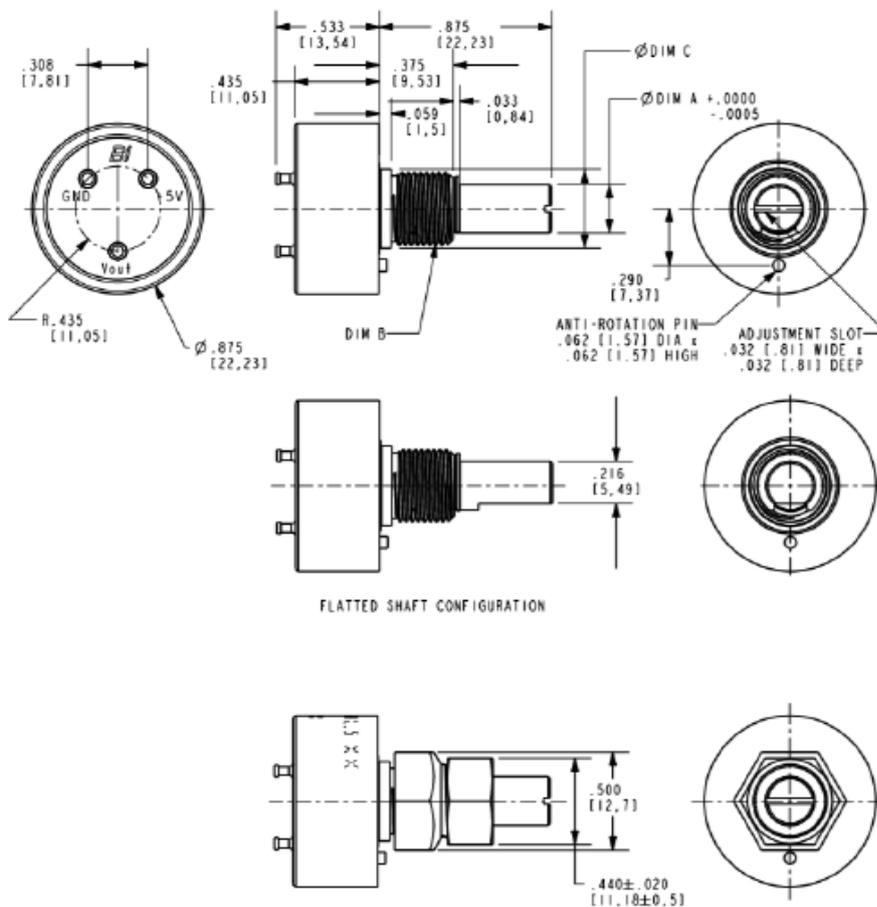
TT Electronics | BI Technologies
4200 Bonita Place, Fullerton, CA USA 92835 | Ph: +1 714 447 2345
www.ttelectronics.com | sensors@ttelectronics.com

Issue A 11/2016 Page 2

Non-Contacting Hall Effect Single Turn Position Sensor



Outline Drawings



DIM	6121	6124	6126	6127
A	.1248	6 mm	.1248	.2497
B	1/4-32 UNEF	3/8-32 UNEF	3/8-32 UNEF	3/8-32 UNEF
C	.281	.406	.406	.406

- NOTES:
1. UNIT SHIPS WITH NUT AND WASHER (NOT SHOWN).
 2. FOR SLOTTED OR FLATTED SHAFT, OUTPUT IS AT 50% IN POSITION SHOWN.
 3. DIMENSIONS: INCHES [mm].
 4. TOLERANCES: ±.015 [.38] UNLESS NOTED.

General Note
TT Electronics reserves the right to make changes in product specification without notice or liability. All information is subject to TT Electronics' own data and is considered accurate at time of going to print.

TT Electronics | BI Technologies
4200 Bonita Place, Fullerton, CA USA 92835 | Ph: +1 714 447 2345
www.ttelectronics.com | sensors@ttelectronics.com

DOCUMENTO II

PRESUPUESTO

En el presente documento se detallará el desglose del presupuesto sin amortizaciones del proyecto expuesto. El desglose de este se realizará en función de la naturaleza de los costes, y así facilitar su comprensión y aportar más información.

1. COSTES DEL HARDWARE

A continuación, se muestran los costes de los materiales que han sido empleados.

