



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERIA ELECTROMECAÁNICA

CONTROL DE NAVEGACIÓN DE UN CUADRICÓPTERO MEDIANTE GESTOS

Autor: Javier del Río López-Corona

Director: Juan Luis Zamora Macho

Madrid

Julio 2018

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Javier del Río López-Corona

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Control de navegación de un cuadricóptero mediante gestos, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e

intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 07 de Julio de 2018

ACEPTA



Fdo:

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

**CONTROL DE NAVEGACIÓN DE UN CUADRICÓPTERO MEDIANTE
GESTOS**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2017/2018 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada

de otros documentos está debidamente referenciada.



Fdo.: Javier del Río López-Corona Fecha: 07/07/2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Juan Luis Zamora Macho Fecha: 07/07/2018



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERIA ELECTROMECAÁNICA

CONTROL DE NAVEGACIÓN DE UN CUADRICÓPTERO MEDIANTE GESTOS

RESUMEN

Autor: Javier del Río López-Corona

Director: Juan Luis Zamora Macho

Madrid

Julio 2018

Control de navegación de un cuadricóptero mediante gestos

Autor: Javier del Río López-Corona. Director: Juan Luis Zamora Macho.

I. INTRODUCCIÓN

Los UAV se definen como vehículos aéreos no tripulados. Los drones tienen su origen en operaciones militares en la segunda mitad del siglo XX. Su utilidad se basaba en misiones de reconocimiento y captación de información sobre el territorio enemigo. No fue hasta principios del siglo XXI cuando los UAV extendieron sus capacidades permitiendo su uso fuera del ámbito militar. Surge así, una diversificación y proliferación del uso de drones en las tareas civiles. En la actualidad, las aplicaciones de los UAV en el ámbito civil, se están extendiendo y cambiando los modelos tradicionales de trabajos en diversos sectores. Por este motivo, los UAV están al alcance de personas no especializadas o sin formación previa. Con esto en mente, en este proyecto se pretende crear una interfaz más intuitiva, sencilla y confortable dirigida al usuario medio.

Para ello se estudió el mercado actual en busca de sensores gestuales. Los dispositivos que mejor se ajustan para desarrollar este proyecto son Kinect, Leap Motion y Myo Armband. El primero hace una lectura del cuerpo entero, las diversas posturas y gestos. El segundo capta la información procedente de las manos: ángulos, posiciones, direcciones... El tercero, es un brazalete que se coloca sobre el antebrazo y hace un reconocimiento de los gestos a través de sensores electromiográficos (EMG) que detectan la actividad eléctrica en los músculos del brazo [1]. Este sensor, por ser inalámbrico, tiene grandes ventajas de movilidad frente a los otros dos. Al no disponer de él en la universidad, este dispositivo puede ser interesante para futuros desarrollos. Se concluyó que a pesar de un par de dificultades, es más sencillo y cómodo utilizar el Leap Motion para este proyecto que el Kinect en su implementación en Matlab y Simulink [2].

El objetivo de este es el control de navegación de un cuadricóptero mediante gestos. En la Figura 1 se muestran los distintos dispositivos y la conexión entre ellos.

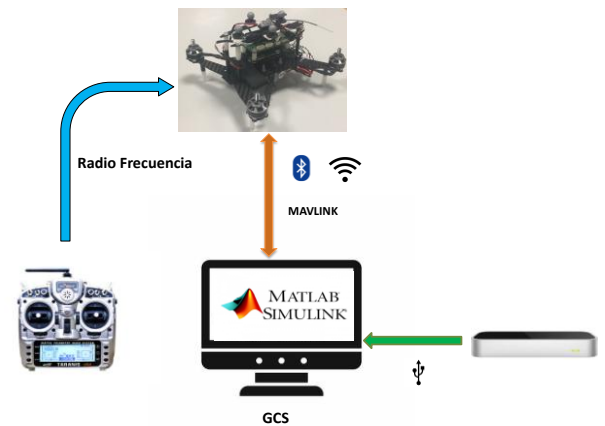


Figura 1: Interconexión dispositivos

II. LEAP MOTION

Es un sensor que a través de dos cámaras infrarrojas hace una lectura exclusivamente de las manos [3]. Sus lentes de gran ángulo que permiten una visión de 120° en el eje Z y 150° en el X y tiene un rango de visión de 80 cm.

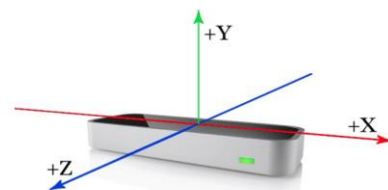


Figura 2: Ejes Leap Motion

El Leap Motion capta información procedente de las manos:

- Dedos: posición, ángulos y direcciones de cada dedo

- Palma de la mano: Posición y ángulos de cada mano
- Mano derecha: Binario. 1 si lee la mano, 0 sino la lee.
- Mano Izquierda: Binario. 1 si lee la mano, 0 sino la lee.
- Apertura: Variable continua entre 0 (cerrada) y 1 (abierta)

El control del cuadricóptero está formado por dos lazos: uno interno llamado control de estabilización y otro externo llamado control de navegación o autónomo. El control del UAV consiste en cuatro controles PID colocados en cascada: dos en el control de estabilización y dos en el control de navegación. El cuadricóptero tiene seis grados de libertad: tres de posición correspondientes a los tres ejes y tres de ángulos (guiñada, cabeceo y alabeo). La posición está incluida en el control de navegación y los ángulos en el control de estabilización. En el siguiente esquema (Figura 3) se muestra resumidamente la estructura del control. La emisora RC se encarga de introducirle referencia de ángulos en el control de estabilización. En este proyecto se intentó atacar con ángulos el control de estabilización pero el Leap Motion no tiene la misma resolución que la emisora. Finalmente se ha tenido que dar referencias de velocidades en el control de navegación. De esta forma se consigue un manejo del UAV más intuitivo que la emisora.

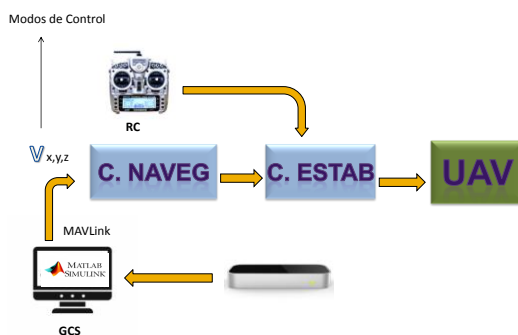


Figura 3: Control cuadricóptero

Con esto se abren infinitad de combinaciones posibles para elegir gestos para controlar el UAV. Para dar un mayor número de alternativas y comodidades se decidió crear tres formas de controlar al UAV en función de la experiencia del usuario. Estos tres modos tienen en común

tanto el armar y como desarmar motores. Para armarlos hace falta una lectura de ambas manos (primero la izquierda y luego la derecha) y para desarmarlo basta con cerrar el puño.

Las dos primeras están basadas en variables discretas y destinadas a usuarios amateurs: Control Básico por Ángulos y Control Básico por Posición. El primero hace una lectura de los ángulos de las manos: guiñada, cabeceo y alabeo (figura 4). El segundo capta únicamente información procedente de la palma de la mano en los tres ejes (x,y,z).

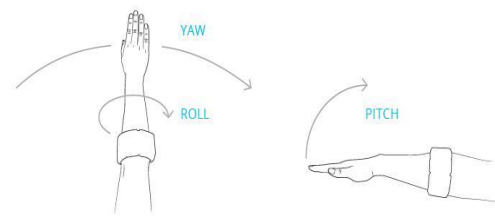


Figura 4: Ángulos de la mano

Tanto el Control Básico por Ángulos y Control Básico por Posición tienen en común el uso de bandas muertas. Si la mano, tanto en ángulo o en posición permanece en la banda muerta, el dron permanece en su sitio. En cambio, si la inclinación o posición de la mano se lleva fuera de esta zona, el UAV recibe la orden de moverse a velocidad constante en la dirección que se ha indicado con la mano. En la figura siguiente, el interior del cubo simboliza la banda muerta de la mano en el que el cuadricóptero no se mueve.

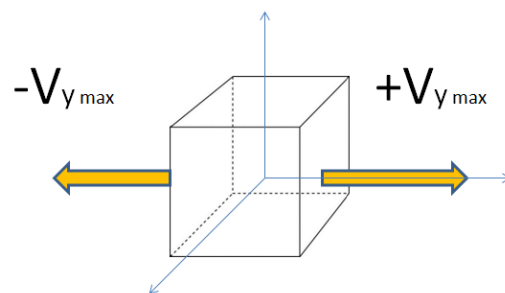


Figura 5: Modos Básicos

La tercera está compuesta por variables continuas y supone un modo de navegación más complejo que los anteriores: Control Avanzado por Posición. No tiene bandas muertas. La mano parte de un punto inicial en el que el dron permanece constante y a medida que alejamos las

manos de ese punto, el UAV va subiendo su velocidad en la dirección a donde apuntan las manos (Figura 6).

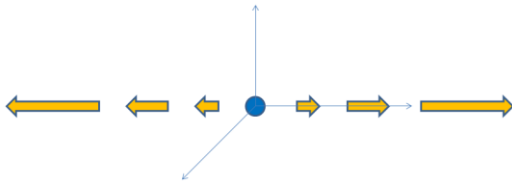


Figura 6: Modo Avanzado

III. ENSAYOS

En la siguiente gráfica se muestra la altura en metros del dron en modo Básico (Posición y Ángulos). La señal naranja corresponde a la referencia de la integral de velocidad que le aplicamos al UAV y la azul la salida que realmente hace.

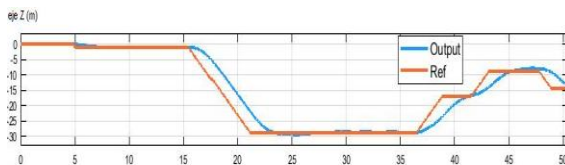


Figura 7: Altura UAV en modo Básico

Como se puede observar en la imagen, a partir del segundo 15 la mano derecha ha sido llevada fuera de la banda muerta y el cuadricóptero ha recibido la orden de moverse a una velocidad concreta. En el segundo 21 la mano ha vuelto a la banda muerta, por lo que el UAV permanece en su posición.