REFERENCIA	DESCRIPCIÓN	FABRICANTE	CANTIDAD (Uds)	PRECIO (€)	TOTAL (€)
M1	Ordenador		1	1.200,00 €	1.200,00 €
M2	Impresora 3D	BQ	1	1.148,00 €	1.148,00 €
M3	Arduino Due	Arduino	1	35,00 €	35,00 €
M4	Sensor efecto hall	TT Electronics	3	18,64 €	55,92 €
M5	Cuadróptero		1	50,00 €	50,00 €
M6	Rodamiento	Rollakyn	4	1,61 €	6,44 €
M7	Otros materiales		1	20,00 €	20,00 €
Coste total					2.515,36 €

2. COSTES DEL SOFTWARE

A continuación, se muestran los costes de las licencias del software usado.

REFERENCIA	DESCRIPCIÓN	FABRICANTE	CANTIDAD (Uds)	PRECIO (€)	TOTAL (€)
S1	SolidWorks	SolidWorks Corp	1	5.000,00 €	5.000,00 €
S2	Matlab	MathWorks	1	2.000,00 €	2.000,00 €
S3	Office Profesional 2016	Microsoft	1	100,00 €	100,00 €
S4	Microsoft Windows 10	Microsoft	1	100,00 €	100,00 €
Coste total					7.200,00 €

3. COSTE DE LOS RECURSOS HUMANOS

A continuación, se detallan los costes del sueldo que se debe percibir a acorde a las horas realizadas. El precio por hora se determina en base al *Convenio colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos*.

REFERENCIA	ACTIVIDAD	HORAS	PRECIO HOARA (€)	TOTAL (€)
R1	Concepción de la idea	10	20,00 €	200,00 €
R2	Diseño del mecanismo	15	20,00 €	300,00 €
R3	Simulación prototipo	20	20,00 €	400,00 €
R4	Implementación	80	20,00 €	1.600,00 €
R5	Test	5	20,00 €	100,00 €
Coste total (sin IVA)				2.600,00 €
Coste total (IVA 21% incluido)				3.146,00 €

4. COSTE TOTAL DEL PROYECTO

Tras ser visto por partes, el presupuesto total del proyecto es de la siguiente manera:

CONCEPTO	COSTE
Recursos Hardware	2.515,36 €
Recursos Software	7.200,00 €
Recursos humanos	3.146,00 €
COSTE TOTAL	12.861,36 €

DOCUMENTO III

PLIEGO DE CONDICIONES

1. DEFINICIÓN Y ALCANCE DEL PLIEGO

El objetivo del siguiente documento es establecer las condiciones bajo las que se rige el proyecto explicado en esta memoria que tiene por título: *Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero*.

Es aplicable al ámbito de todos los equipos utilizados durante la realización de este proyecto.

2. CONDICIONES GENERALES Y NORMATIVA

Teniendo en cuenta que se toma como autor del trabajo como contratista y a cualquier persona física o entidad que solicita su servicio como contratante, tras la firma del documento que reconozca el acuerdo entre ambas se considera que están conformes con el presente documento.

Se declara que el equipo ofertado incluye los elementos siguientes:

- Mecanismos de lectura de actitud del cuadricóptero
- Licencias con el software necesario

Todo material y equipos deben ser los descritos en la memoria o con características similares bajo estudio de viabilidad. Además, deberán cumplir con la norma vigente que les aplique.

La instalación del equipo debe ser llevada a cabo por personal cualificado. Asimismo, se incluye la puesta en marcha de este con las pruebas pertinentes y como constancia de ello se firmará un documento por ambas partes.

El cliente dispone de un año de garantía tras recibir el equipo. Dicha garantía cubre con asistencia gratuita y en caso de defecto la reparación del equipo o en última instancia su sustitución.

Fuera de la garantía, toda asistencia, reparación o sustitución del equipo queda por determinar por medio de oferta del contratante.

- Todos los elementos del proyecto, así como su uso cumple con la normativa expuesta en el Real **Decreto 1215/1997** sobre *las Disposiciones mínimas de seguridad y salud de equipos de trabajo usados por trabajadores*.