El siguiente ensayo corresponde a la altura en centímetros del dron en modo Avanzado por Posición.

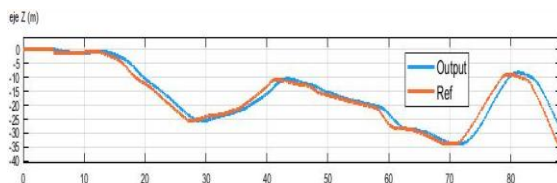


Figura 8: Altura UAV en modo Avanzado

En la imagen se puede ver que son valores continuos y no son rampas como en los dos modos anteriores. A cada posición concreta está vinculada una referencia de velocidad.

IV. SIMULACIÓN

Para conseguir depurar errores con mayor facilidad, antes de realizar la navegación sobre el cuadricóptero real, se han efectuado pruebas sobre el simulador de vuelo FlightGear [4]. Es un simulador de vuelo multiplataforma, libre y gratuito. La ventaja del FlightGear frente a otros simuladores de vuelo comerciales es su extensibilidad y flexibilidad. A continuación se muestra un esquema de las conexiones para realizar la simulación en el que un PC es la estación base y el otro ordenador emula el UAV:

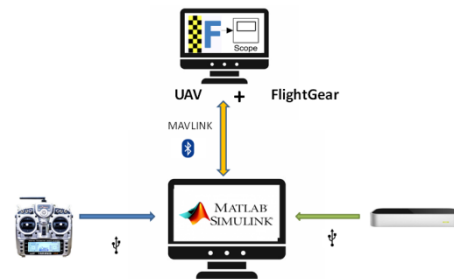


Figura 9: Conexión simulación

Los primeros ensayos antes han sido con la emisora. Luego se comenzó la sustitución por el Leap Motion. La estación base (GCS) recibe los datos por USB del Leap Motion y luego se realizó una comunicación UART inalámbrica (Bluetooth) a través del protocolo de comunicación MAVLINK [5], muy extendida en sector de la aeronáutica. Mediante este protocolo se envían los datos ya procesados del Leap Motion para poder visualizar FlightGear (Figura 10).

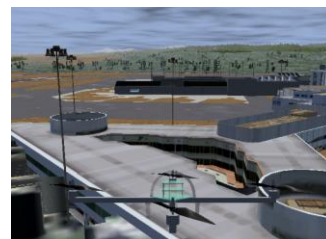


Figura 10: FlightGear

V. CUADRICÓPTERO

Una vez que se logró el control de navegación del UAV mediante el Leap Motion se realizó el

montaje de un cuadricóptero para poder hacer las pruebas y ensayos en condiciones reales. Hay que añadir que este proyecto no se centra en la construcción, sino en el control del cuadricóptero. El material empleado en la fabricación tiene que tener propiedades de ser ligero, fuerte y rígido. Este material empleado ha sido proporcionado por la universidad:

- Raspberry Pi Zero W: más pequeña y barata que las demás hermanas de Raspberry. El *Wireless* permite conexión Wifi a través de un Router para controlar el dron desde la estación base (GCS).
- PFX Mini: Incluye la unidad de medición inercial (IMU) y está compuesto por giróscopos y acelerómetros, así como de barómetros, magnetómetros y de sensores de temperatura y monitor de batería.
- Motores: se han testado diferentes motores para ver cual se ajusta mejor al peso del dron.
- Batería: LiPo de 1000 mAh con un voltaje nominal de 11.1V de tres celdas.
- Reflectores: para controlar su posición con cámaras.

VI. CONCLUSIONES

A la vista de los resultados, se puede afirmar que se ha conseguido con gran éxito el manejo del cuadricóptero de manera más intuitiva. Su implementación en Matlab/Simulink y la integración en el simulador de vuelo en 3D FlightGear avalan el proyecto. Se puede considerar un logro pero siempre pensando que es posible realizar mejoras y nuevas aportaciones.

En lo que se refiere al cuadricóptero se ha construido y configurado correctamente para su vuelo. Sin embargo, por falta de tiempo, no se ha llegado a realizar la navegación del cuadricóptero. Por este motivo el dron está habilitado para futuros desarrollos del laboratorio para el curso venidero.

Un posible estudio sería la investigación del sensor Myo Armband y ver si su resolución y las señales captadas son superiores o inferiores que el Leap Motion. El brazalete es compatible con el microcontrolador Raspberry Pi que ha sido utilizado para este proyecto.

BIBLIOGRAFÍA

- [1] Myo Armband. Obtenido de: www.myo.com
- [2] Menéndez botella, Marta. Proyecto fin de grado: Control de navegación mediante gestos. Escuela Técnica Superior de Ingeniería-Icai.
- [3] Leap motion app store | leap motion. Obtenido de: [Apps.leapmotion.com](https://apps.leapmotion.com/). Available at: <https://apps.leapmotion.com/>
- [4] The FlightGear Manual. Michael Basler. Getting Start (Enero 2017)
- [5] MAVLINK Common Message Set Specifications

Navigation Control of a Quadcopter Using Gestures

Author: Javier del Río López-Corona. Director: Juan Luis Zamora Macho.

I. INTRODUCCIÓN

UAVs are defined as unmanned aerial vehicles. Drones original purpose was merely for military operations at it dates back to the second half of the 20th century. They were originally used for recon missions over the enemy territory, thus only risking material damage and no humans involved. At the beginning of the 21th century UAVs extended their capabilities beyond military usage and found a way in our everyday lives. There is a diversification of drones in civil tasks. Currently, the applications of the UAV in the civil field are changing traditional models of work in different sectors like, for example, package deliveries. For this reason, UAVs are available for people without any specialization. So this project aims to create a more intuitive, simple and comfortable interface to provide a better experience for an average user.

The current market was studied and the absence of gesture recognition sensors was found. The devices that are currently fit to develop this project are Kinect, Leap Motion and Myo Armband. The first one makes a reading of the whole body with different gestures. The second one captures the information from the hands: angles, positions, directions... The last one is a bracelet that is placed on the forearm and recognizes the gestures through electromyography sensors (EMG) that detect the electrical activity of the muscles [1]. This sensor is wireless, so it has great mobility advantages. This device is not available at the university, but it will be interesting for future developments. In spite of a couple of difficulties, it is easier and more comfortable to use Leap Motion than Kinect in its implementation in Matlab and Simulink [2].

The objective of this project is the navigation control of a quadcopter using gestures. In the

first figure, it is shown the connection between the different devices.

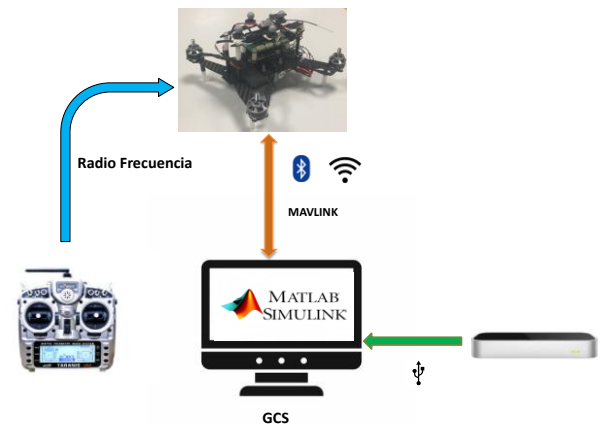


Figure 1: Diagram

II. LEAP MOTION

This sensor uses two infrared cameras to read exclusively the hands [3]. Its wide angle lenses that allow a 120° vision on the Z axis and 150° on the X and have a vision range of 80 cm.

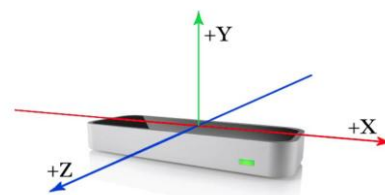


Figure 2: Leap Motion axes

Leap Motion detects information from the hands:

- Fingers: position, angles, directions...
- Palm: Positions and angles.
- Right hand: binary value. 1 detected. 0 not detected

- Left hand: binary value. 1 detected. 0 not detected
- Is Closed: if the hand is open or closed. Not binary value.

The control consists of two loops: an internal one called, stabilization, and an external one, called navigation or autonomous. The control of the UAV has four PID controls placed in cascade: two in the stabilization control and two in the navigation control. The quadcopter has six degrees of freedom: three positions corresponding to the three axes and three angles (yaw, pitch and roll). The position is included in the navigation control and the angles in the stabilization control. In the following diagram (Figure 3), the structure of the control is summarized. The RC transmitter is responsible for making reference of angles in the stabilization control. Leap Motion does not have the same resolution as the RC transmitter. So it is necessary to give references of speeds in the navigation instead of stabilization. In this way, UAV management is more intuitive.

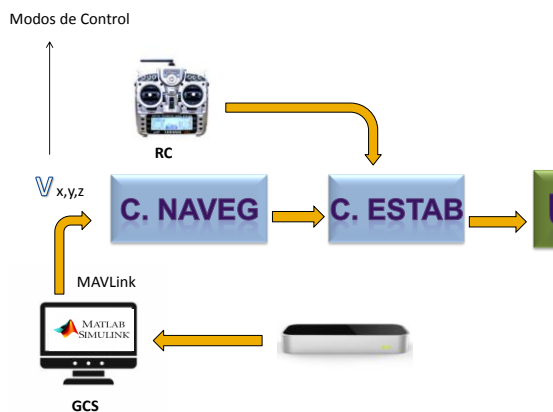


Figure 3: Control

A great number of combinations could be chosen to control the UAV with gestures. Three ways to control UAVs have been created based on the user's experience. These three modes have in common the arm and disarm electric engines.

The first two are based on discrete variables: Basic Control by Angles and Basic Control by

Position. The first one makes a reading of the angles of the hands: yaw, pitch and roll. The second one captures only information from the palm of the hand on the three axes (x, y, z).