- Los equipos electrónicos deben cumplir con la normativa **IEC 61000 – UNE-EN 61000** concretamente con la **Parte 6-1: Normas genéricas que tiene como título Inmunidad en entornos residenciales, comerciales y de industria.**
- Se debe delimitar el espacio del mecanismo para evitar accidentes durante su uso, y así cumplir con la normativa de Seguridad de máquinas **ISO 13857:2008** por la que se rigen las *Distancias de seguridad para impedir que se alcancen zonas peligrosas con los miembros superiores e inferiores.*

3. CONDICIONES DE CARÁCTER ECONÓMICO

El precio indicado en este proyecto puede verse modificado teniendo en cuenta las peticiones del cliente que no se incorporen en el equipo bajo estudio de viabilidad.

Tras la firma del contrato, el pago se realizará en dos pagos:

- Se abona el 70% en un periodo de 3 días tras firmar el contrato.
- Se abona el 30% en un periodo de 10 días tras la puesta en marcha del equipo y que funciona correctamente.

Se establece penalizaciones del 10% del importe retrasado aplicable en cualquiera de los casos.

En caso de no recibir en la fecha acordada el equipo, el contratante tendrá derecho a la reducción del 5% del importe total del equipo.

El pago solo se podrá efectuar mediante transferencia bancaria.

4. ESPECIFICACIONES DE EJECUCIÓN

Una vez instalados el equipo, la puesta en marcha se llevará a cabo por personal cualificado perteneciente al contratista. Este incluye:

- Conexión eléctrico/electrónico
- La instalación del software
- Formación de la aplicación
- Prueba de vuelo con el cuadricóptero a estudiar

Se establecen las siguientes consideraciones para asegurar el correcto funcionamiento del equipo:

- El hardware del ordenador usado debe ser igual o superior al descrito.
- Correcto acople del cuadricóptero
- Dejar espacio suficiente para los movimientos del cuadricóptero y mecanismo para evitar golpes.
- El movimiento del cuadricóptero debe ser aquel que se considera como adecuado para su vuelo.

El contratista no se hace responsable de los posibles daños personales o en equipo si no se cumplen lo dispuesto en este documento.

DOCUMENTO IV

PLANOS

Todos los planos se adjuntan a continuación de esta página para respetar el formato y la escala de estos.

PLANO 1: Ensamblado explosionado del mecanismo

PLANO 2: Marco exterior

PLANO 3: Marco interior

PLANO 4: Plataforma

PLANO 5: Ejes

PLANO 6: Pilares

PLANO 7: Base

4

3

2

1

F

F

E

E

D

D

C

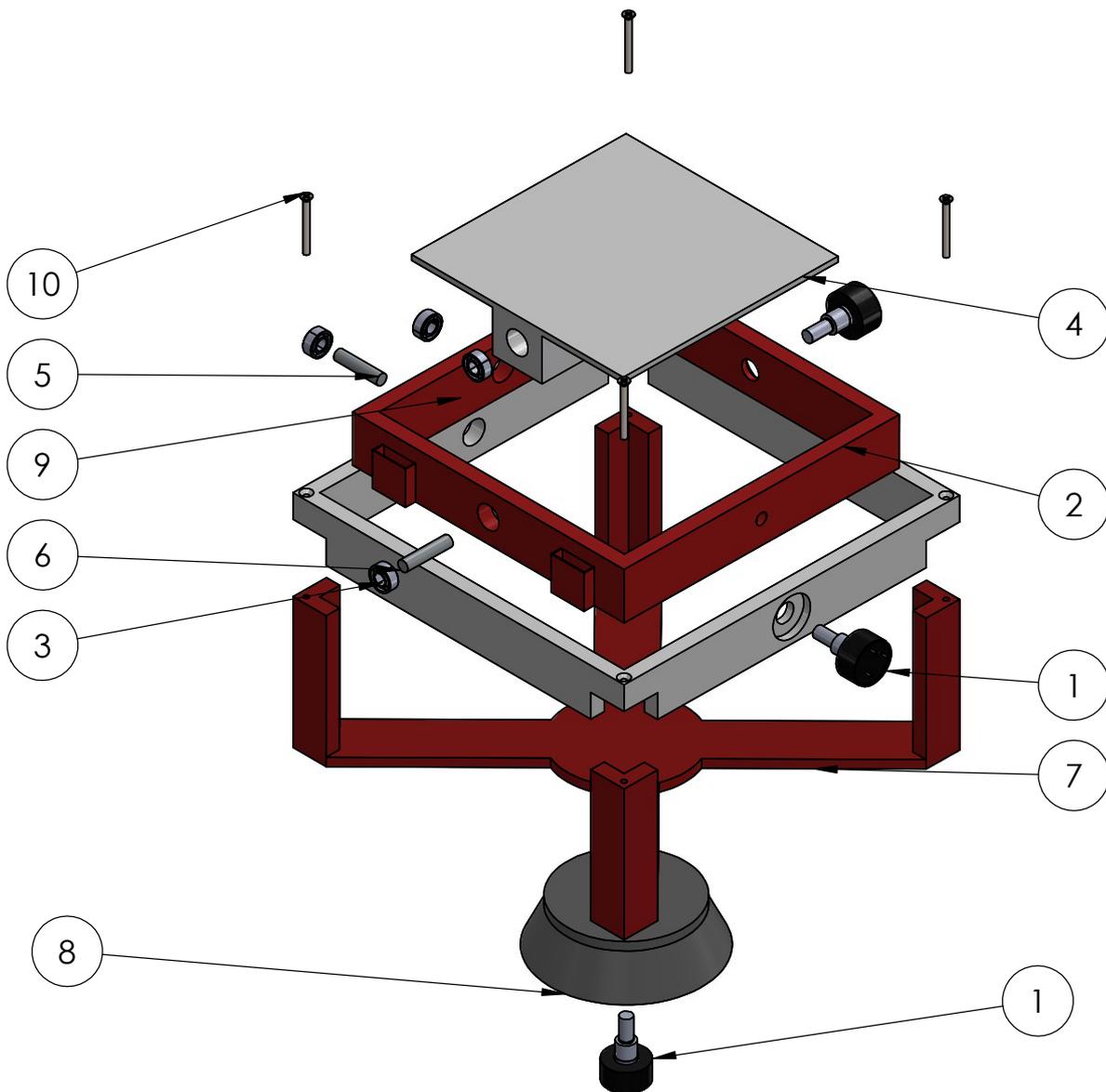
C

B

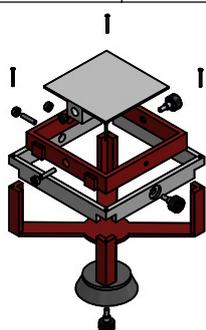
B

A

A



N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Sensor_BI6127		3
2	Marco_Interior		1
3	Rodamiento		4
4	Plataforma		1
5	Eje_Exterior		1
6	Eje_Interior		1
7	Pilares		1
8	Base		1
9	Marco_Exterior		1
10	ISO 7046-1 - M3 x 30 - Z - 30C		4




Escuela Técnica Superior de Ingeniería del Diseño

FECHA: 01/09/2021

NOMBRE:
Diego Aldair
Chango Figueroa

PESO:

NO CAMBIE LA ESCALA

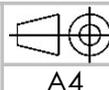
UNIDADES EN mm

PROYECTO:

Diseño e implementación de un
mecanismo de medición de la
actitud de un cuadricóptero

PLANO:

Ensamblado



A4

ESCALA 1:3

HOJA 1 DE 1

2

1

4

3

2

1

F

F

E

E

D

D

C

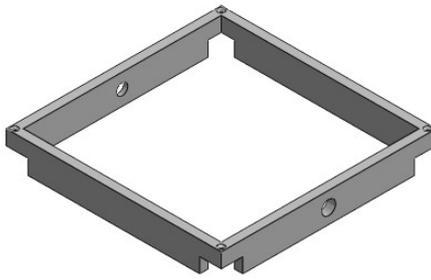
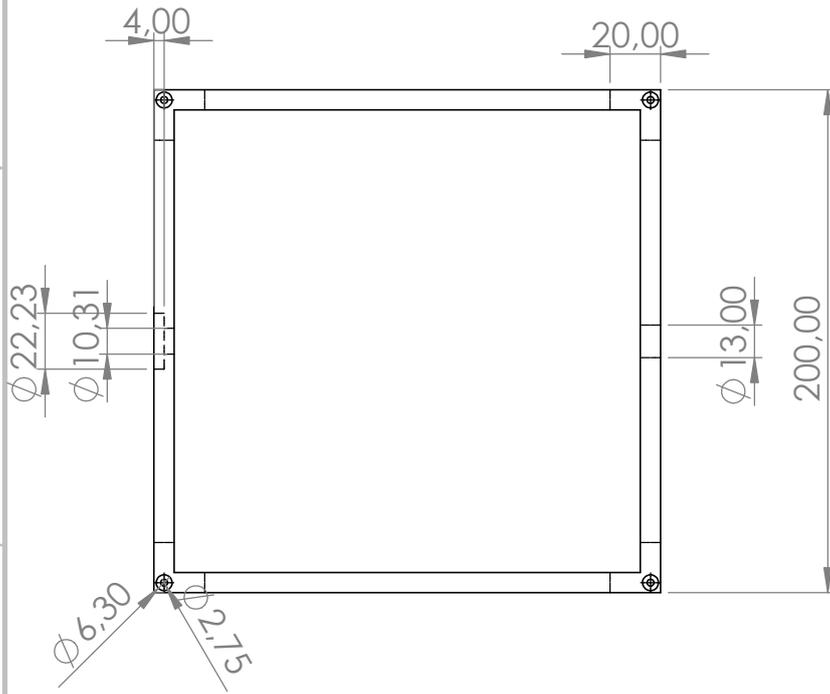
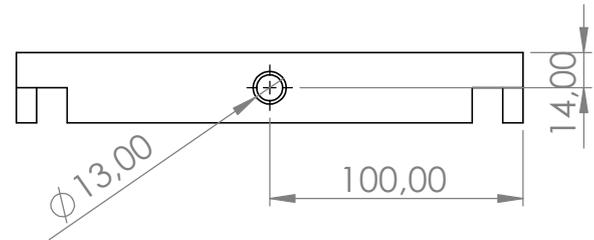
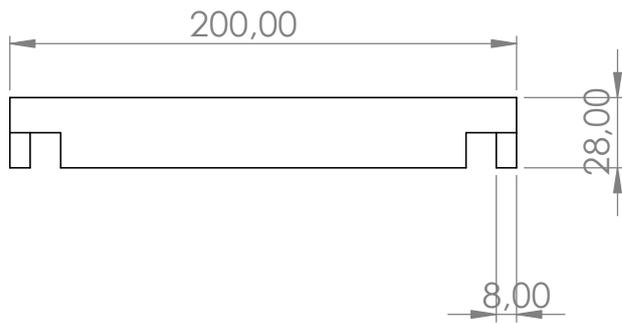
C

B

B

A

A



Escuela Técnica Superior de Ingeniería del Diseño

FECHA: 01/09/2021

NOMBRE: Diego Aldair Chango Figueroa

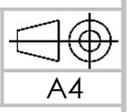
PESO:

NO CAMBIE LA ESCALA UNIDADES EN mm

PROYECTO: Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero

PLANO: **Marco exterior**

ESCALA 1:3 HOJA 1 DE 1



4

3

2

1

F

F

E

E

D

D

C

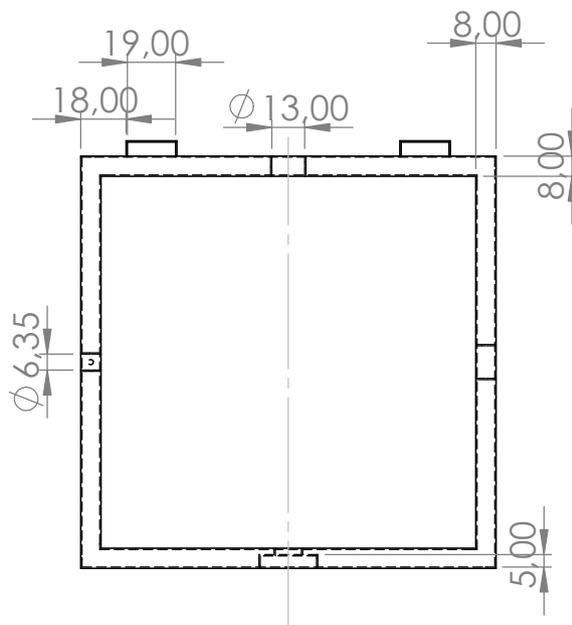
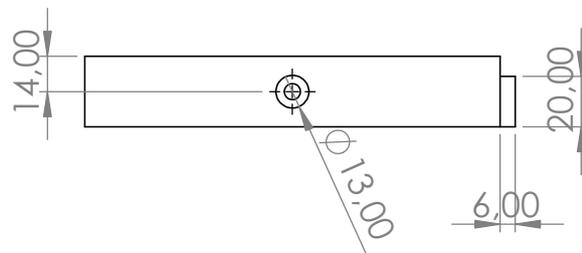
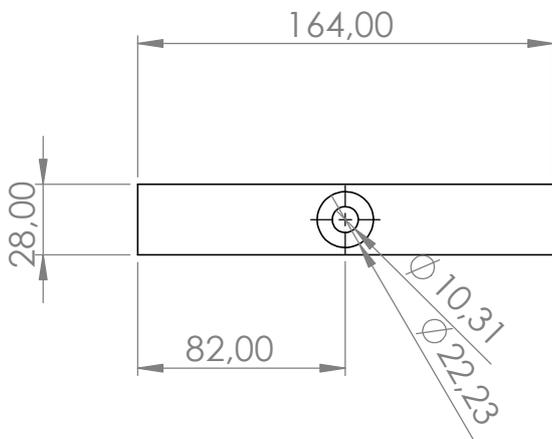
C