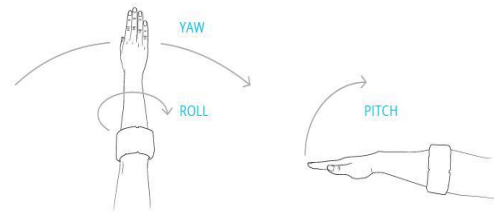


Figure 4: Angles

Both controls have in common the use of dead bands. If the right hand is in the dead band, the drone remains in place. On the other hand, if the inclination or position of the hand is carried out of this zone, the UAV receives the command to move at a constant speed in the direction indicated by the hand. In the following figure, the inside of the cube symbolizes the dead band of the hand in which the quadcopter do not move.

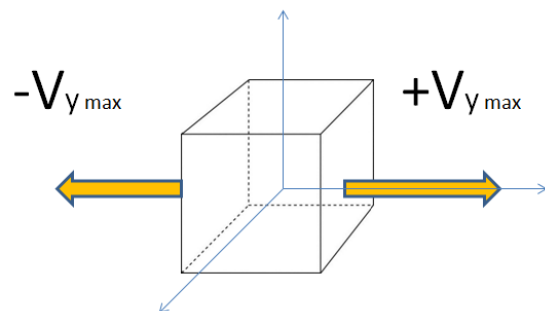


Figure 5: Basic Control

The third one is composed of continuous variables and it has a more complex navigation mode than the previous ones: Advanced Position Control. It does not have dead bands. The hand starts from an initial point where the drone remains constant and as we move our hands away from that point, the UAV goes up its speed in the direction where the hands are pointing (Figure 6).

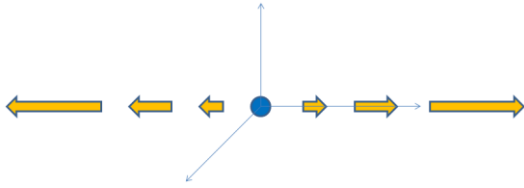


Figure 6: Advanced Control

III. TESTS

The following graphic shows the height in meters of the drone in Basic Mode (Position and Angles). The orange signal corresponds to the reference of the speed integral and the blue one the output that really does.

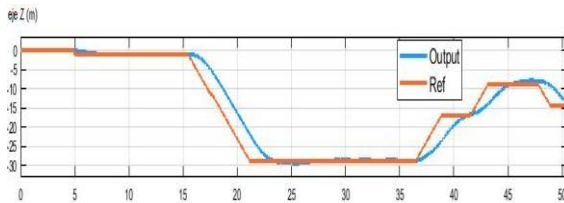


Figure 7: UAV height in Basic mode

As it has been seen in the image, from the second 15, the right hand has been taken out of the dead band and the quadcopter has been ordered to move at a specific speed. In the second 21, the hand has returned to the dead band, so the UAV remains in its position.

The following test corresponds to the height in centimeters of the drone in Advanced by Position mode.

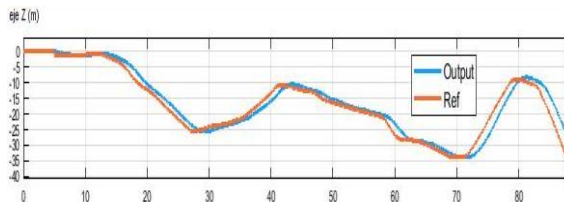


Figure 8: UAV height in Advanced mode

There are continuous values in the graphic and they are not ramps like in the two previous modes. A speed reference is linked to each specific position.

IV. SIMULATION

In order to make debugging easier, before carrying out the navigation on real conditions with the quadcopter, tests have been done on the flight simulator FlightGear [4]. It is a free multiplatform flight simulator. The advantage of FlightGear, unlike other commercial flight simulators, is its extensibility and flexibility. Below there is a diagram of the connections to perform the simulation in which one PC is the GCS and the other one emulates the UAV:

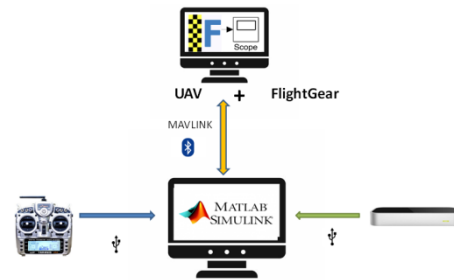


Figure 9: Simulation

The first couple of tests were done with the radio transmitter. Then Leap Motion replaces it. GCS receives the data via USB from Leap Motion and then a UART communication (Bluetooth) was made. The communication protocol used with GCS was MAVLINK [5], specialized protocol for aerial vehicles. Through this protocol, the processed data of Leap Motion is sent and able to be visualized in FlightGear (Figure 10).

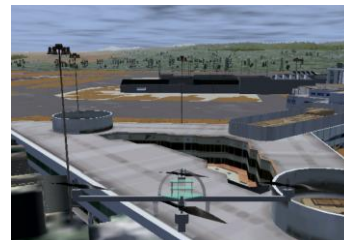


Figure 10: FlightGear

V. QUADCOPTER

In order to make tests in real conditions, a quadcopter was built. For this project it is more important the design of the control instead of building a UAV. The material used in manufacturing has properties of being light, strong and rigid. This material has been provided by the university:

- Raspberry Pi Zero W: smaller and cheaper than other Raspberry. The Wireless allows Wifi connection through a Router to control the drone from the Ground Control Station (GCS).
- PFX Mini: includes the inertial measurement unit (IMU) and it is composed of gyroscopes and accelerometers, as well as barometers, temperature and battery sensors.
- Electric Engines: different engines have been tested to see which one it is better with drone's weight.
- Battery: LiPo of 1000 mAh with a nominal voltage of 11.1V with 3 cells.
- Reflectors: to control position with cameras.

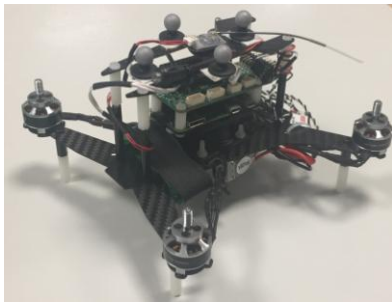


Figura 11: Quadcopter

VI. CONCLUSION

Once the project was concluded, it was found that the management of the quadcopter has been achieved in a more intuitive way with great

success. Its implementation in Matlab/Simulink and the integration in the flight simulator in 3D FlightGear guarantee this project. It can be considered an achievement but always thinking that it is possible to make improvements and new contributions.

On the other hand, UAV has been correctly constructed and configured for your flight. However, due to lack of time, UAV's navigation has not been carried out. For this reason the drone is enabled for future laboratory developments for the upcoming course.

A possible future study could be the investigation of the Myo Armband sensor and study if its resolution and the signals captured are higher or lower than Leap Motion. The bracelet is compatible with Raspberry Pi microcontroller that has been used for this project.

BIBLIOGRAPHY

- [1] Myo Armband. At: www.myo.com
- [2] Menéndez botella, Marta. Proyecto fin de grado: Control de navegación mediante gestos. Escuela Técnica Superior de Ingeniería-Icai.
- [3] Leap motion app store | leap motion. Obtenido de: [Apps.leapmotion.com](https://apps.leapmotion.com/). Available at: <https://apps.leapmotion.com/>
- [4] The FlightGear Manual. Michael Basler. Getting Start (Enero 2017)
- [5] MAVLINK Common Message Set Specifications



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERIA ELECTROMECAÁNICA

CONTROL DE NAVEGACIÓN DE UN CUADRICÓPTERO MEDIANTE GESTOS

MEMORIA

Autor: Javier del Río López-Corona

Director: Juan Luis Zamora Macho

Madrid

Julio 2018

AGRADECIMIENTOS

*A Juan Luis por su entera
disposición, ayuda, experiencia y
especialmente por la gran
paciencia que tiene por
aguantarme.*

*A Javi, por enseñarme el camino
siempre que estaba perdido y
siempre de forma desinteresada.*

*A mi familia por su preocupación
y apoyo incondicional.*



Índice de la memoria

Parte I	Memoria.....	5
Capítulo 1	Introducción	7
1.1	Motivación del Proyecto	8
1.2	Objetivos	9
1.3	Metodología / Solución Desarrollada.....	10
1.4	Recursos / herramientas empleadas.....	11
1.5	Estructura de la Memoria	12
Capítulo 2	Estado del Arte.....	13
2.1	Introducción.....	13
2.2	Kinect.....	15
2.3	Leap Motion.....	19
2.4	Myo Gesture Control Armband.....	22
Capítulo 3	Modelado y Control del Cuadricóptero	27
3.1	Material Utilizado para la Fabricación	27
3.1.1	Motores	28
3.1.2	Batería	31
3.1.3	Raspberry Pi Zero w	32
3.1.4	PFX Mini	33
3.1.5	ESC's (Electronic Speed Control).....	34
3.2	Comunicaciones.....	34
3.2.1	UART.....	35
3.2.2	MAVlink.....	37
3.3	Emisora RC.....	38
3.4	Control del Cuadricóptero	40
3.5	Estación Base	42



Capítulo 4	<i>Leap Motion</i>	45
4.1	Hardware	45
4.2	Software-Matleap	47
4.3	Lectura del Dispositivo	48
4.4	Comprobaciones y Pruebas	51
4.5	Procesamiento de las Señales.....	53
4.5.1	Control Básico por Posición.....	55
4.5.2	Control Básico por Ángulos.....	57
4.5.3	Control Avanzado por Posición	58
Capítulo 5	<i>Simulación y Visualización</i>	61
5.1	Visualización en Simulink.....	61
5.1.1	Control Básico por Posición.....	63
5.1.2	Control Básico por Ángulos.....	63
5.1.3	Control Avanzado por Posición	64
5.2	FlightGear	65
Capítulo 6	<i>Resumen y Conclusiones</i>	69
6.1	Resumen Tareas Realizadas	69
6.2	Conclusiones.....	70
Capítulo 7	<i>Futuros desarrollos</i>	71
Bibliografía	73	
Parte II	<i>Presupuestos</i>	75
Capítulo 1	<i>Recursos Empleados</i>	77
1.1	Componentes Principales.....	77
1.2	Herramientas y Software	78
1.3	Mano de Obra.....	78
Capítulo 2	<i>Costes Unitarios</i>	79
2.1	Componentes Principales.....	79



2.2	Herramientas y Software	80
2.3	Mano de Obra	80
<i>Capítulo 3</i>	<i>Sumas Parciales.....</i>	<i>81</i>
3.1	Componentes Principales.....	81
3.2	Herramientas y Software	82
3.3	Mano de Obra	82
<i>Capítulo 4</i>	<i>Presupuesto General</i>	<i>83</i>
<i>Parte III</i>	<i>Código fuente.....</i>	<i>85</i>
	<i>Código fuente 1.....</i>	<i>87</i>



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

ÍNDICE DE LA MEMORIA



Índice de figuras

Figura 1: Objetivo del proyecto.....	10
Figura 2: Ejes del avión.....	14
Figura 3: Kinect v1	16
Figura 4: Adaptador Kinect.....	17
Figura 5: Leap Motion.....	20
Figura 6: Evolución Leap Motion	21
Figura 7: VR Leap Motion	22
Figura 8: Brazaletes Myo Amrband.....	23
Figura 9: Myo con Oculus Rift.....	24
Figura 10: Motores Dragon Fly MC1806 2300kv.....	28
Figura 11: Banco de potencia.....	29
Figura 12: Empuje motores	30
Figura 13: Batería Lipo 11.1 voltios	31
Figura 14: Imax Lipo Balance Charger	32
Figura 15: Raspberry Pi Zero W	33
Figura 16: PFX Mini	33
Figura 17: SC 20A.....	34
Figura 18: HC 05	35
Figura 19: Conexión UART	36
Figura 20: Protocolo comunicación Uart	36
Figura 21: Serial Configuration UART.....	37
Figura 22: Tanaris Plus X9D.....	39
Figura 23: Bloque Joystick Input Simulink.....	39



Figura 24: Estructura del control.....	41
Figura 25: UAV Control System en Simulink	42
Figura 26: Interconexión de dispositivos	43
Figura 27: PC Control Station.....	43
Figura 28: Hardware Leap Motion.....	46
Figura 29: Interaction Box	47
Figura 30: Ejes del Leap Motion.....	48
Figura 31: Bloque Leap Motion en Simulink.....	49
Figura 32: Ángulos de la mano	50
Figura 33: Posición manos eje X.....	51
Figura 34: Ángulos de la mano	52
Figura 35: Altura mano	53
Figura 36: Leap Motion Update en Simulink.....	54
Figura 37: Yaw	55
Figura 38: Cubo en el sistema de coordenadas	56
Figura 39: Pitch	57
Figura 40: Roll.....	58
Figura 41: Control Avanzado	59
Figura 42: Esquema simulación dron.....	61
Figura 43: Ejes de coordenadas aviación	62
Figura 44: Posicionamiento UAV en Control Básico por Posición	63
Figura 45: Posicionamiento UAV en Control Básico por Ángulos	64
Figura 46: Posicionamiento UAV en Control Avanzado por Posición.....	65
Figura 47: Arducopter en FlightGear	66
Figura 48: Bloque FlightGear en Simulink.....	67
Figura 49: Esquema conexión FlightGear.....	68



Índice de tablas

Tabla 1: Comparativa Kinect v1/v2	18
Tabla 2: Potencia motores	30
Tabla 3: Recursos Empleados de Componentes Principales.....	77
Tabla 4: Recursos Empleados de Herramientas y Software.....	78
Tabla 5: Recursos Empleados de Mano de obra	78
Tabla 6: Costes Unitarios de Componentes Principales.....	79
Tabla 7: Costes Unitarios de Herramientas y Software	80
Tabla 8: Costes Unitarios Mano de obra	80
Tabla 9: Sumas Parciales de Componentes Principales	81
Tabla 10: Sumas Parciales de Herramientas y Software.....	82
Tabla 11: Sumas Parciales de Mano de obra.....	82
Tabla 12: Presupuesto General del proyecto.....	83



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Parte I MEMORIA



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 INTRODUCCIÓN

Los UAV (Unmanned Aerial Vehicle) por sus siglas en inglés, o como se les llama hoy en día , Drones, son vehículos aéreos sin tripulación, capaces de mantener un control de vuelo controlado y sostenido, y propulsado por un motor de explosión, eléctrico o de reacción. El término vehículo aéreo no tripulado (UAV) se hizo común en los años 90 para describir las aeronaves robóticas y se reemplazó por el término vehículo aéreo pilotado remotamente (RPV), muy utilizado durante la guerra de Vietnam y sus años posteriores [1].

Los UAV tienen su origen en operaciones militares de la Guerra Fría. Los vehículos aéreos no tripulados, equipados con cámaras, eran utilizados para misiones de reconocimiento sobre el territorio enemigo. La investigación en los UAV se basaba en mejorar sus condiciones para no ser visto por el enemigo y así dotarlo de invisibilidad. Se hicieron más rápidos y pequeños, con capacidad de volar a gran altitud sin ser visto. Estos sistemas se fueron haciendo más sofisticados tanto en los requisitos de la misión como en la seguridad de las comunicaciones. El objetivo principal de los drones se convirtió en la captación de información.

A pesar de su exitosa capacidad de vigilancia, los ingenieros tenían problemas para alcanzar niveles aceptables de fiabilidad. Hubo que esperar a la década de 1970 para que los UAV contaran con la tecnología más avanzada y revolucionaria, la capacidad de utilización GPS para la navegación autónoma.

No es hasta a partir del siglo XXI cuando los UAV han ido adquiriendo nuevas capacidades que permiten su uso fuera del ámbito militar. Surge así, una diversificación en las operaciones civiles. En la actualidad, las aplicaciones de los UAV en el ámbito civil, se están extendiendo y cambiando los modelos tradicionales de trabajos en sectores como la agricultura, la topografía, la inspección y mantenimiento de infraestructura industrial, así como en tareas del sector audiovisual y otras muy diversas. Es por lo tanto el ámbito civil, el que a



partir de ahora marcará la total expansión e internacionalización de esta tecnología.

1.1 MOTIVACIÓN DEL PROYECTO

Durante los últimos años se ha observado una proliferación del uso de drones para tareas civiles. Se está incrementando su empleo en áreas de la vida cotidiana. Algunos ejemplos son el reparto de paquetes, llevado a cabo por parte de Amazon; lucha contra el fuego; gestión de explotaciones agrícolas o de líneas de alta tensión. También ha evolucionado el mundo de la fotografía y vídeos aéreos debido a su capacidad de captar ángulos de visión que eran imposibles de alcanzar anteriormente. Se está implementando su uso por ser capaces de realizar tareas que son difíciles para el ser humano como el de alcanzar lugares de difícil acceso. Incluso se han llegado a probar aerotaxis autónomos no tripulados [3].

El desarrollo tecnológico tiende a facilitar la actividad humana y de ofrecer un mayor número de alternativas. Un claro ejemplo son los cambios que se vivieron en el ámbito de los videojuegos con la innovación del sensor Kinect en la Xbox, que con su lectura del cuerpo humano permitía su uso en sustitución de un mando a distancia. Sensores como Kinect ofrecen alternativas diferentes y más reales que los que ofrecen los mandos a distancia o emisoras [4]. De este modo, esta sustitución de las emisoras es posible alcanzarse en el control de navegación de los UAV. Es por tanto que la revolución de la realidad virtual está llegando a los vehículos aéreos.

Los drones ya están al alcance de personas sin formación previa en el manejo de vehículos aéreos. Con esto en mente, el objetivo de este proyecto es la creación de una interfaz intuitiva para el usuario medio para el control de navegación de un cuadricóptero mediante gestos corporales. La intención es ofrecer al usuario alternativas diferentes, y más cómodas, que su manejo mediante la emisora. Dar al propio usuario la capacidad de elegir lo que más le convenga en cada momento.



Para la consecución de dicho proyecto se pretende emplear el sensor Leap Motion [6]. Dicho sensor hace una lectura de los gestos de la mano. Anteriormente en [8] ya se estudió la viabilidad de los distintos sensores.

1.2 OBJETIVOS

El objetivo final de este proyecto es el control de navegación de un cuadricóptero mediante gestos corporales. Para alcanzar dicha meta se estudiarán las diferentes vías posibles que se irán abriendo paso y la mejor manera de conseguirlo:

Modelado y control del cuadrcóptero.

- Conseguir la adaptación e implementación del sensor Leap Motion con la lectura de los gestos, al entorno de Matlab/Simulink.
- Implantación del sensor para el control del cuadricóptero. Se harán las pruebas sobre un simulador de vuelo FlightGear.
- Montaje y control de un cuadricóptero físico para realizar pruebas y ensayos.

En la Figura 1 se muestra resumidamente el objetivo final del proyecto. Se pretende sustituir totalmente la emisora RC por el sensor Leap Motion para el control de navegación del cuadricóptero. A la estación base (GCS) le llegan las señales captadas por el Leap Motion por USB. El GCS y el UAV se comunican a través del protocolo de comunicación MAVLINK.

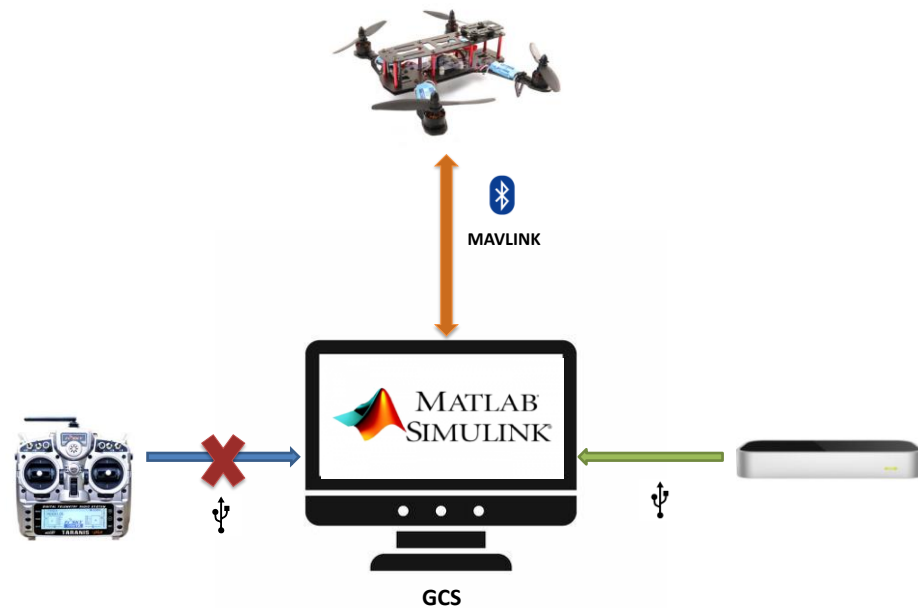


Figura 1: Objetivo del proyecto

1.3 METODOLOGÍA / SOLUCIÓN DESARROLLADA

La metodología que se adoptó para el desarrollo del proyecto fue la siguiente:

- Familiarización con el nuevo entorno Matlab/Simulink.
- Estudio e investigación del funcionamiento del sensor Leap Motion para captar los gestos de las manos.
- Adaptación del Leap Motion en el entorno de Matlab y Simulink.
- Procesamiento de los datos captados por el Leap Motion.
- Implementación del control del cuadricóptero.
- Configuración de una comunicación inalámbrica Bluetooth entre ordenadores para posteriores ensayos.
- Adaptación y estudio del funcionamiento de simuladores de vuelo como el FlightGear.