B

B

A

A



FECHA: 01/09/2021

NOMBRE:
Diego Aldair
Chango Figueroa

PESO:

NO CAMBIE LA ESCALA

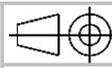
UNIDADES EN mm

PROYECTO:

Diseño e implementación de un
mecanismo de medición de la
actitud de un cuadricóptero

PLANO:

Marco interior



A4

ESCALA 1:3

HOJA 1 DE 1

2

1

4

3

2

1

F

F

E

E

D

D

C

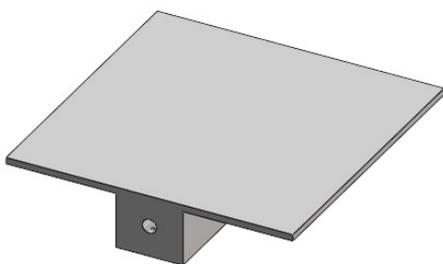
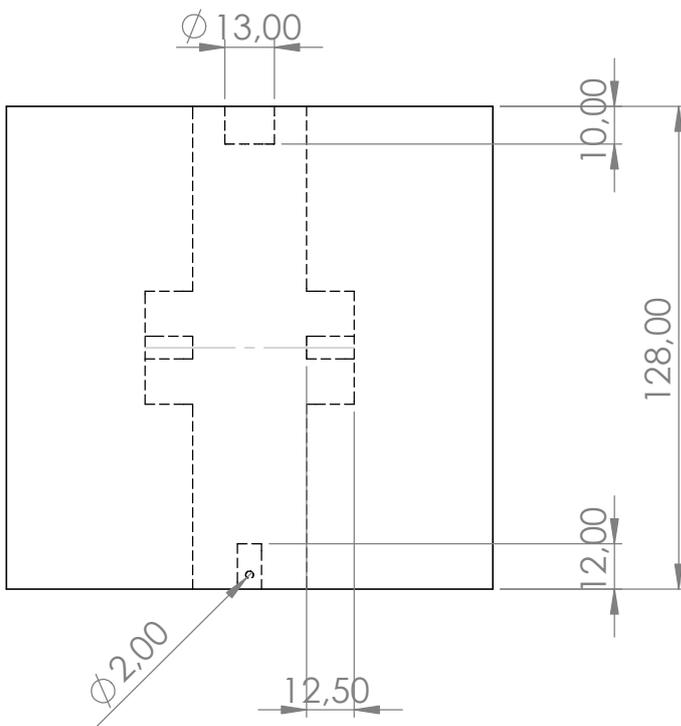
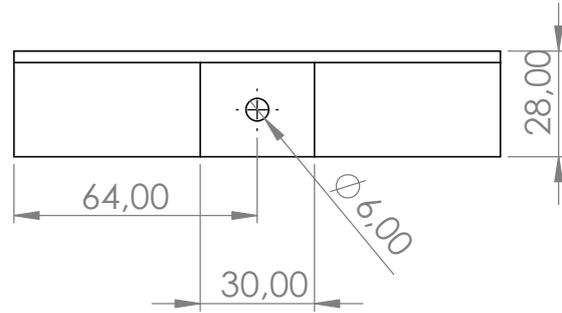
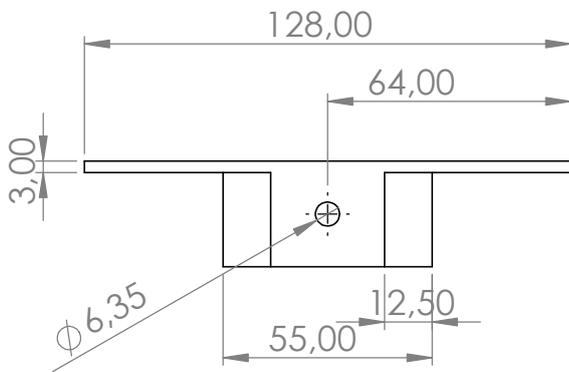
C