- Pruebas y ensayos del control de navegación del simulador de vuelo FlightGear con la emisora RC.
- Ensayos del control de navegación en el simulador mediante gestos corporales para revisar su funcionamiento.
- Montaje y configuración del cuadricóptero físico: tanto el software y hardware.
- Control de navegación del cuadricóptero mediante la emisora.
- Realizar pruebas del control de navegación del cuadricóptero con el Leap Motion.

1.4 RECURSOS / HERRAMIENTAS EMPLEADAS

Los recursos y herramientas que se pretenden emplear son:

- Leap Motion 2.0: sensor capaz de reconocer manos u objetos pequeños, captando de ellos las posiciones o velocidades.
- Módulo de Bluetooth HC-05: para realizar la comunicación serie.
- Matlab/Simulink versión 2018a: utilizado para el diseño de controles, implementación y del Leap Motion y simular el comportamiento de la aeronave.
- Raspberry Pi: tarjeta de control usada como *Hardware* para implantar los controles de la aeronave y simularlos.
- FlightGear: simulador de vuelo que se utilizó para las simulaciones en 3D de los controles.



1.5 ESTRUCTURA DE LA MEMORIA

En la memoria descriptiva se pretende hacer entender al lector de la forma más lógica el desarrollo del proyecto. Esta memoria descriptiva va desde lo más general del proyecto hasta lo más particular y específico.

La estructura general se puede dividir en las siguientes partes:

1. Investigación y estudio a fondo de los sensores de estos que se encuentran actualmente en el mercado. Capítulo 2.
2. Análisis y desarrollo del control y modelado del cuadricóptero así como de las diferentes partes que lo integran. Capítulo 3.
3. Todo lo relacionado con el estudio e implementación del Leap Motion en el control del cuadricóptero. Capítulo 4.
4. Ensayos y resultados de la implementación del Leap Motion en Simulación y en el simulador de vuelo Flightgear.
5. Conclusiones sobre el proyecto, reflexiones sobre posibles estudios desarrollos y búsqueda de nuevas propuestas o ideas.

Incluye todos los apartados que aportan información adicional a la memoria: bibliografías, presupuestos, anexos...



Capítulo 2 ESTADO DEL ARTE

2.1 INTRODUCCIÓN

La denominación de los UAV es muy general y poco específica. Su diseño tiene gran variedad de tamaños, configuraciones y características. Los drones tienen dos variantes: los controlados desde un lugar remoto y los de vuelo autónomo a partir de planes de vuelo preprogramados. En este proyecto, se empleará la primera variante. El término dron hace referencia a multihélices formados por diversas hélices: tricópteros, cuadricópteros, octacópteros... En este trabajo se ha trabajado con cuadricópteros, así que se hablará indistintamente de cuadricópteros y drones refiriéndose a UAV de 4 hélices.

A la hora de trabajar con los cuadricópteros es importante definir los movimientos de estos para dar una explicación concisa. Se ha estudiado de la aeronáutica las diversas rotaciones que realizan los vehículos aéreos alrededor de sus ejes. Son equivalentes estos movimientos tanto en aviones como en vehículos no tripulados, salvo en pequeños matices. Sus cuatro movimientos característicos son la guiñada, cabeceo, alabeo y empuje. El empuje se refiere a la elevación del dron a través de la fuerza ejercida por sus cuatro hélices. En este proyecto se hará referencia al empuje con el término inglés *throttle*, cuya traducción al castellano es *acelerador*.

El movimiento ejercido sobre el eje vertical corresponde a la guiñada. El cabeceo es la rotación respecto al eje transversal y el alabeo respecto al longitudinal. En este proyecto se hará referencia a los términos de guiñada, alabeo y cabeceo por su traducción al inglés *yaw*, *roll* y *pitch*, respectivamente. Estos ejes de rotación se pueden observar en la siguiente la Figura 2:

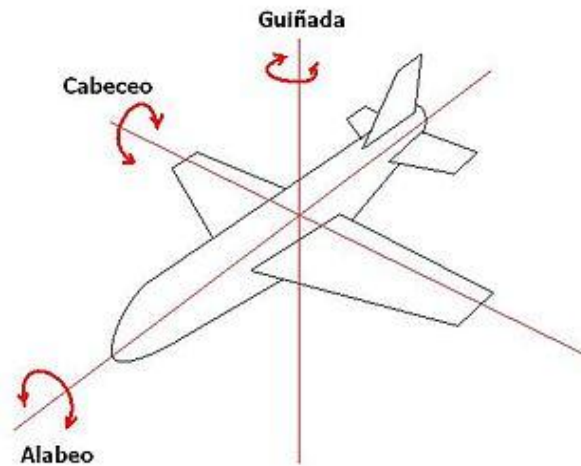


Figura 2: Ejes del avión

El estudio de estos movimientos tiene gran relevancia a la hora de elegir los gestos corporales más adecuados para controlar el UAV.

Se ha observado el mercado actual de los sensores, para elegir qué gestos y la manera en la cual se iba a implantar en Matlab.

Como se ha dicho anteriormente, la intención de este proyecto es la implementación de una interfaz intuitiva para el usuario medio. A continuación se explicarán las dos interfaces naturales de usuario (NUI) que mejor se ajustan a este proyecto. Estos son dispositivos que interactúan con un sistema sin utilizar sistemas de mando o dispositivos de entrada, que se ven sustituidos por movimientos gestuales del cuerpo. Los dispositivos de los que se ha realizado un estudio previo para este proyecto han sido el Kinect, Leap Motion y Myo Gesture Control Armband.



2.2 *KINECT*

Kinect es un dispositivo creado por Microsoft en 2011, capaz de hacer una lectura del cuerpo con las diferentes articulaciones. Los desarrolladores de software pueden hacer uso de él para gran variedad de aplicaciones cuyo destino es la interacción con los elementos virtuales con los distintos movimientos del cuerpo humano. El Kinect compite contra sistemas de otras consolas como el Wii Remote con Wii MotionPlus de Nintendo y PlayStation Move de la PlayStation 3 perteneciente a Sony.

El 2 de junio de 2009 Kinect fue presentado en la Electronic Entertainment Expo (E3) 2009 como *Proyecto Natal* [9]. Microsoft presentó el proyecto como el nacimiento de la próxima generación de entretenimiento en el hogar. En la misma exposición del siguiente año se presentó el Kinect (Figura 3) junto a la nueva consola Xbox 360. El Kinect permite controlar la consola sin necesidad de contacto físico para interactuar con los videojuegos mediante una interfaz de usuario que reconoce gestos e incluso comandos de voz. El 13 de junio de 2010, Microsoft reveló antes de su demostración en la Electronic Entertainment Expo de 2010 que el nombre oficial del dispositivo sería Kinect.

Finalmente, el 4 de noviembre de 2010 salió a la venta en Estados Unidos.



Figura 3: Kinect v1

Como se puede observar en la imagen, el Kinect consta de una barra horizontal de aproximadamente 23 cm que está conectada a una base circular mediante una rótula. Cuenta con una cámara RGB, un sensor de profundidad, un micrófono de múltiples matrices y un procesador personalizado que ejecuta el software patentado, que proporciona captura de movimiento de todo el cuerpo en 3D, reconocimiento facial y capacidades de reconocimiento de voz.

La Xbox 360 por medio del micrófono de matrices del sensor de Kinect es capaz de realizar la localización de la fuente acústica, suprimiendo el ruido ambiente y permitiendo participar en el chat de Xbox Live sin auriculares.

El sensor de profundidad está formado por un sensor CMOS monocromo combinado con un proyector de infrarrojos que permite a Kinect ver la habitación en 3D independientemente de la luz ambiental que halla en ese entorno específico. El rango de detección de la profundidad del sensor es ajustable gracias al software de Kinect con la capacidad de calibrar automáticamente el sensor, basado en la jugabilidad y en el ambiente físico del jugador, tal como la presencia de sofás.

Microsoft lanzó el 16 de junio de 2011 un SDK (Software Development Key) para aplicaciones de Windows 7, apoyado en el hardware del Kinect para la Xbox 360 conectado a un Pc para desarrolladores de aplicaciones no comerciales [10]. Este SDK tenía como finalidad permitir a usuarios escribir aplicaciones Kinect en C++ / CLI, C# o Visual Basic.



El 1 de febrero de 2012 se lanzó una versión de hardware similar a Kinect para Windows. La versión 1.0 del SDK de Windows, destinada al desarrollo de aplicaciones comerciales. En 2014 se lanzó al mercado el Kinect v2 para Windows junto con su SDK compatible. Windows comercializó esta nueva versión del Kinect para Windows con un adaptador como el que se muestra en la Figura 4 la conexión del adaptador con el PC se realiza a través de un puerto USB 3.0.

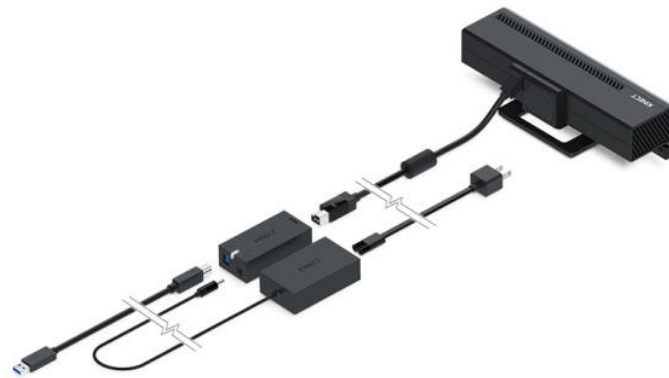


Figura 4: Adaptador Kinect

Entre ambas versiones del Kinect hay diferencias notables tanto en la resolución de las cámaras como en los datos que se obtienen de ellas. En la siguiente tabla se muestran las principales diferencias entre la primera y segunda versión [11].



Video	Kinect V1 640×480 @30 fps 1280×960 @12 fps	Kinect V2 1920×1080 @30 fps High Definition
Profundidad	*320×240, 640×480	*512×424
Rastreo del cuerpo	6 personas, pero solo dos rastreadas completamente	6 cuerpos, todos completamente rastreados.
Precio	Kinect for Xbox 360 \$150 dólares Adaptador para PC \$50 dólares Kinect for Windows \$250 dólares	Kinect for Xbox One \$150 dólares Adaptador para PC \$50 dólares Kinect for Windows V2 \$200 dólares
Motor de Inclinación	Si.	No (manualmente)
USB	2.0	3.0
Sistema Operativo	Win7 o superior	win 8.1 o superior

Tabla 1: Comparativa Kinect v1/v2



El dispositivo Kinect se acabó convirtiendo en un actor secundario de los pocos videojuegos que lo soportan, y cuando era protagonista competía con los clásicos juegos de la Wii. El Kinect nunca ha tenido la trayectoria esperada en el sector de los videojuegos, pero su tecnología ha sido aplicada en muchos otros servicios y dispositivos. El accesorio nunca ha tenido gran popularidad entre los consumidores por lo que se paró la producción en abril de 2016 para Xbox 360 y en octubre de 2017 para Xbox One

2.3 LEAP MOTION

Es una compañía americana fundada en 2010 en San Francisco por Michael Buckwald y David Holz. Su actividad se centra en la producción y venta de sensores para ordenadores que leen como entrada el movimiento de los dedos y de las manos, análogo al ratón de ordenador, pero este no requiere de ningún contacto. Desde 2016, esta compañía focaliza todas sus actividades e inversiones en nuevos desarrollos de software para la realidad virtual.

Leap Motion es un dispositivo pequeño capaz de reconocer manos u objetos pequeños, captando de ellos las posiciones o velocidades [6]. Poco se puede decir de cómo funciona debido al secreto de los desarrolladores. Lo único que se sabe es que usa dos cámaras monocromáticas IR y tres Leds infrarrojos. Sorprende su pequeño tamaño y su gran precisión (Figura 5) y con un precio asequible.



Figura 5: Leap Motion

La tecnología del Leap Motion empezó a ser desarrollada a principios del 2008. La idea surgió de la tesis doctoral en matemáticas del cofundador de la compañía David Holz en la universidad de Carolina del Norte. Holz fundó la empresa junto con su amigo de la infancia Michael Buckwald. Desde entonces, la compañía había trabajado en silencio y en mayo de 2012 presentaron públicamente su primer producto: *The Leap*. El dispositivo fue variando de tamaño hasta que llegaron al actual (Figura 6). El dispositivo se puso a la venta en Octubre de 2012. La compañía lanzó un programa de desarrollo de software en octubre de 2012 [12] y distribuyó aproximadamente 12,000 unidades a los desarrolladores interesados en crear aplicaciones para el dispositivo. Mientras que el dispositivo estaba programado para su lanzamiento en mayo de 2013, el envío a gran escala se retrasó más tarde hasta julio. La venta del *Leap* no tuvo la aceptación esperada en el mercado. A pesar de las innovaciones, el producto no estaba a la altura de las expectativas, ya que realmente no era un instrumento productivo. La compañía se vio obligada a reducir el número de empleados.



Figura 6: Evolución Leap Motion

En mayo de 2014, Leap Motion lanzó su segunda versión de software para desarrolladores: el *Leap Motion 2.0*. Éste es el dispositivo que se ha empleado para realizar el presente proyecto. En agosto de 2014, la compañía lanzó un modo de seguimiento de realidad virtual para su software central basado en cascos de realidad virtual [13]. En marzo de 2015, se anunció que el próximo OSVR (Open-Source Virtual Reality) Hacker Development Kit que incluía una placa frontal con un módulo opcional para incluir el Leap Motion (Figura 7). Éste consiste en un casco diseñado para que los ojos puedan ver, a través de las gafas, la realidad virtual. El *open source software* permite a los desarrolladores de realidad virtual detectar, configurar y operar dispositivos de realidad virtual en una amplia gama de sistemas operativos. En febrero de 2016, Leap Motion lanzó un nuevo software, llamado Orion, creado específicamente para VR [6].



Figura 7: VR Leap Motion

En este proyecto no se han empleado las gafas de realidad virtual, ya que no se encuentran dentro de nuestros objetivos. Aunque, por otro lado, si que se puede tener en cuenta este dispositivo para futuros desarrollos. Partiendo de la base de este proyecto, se podría investigar sobre las gafas de realidad virtual.

2.4 MYO GESTURE CONTROL ARMBAND

Myo Armband es un dispositivo creado por Thalmic Labs. Se coloca en el antebrazo como si fuera un brazalete. Este producto permite controlar inalámbricamente la tecnología con los movimientos y gestos de la mano. Utiliza un conjunto de sensores electromiográficos (EMG) que detectan la actividad eléctrica en los músculos del antebrazo (Figura 8). La electromiografía es el estudio destinado a conocer, mediante sensores, la actividad eléctrica muscular [15]. Myo puede diferenciar entre los distintos movimientos de los dedos y detectar rotaciones y el movimiento de la mano mediante la medición de los distintos patrones de impulsos eléctricos que generan los movimientos, y usa un



sensor inercial para entender dichos gestos. Las fibras musculares, al contraerse, producen descargas que, recogidos por estos electrodos, dan unas señales que luego se traducen en acciones y órdenes.



Figura 8: Brazaletes Myo Armband

Los desarrolladores han creado aplicaciones que permiten al dispositivo hacer de todo, desde controlar sistemas de realidad virtual hasta instrumentos musicales. Una de las mayores aportaciones de este sensor puede ser al avance de la medicina. Actualmente investigadores del Laboratorio de Física Aplicada Johns Hopkins están cambiando la vida de personas con extremidades amputadas al darles manos o piernas protésicas controladas por medio de este sensor. También está revolucionando la forma de trabajar de los cirujanos. Junto con varias cámaras y software de reconocimiento de voz, el Myo está dotando a los médicos de una mayor precisión en ciertos ámbitos de la práctica de la cirugía [16].

La invención del Myo nació a raíz de un proyecto elaborado por los fundadores y directores de Thalmic Labs: Matthew Bailey y Stephen Lake en el departamento de ingeniería mecatrónica de la Universidad de Waterloo. Este proyecto trataba de un láser que escaneaba obstáculos y traducían un feedback táctil para ayudar a los



invidentes. Aunque no guarda mucha relación con el Myo, este proyecto les hizo pensar en cómo implantar sensores en el cuerpo humano.

Atualmente, Myo se ha configurado para dispositivos iPhone, iPad, ordenadores Mac Y Windows, Raspberry Pi, Parrot AR.Drone... Thalmic Labs está investigando como mejorar el Myo con el casco de realidad virtual Oculus Rift (Figura 9), y con Google Glass, el ordenador de Google que se lleva puesto en la cabeza [17].



Figura 9: Myo con Oculus Rift

A raíz de su buen funcionamiento al aire libre, este dispositivo resulta de gran utilidad para controlar un avión no tripulado. Un proyecto de la Universidad de Stuttgart en Alemania utilizó este brazalete para conseguir controlar el Parrot AR.Drone 2.0 [18].

Supone una gran ventaja del Myo sobre otros sensores gestuales que no dependa de cámaras o que no requiera un hardware voluminoso. Casi no ocupa espacio y no tiene problemas de limitación con la luminosidad ambiental como el Kinect, tanto bajo la luz solar brillante como con los cuartos oscuros. El único inconveniente es que no llega a captar todos los movimientos ejecutados con la mano. En función de su finalidad puede llegar a ser muy útil.



En [8] se ha estudió la viabilidad de los sensores Kinect y Leap Motion para el control de navegación por gestos de un cuadricóptero. En ese proyecto se ha hecho una primera aproximación al control de gestos de un cuadricóptero. Se concluyó que a pesar de un par de dificultades, es más sencillo y cómodo utilizar el Leap Motion que el Kinect en su implementación en Matlab y Simulink. Lo que ha hecho diferenciar el Leap del Kinect en la comparativa de ambos dispositivos ha sido su mejor procesamiento de la información y la estabilidad en la lectura de las medidas independientemente del manejo del piloto. Por estas razones, se ha decidido descartar el Kinect y realizar este proyecto exclusivamente con el Leap Motion.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Estado del Arte



Capítulo 3 MODELADO Y CONTROL DEL CUADRICÓPTERO

En este apartado de la memoria se analiza y se muestra el modelado y control del UAV. También se exponen los distintos materiales usados para la construcción y ejecución del cuadricóptero.

3.1 MATERIAL UTILIZADO PARA LA FABRICACIÓN

Se ha realizado el montaje de un UAV para poder hacer las pruebas y ensayos. Hay que añadir que este proyecto no se centra en la construcción, sino en el control del cuadricóptero.

Los UAV pueden tener diferentes diseños y tamaños, pero su principal característica es que sean lo suficientemente resistentes para soportar las fuerzas opuestas de los motores sin flexionar y sin llegar a la ruptura. El material empleado en la fabricación tiene que tener propiedades de ser ligero, fuerte y rígido. Estos materiales deben de reducir al máximo la vibración producido por los motores. Por este motivo, la estructura se compone de fibra de carbono. Este material empleado ha sido proporcionado por la universidad.

Debido a la peor resolución de los sensores con respecto a la emisora RC, se ha buscado la fabricación de un cuadricóptero pequeño y sencillo de manejar para reducir en la mayor parte de lo posible las pérdidas por choques fortuitos.