B

B

A

A



FECHA: 01/09/2021

NOMBRE: Diego Aldair Chango Figueroa

PESO:

NO CAMBIE LA ESCALA

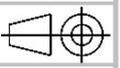
UNIDADES EN mm

PROYECTO:

Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero

PLANO:

Plataforma



A4

ESCALA 1:2

HOJA 1 DE 1

2

1

4

3

2

1

F

F

E

E

D

D

C

C

B

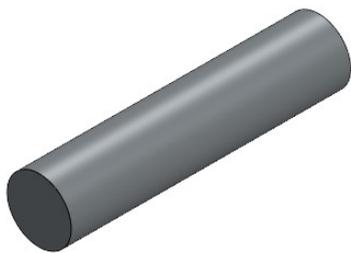
B

A

A

∅ 6,00

26,00



FECHA: 01/09/2021

NOMBRE:
Diego Aldair
Chango Figueroa

PESO:

NO CAMBIE LA ESCALA

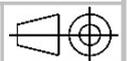
UNIDADES EN mm

PROYECTO:

Diseño e implementación de un
mecanismo de medición de la
actitud de un cuadricóptero

PLANO:

Ejes



A4

ESCALA 5:1

HOJA 1 DE 1

2

1

4

3

2

1

F

F

E

E

D

D

C

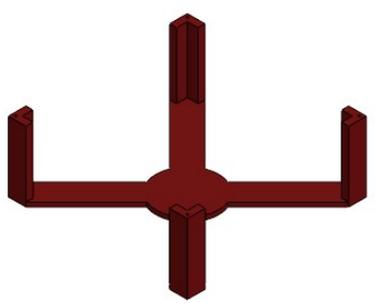
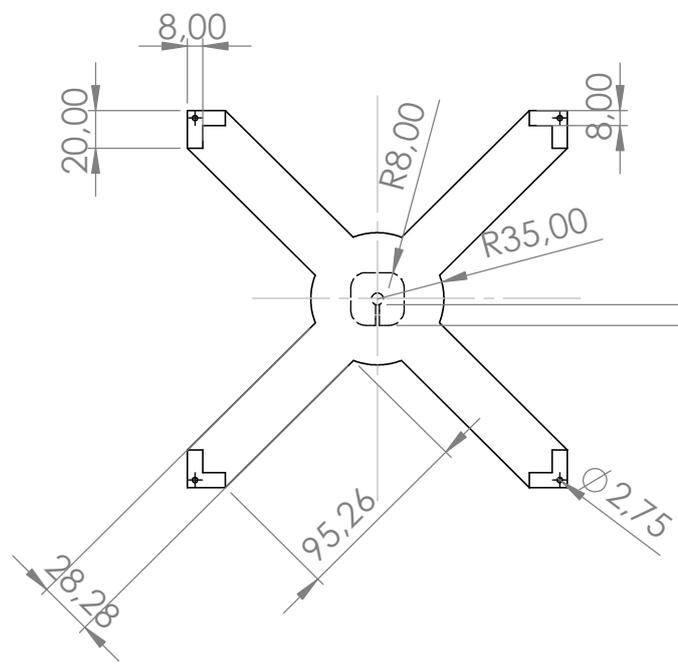
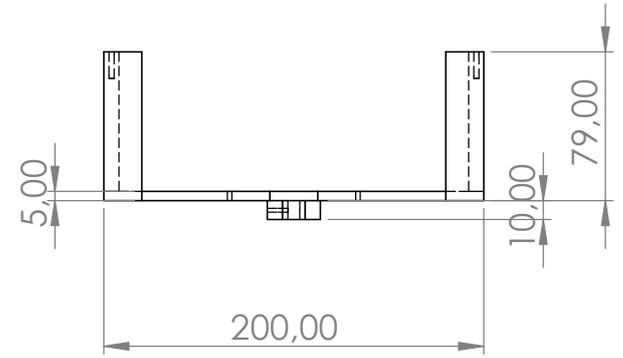
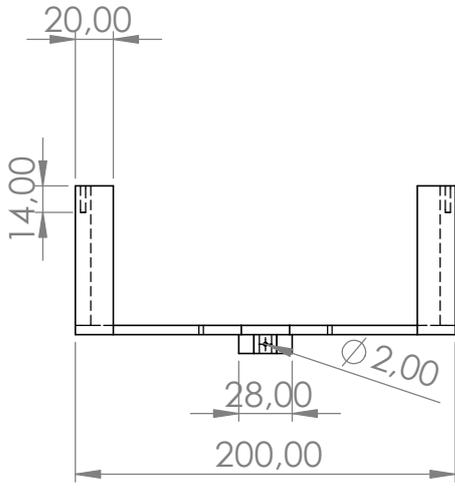
C