3.1.1 MOTORES

La elección de los motores adecuados es muy importante para un buen rendimiento del dron. Se utilizan motores que son compatibles con la utilización de baterías LiPo de corriente continua. Es necesaria la utilización de motores pequeños con gran potencia y sin escobillas, también llamados *brushless*. La mayoría de los motores *brushless* utilizan corriente alterna y las baterías proporcionadas por la universidad utilizan corriente continua. Aquí es donde entra en juego los variadores o ESC's. Se usará PWM (Pulsos de Anchura Modulada) para conseguir corriente alterna.

Para poder hacer volar el cuadricóptero, debe de existir una relación entre su peso y la suma de la potencia de los motores, ya que con poca fuerza el cuadricóptero no es capaz de despegar y con mucha puede hacerse incontrolable. Esta proporción es entre el 40% y el 60% inferior el peso total del cuadricóptero a la fuerza de los motores. Si el peso estimado del dron es de aproximadamente 300 g, calculamos la proporción midiendo la potencia de distintos motores.

El que se mejor se ajusta a esta relación potencia/peso han sido los motores DragonFly (Figura 10). Son pequeños motores trifásicos síncronos, que emplean bobinas e imanes permanentes que generan un campo magnético giratorio.

Se han visto en las especificaciones eléctricas de estos motores que son compatibles con las baterías de Lipo de 7.4 y 11.1 voltios.



Figura 10: Motores Dragon Fly MC1806 2300kv



- 200g de potencia máxima por motor * 4 motores = 800g totales de máxima potencia.
- La franja de peso del dron está comprendida entre 280 y 312g.

Por motivos del pequeño tamaño del UAV, se han utilizado hélices bipalas. A continuación se ha realizado un ensayo sobre un banco de potencia (Figura 11) para medir la potencia de los motores. En un extremo se coloca el motor con las palas para que el banco de la fuerza en gramos.

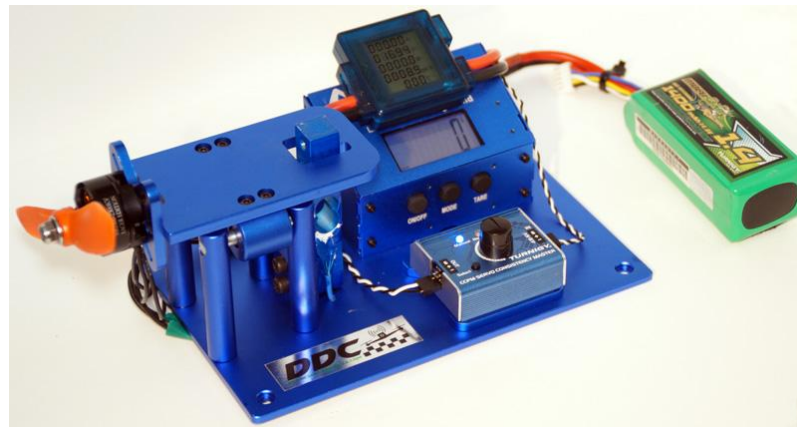


Figura 11: Banco de potencia

Con el banco de potencia se ha medido los gramos de empuje en función del PWM (pulsos) en microsegundos que será de utilidad posteriormente para la configuración del cuadricóptero. En la siguiente prueba se muestra el resultado del ensayo:



Pulsos (μs)	Empuje (g)
1000	0
1050	14
1100	19
1150	31
1200	38
1250	53
1300	72
1350	85
1400	100
1450	114
1500	128
1550	142
1600	161
1650	178
1700	198
1750	210
1800	208
1850	204
1900	203
1950	200
2000	200

Tabla 2: Potencia motores

A continuación se muestra una gráfica con los valores de la tabla anterior. La pendiente es aproximadamente lineal como se buscaba desde un principio. El pico lo encontramos el $1750\mu\text{s}$ debido a la descarga de la batería durante el ensayo.

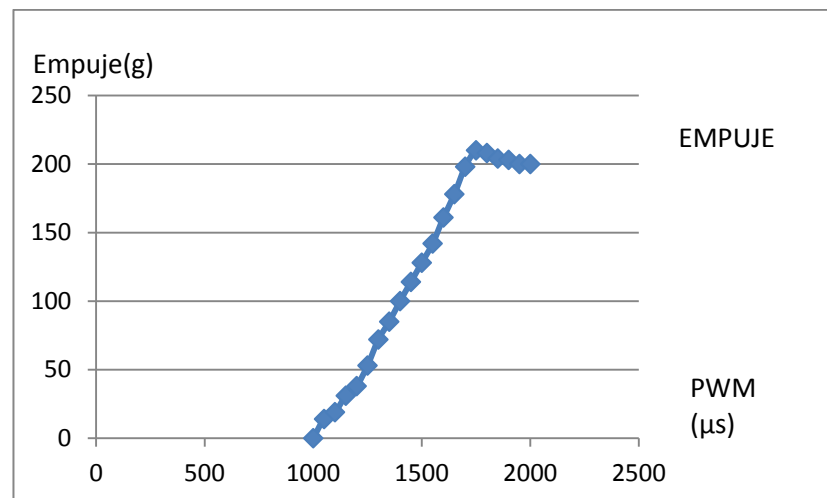


Figura 12: Empuje motores



3.1.2 BATERÍA

Para la realización del presente proyecto se han empleado una batería LiPo de 1000 mAh con un voltaje nominal de 11.1V de tres celdas (Figura 13). LiPo proviene de la abreviatura de Litio y polímero. Son un tipo de baterías recargables, muy habitual para el uso de drones y radiocontrol en general. Se caracterizan por su densidad energética. Son una fuente de energía potente y eficiente en relación al tamaño que posee. Tan solo pesa 80 gramos, ideales para el empleo de drones. Se considera que han sido una de las piezas clave en el desarrollo y evolución de los vehículos aéreos.



Figura 13: Batería Lipo 11.1 voltios

A lo largo de este proyecto se han utilizado diferentes baterías: de 7.4 voltios de 2 celdas y 11.1 voltios de tres celdas. Se ha escogido un cargador que tuviera la capacidad de cargar baterías LiPo con la opción de escoger distinto número de celdas. Así mismo, se ha buscado que fuese balanceador para regular la carga de cada celda y que no haya sobrecargas. Con estos requisitos se ha utilizado en cargador IMAX (Figura 14):



Figura 14: Imax Lipo Balance Charger

3.1.3 RASPBERRY PI ZERO W

Las Raspberry Pi son microcontroladores de bajo coste desarrolladas en Reino Unido desde el 2006. Su software es open source, siendo Raspbian su sistema operativo oficial, aunque permite el uso de otros sistemas operativos. En todas sus versiones se incluye un procesador Broadcom, memoria Ram, GPU, puertos de USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara. No incluyen memorias, por lo que es necesario el uso de tarjetas SD para sus primeras versiones y MicroSD para las versiones posteriores.

Las versiones Pi Zero W (Figura 15) son mucho más pequeñas y baratas que sus hermanas. Debido al peso, son ideales para implantarlas en el cuadricóptero. El puerto HDMI se ve sustituido por MiniHDMI, el puerto USB por MicroUSB y también hay puerto para MicroSD. La W es por *Wireless*, que se refiere a la innovación respecto a las antecesoras de incluir conexión Wifi y Bluetooth. Esta novedad permite controlar con mayor facilidad el dron desde la estación base. Para este proyecto la conexión se efectuará por wifi.



Figura 15: Raspberry Pi Zero W

Las Raspberry Pi se diferencian de los microcontroladores STM por poseer una mayor velocidad de cálculo y un sistema operativo en tiempo real.

3.1.4 PFX MINI

Diseñada para acompañar a la Raspberry Pi Zero, siendo también compatible con el resto de plataformas de la familia Raspberry Pi. Su montaje se efectúa sobre la Raspberry y pesa solamente 15 gramos. El PFX- Incluye la unidad de medición inercial (IMU) y está compuesto por giróscopos y acelerómetros, así como de barómetros, magnetómetros, sensores de temperatura y monitor de batería. De esta forma se monitorizan constantemente los cambios sobre el UAV, tales como el cabeceo, la guiñada y alabeo. A continuación se muestra una figura del PFX Mini:



Figura 16: PFX Mini



3.1.5 ESC's (ELECTRONIC SPEED CONTROL)

Un control electrónico de velocidad electrónico o ESC es un circuito inversor que controla y regula la velocidad de un motor eléctrico. A partir de una corriente y una señal PWM que controla la frecuencia, genera corriente alterna trifásica. Se usan estos controladores porque son la mejor opción para trabajar con motores sin escobillas al ser los encargados de transmitir la potencia en forma de corriente alterna trifásica.

Para este proyecto los ESC se ha alimentado a la placa de potencia y colocada entre esta y los motores.

Los ESC se clasifican en función de la corriente máxima, por ejemplo en este proyecto se ha usado una de 20 amperios o 20A (Figura 17). En general, cuanto mayor es la calificación, más grande y más pesado tiende a ser el ESC, que luego hay que tenerlo en cuenta para la masa y equilibrio del UAV.



Figura 17: SC 20A

3.2 COMUNICACIONES

Al comienzo de este proyecto se estudió a fondo la radio Crazyradio la cual tiene un receptor incorporado en el cuadricóptero Crazyflie 2.0 para establecer la comunicación para controlar dicho dron. Esta comunicación está preparada para



funcionar dentro de una API en Python [3.1], y se concluyó que no entraba en los objetivos de aprendizaje del proyecto. En [2] se descartó utilizar una comunicación Bluetooth por la baja potencia de la batería para realizar la comunicación con el Crazyflie. Tampoco era posible añadirle una batería más grande ya que el peso del Crazyflie aumentaría considerablemente.

Descartadas estas opciones, se decidió construir un dron con mayor peso que permitiese una comunicación inalámbrica Bluetooth que fuera efectiva sin interferir en su vuelo. Esta emisora conecta el dron con nuestra estación base y se comunica como un puerto UART. En particular se ha utilizado la HC 05 (Figura 18). El cuadricóptero tiene la capacidad de llevarlo encima ya que este módulo solo pesa 10 gramos.