B

B

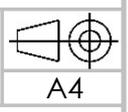
A

A



FECHA: 01/09/2021
 NOMBRE: Diego Aldair Chango Figueroa
 PESO:

NO CAMBIE LA ESCALA	UNIDADES EN mm
PROYECTO: Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero	
PLANO: Pilares	
ESCALA 1:4	HOJA 1 DE 1



2

1

4

3

2

1

F

F

E

E

D

D

C

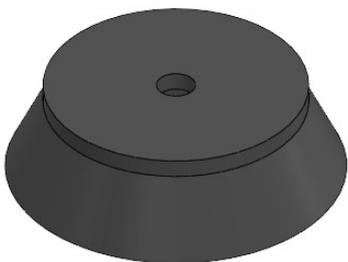
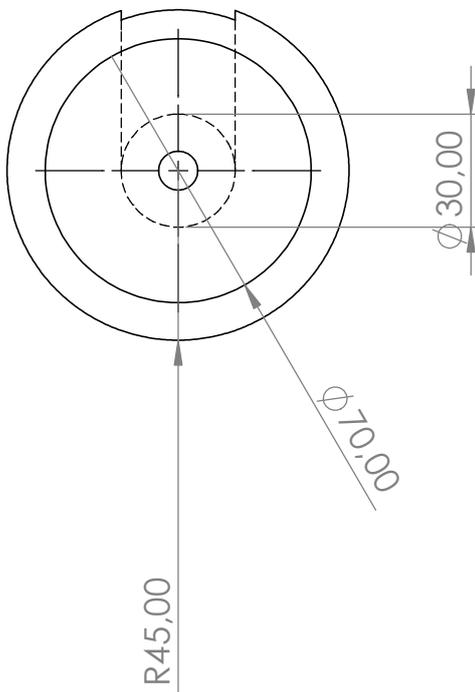
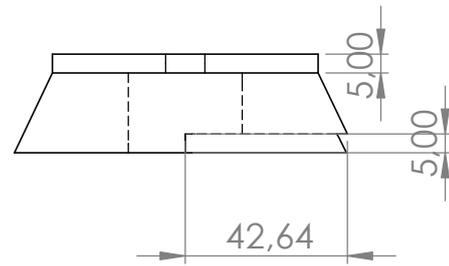
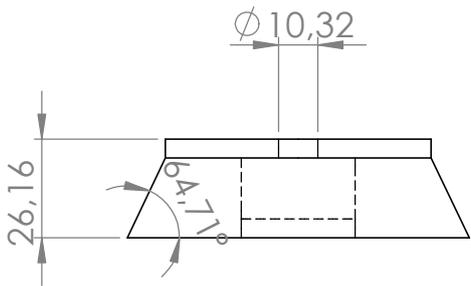
C

B

B

A

A



FECHA: 01/09/2021

NOMBRE: Diego Aldair Chango Figueroa

PESO:

NO CAMBIE LA ESCALA

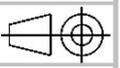
UNIDADES EN mm

PROYECTO:

Diseño e implementación de un mecanismo de medición de la actitud de un cuadricóptero

PLANO:

Base



A4

ESCALA 1:2

HOJA 1 DE 1

2

1