Figura 18: HC 05

3.2.1 UART

Uart, son las siglas en inglés de Transmisor-Receptor Asíncrono Universal (*Universal Asynchronous Receiver-Transmitter*). Es un dispositivo que controla los puertos serie.

Tx se refiere a la transmisión de datos y Rx a la recepción de datos, como se muestra en la Figura 19. Tanto el emisor como el receptor tienen que tener la misma referencia a tierra (Gnd) para que sea efectiva la transmisión y recepción de datos.

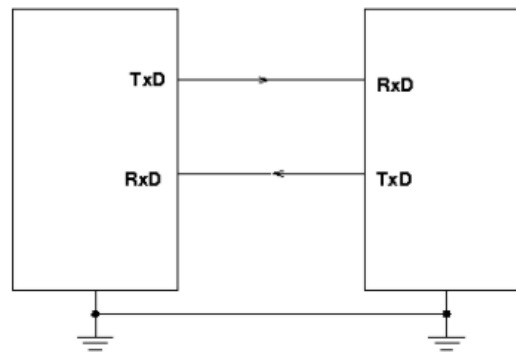


Figura 19: Conexión UART

Como la comunicación es asíncrona, el proceso de sincronización entre emisor y receptor se realiza en cada palabra transmitida. Esta sincronización se lleva a cabo a través de unos bits especiales que definen el entorno de cada código. Debido a que el receptor no sabe cuando recibirá el mensaje, cada palabra transmitida comienza por un bit de arranque que sincroniza el transmisor con el receptor. A continuación se envían los bits con la información. Por último, uno o dos bits de parada, para separar los distintos caracteres. En la siguiente imagen (Figura 20) se muestra una gráfica para mayor comprensión.

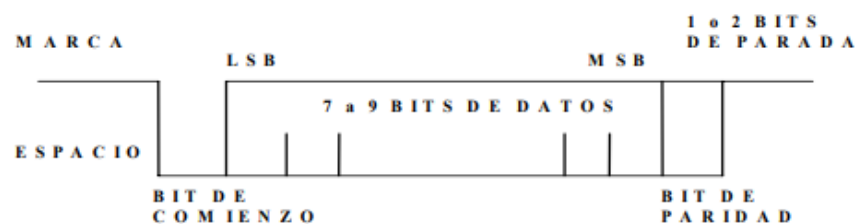


Figura 20: Protocolo comunicación Uart

La tasa de transmisión de un puerto UART se mide en baudios. El baudio es una unidad de medida utilizada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión digital. En este caso coincide el número de símbolos con el número de bits. La velocidad de transmisión puede ser de: 9600, 19200, 38400, 57600, 115200, 230400 ó 460800 baudios.



Para este caso configuramos nuestro ordenador como maestro y el cuadricóptero como esclavo. La velocidad de transmisión será de 57600 baudios. Otros parámetros de la comunicación son: paridad impar, ocho data bits y dos bit de stop (Figura 21).

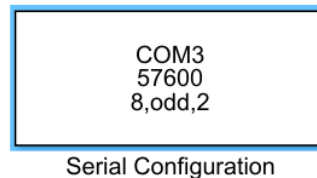


Figura 21: Serial Configuration UART

No se usarán cables eléctricos, sino que la comunicación será inalámbrica a través del Bluetooth.

3.2.2 MAVLINK

El nombre proviene de las siglas en inglés: Micro Air Vehicle Link. Es un protocolo de comunicación utilizado para pequeñas aeronaves no tripuladas. MAVLink fue lanzado por Lorenz Meier por primera vez a principios de 2009.

Se ha utilizado para establecer el protocolo de comunicación entre la estación base (PC Control Station) y nuestro cuadricóptero no tripulado.

Se han configurado una serie de mensajes, con su registro correspondiente y con un tamaño determinado [19]. En este proyecto se ha utilizado mensajes en dos direcciones: una desde la estación base al UAV y del UAV de vuelta a la estación base.

Los mensajes que se envían desde PC Control Station al dron son los registros con los valores deseados de guiñada, cabeceo, alabeo, posición de la mano en los distintos ejes mencionados anteriormente y la apertura de la mano junto con la lectura de ambas manos, respectivamente.



Como se dirá posteriormente, las visualizaciones del dron se realizarán desde la estación base. Por lo tanto, del cuadricóptero a la estación base se envía la información correspondiente al posicionamiento y los ángulos actuales de este.

3.3 EMISORA RC

Se utilizan las emisoras como sistemas de radio-control de UAV. Lo que se busca con este proyecto es conseguir sustituir con la mayor precisión posible la emisora por el Leap Motion. También se pretende ofrecer al usuario la capacidad de complementar ambos y utilizar en cada momento el sistema que prefiera en función de los movimientos que vaya a realizar con el cuadricóptero. Esto implicará un estudio exhaustivo de la emisora de radiocontrol para comprender su funcionamiento para sustituirlo por el Leap Motion. Para este proyecto se ha utilizado la emisora *Tanaris Plus X9D* (Figura 22). Esta emisora puede funcionar con el sistema inalámbrico de radiocontrol o si se prefiere por cable USB. Luego la estación base (PC_control_station) es la encargada de enviar el mensaje al dron.



Figura 22: Tanaris Plus X9D

Esta emisora consta de 16 canales, de cuatro cardanes de cojinete de bolas, un sistema de dos frecuencias de dos módulos, tres modos seleccionables y una batería de 200mAh.

Se conecta la emisora al ordenador mediante cable USB y se recibe los diferentes valores en el bloque de Simulink (Figura 23). Mediante un ajuste de ganancias, se busca obtener valores en el rango entre 1 y -1 de la posición de los distintos joystick de la emisora: impulso (altura), alabeo, cabeceo y guiñada, respectivamente.



Figura 23: Bloque Joystick Input Simulink



3.4 CONTROL DEL CUADRICÓPTERO

El control del cuadricóptero está formado por dos lazos: uno interno, llamado control de estabilización y otro externo, el control de navegación o autónomo. El control del UAV está formado por cuatro controles PID colocados en cascada: dos en el control de estabilización y dos en el de navegación. El control PID posee unos mandos no lineales, los cuales a partir de unos mandos virtuales se linealizan para la obtención de los mandos reales. El cuadricóptero tiene seis grados de libertad: tres de posición correspondientes a los tres ejes (x,y,z) y tres de ángulos (guiñada, cabeceo y alabeo). La posición está incluida en el control de navegación y los ángulos en el control de estabilización.

En el siguiente esquema (Figura 24) se muestra resumidamente la estructura del control. La emisora RC se encarga de introducirle referencia de ángulos en el control de estabilización. En este proyecto se intentó atacar con ángulos el control de estabilización pero el Leap Motion no tiene la misma resolución que la emisora. Finalmente se ha tenido que dar referencias de velocidades en el control de navegación. De esta forma se consigue un manejo del UAV más intuitivo que la emisora.

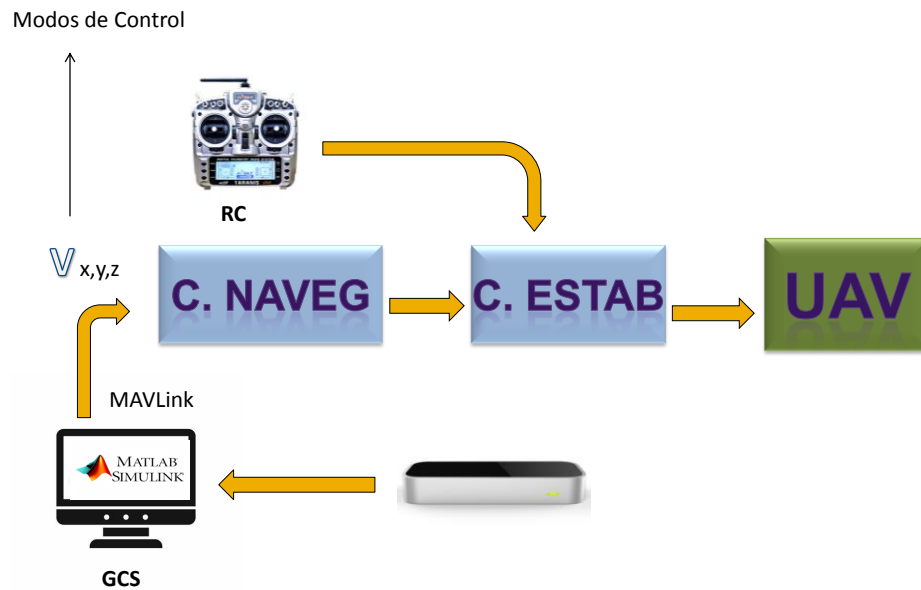


Figura 24: Estructura del control

El control de estabilización se encarga de dar el par correspondiente a los ángulos al Mixer. El Mixer es el encargado de convertir los mandos en señales correspondientes a los ESC para controlar los giros de los motores. ESC es un circuito inversor que genera corriente alterna trifásica a partir de corriente continua y una señal PWM que controla la frecuencia. Las cuatro señales del PWM tienen un valor de entre 1000 y 2000 microsegundos.

El sistema de control del cuadricóptero está dividido en tres bloques de simulink: Hardware, Control y Simulation (Figura 25). En función de si se quiere hacer volar un UAV físico o en simulación habrá que desactivar/activar el bloque de Simulation o el de Hardware. Las primeras sesiones de pruebas son realizadas en simulación, ya que es más fácil detectar los errores. Una vez pasadas las simulaciones, se realizarán los últimos tests sobre el hardware real.

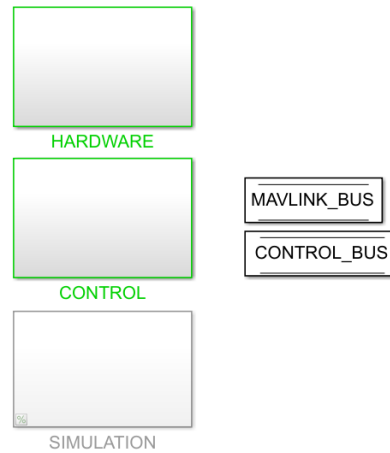


Figura 25: UAV Control System en Simulink

3.5 ESTACIÓN BASE

En este proyecto se hablará indistintamente de Estación Base o en inglés Ground Control Station (GCS) o PC Control Station.

La función de la estación base es la interrelación de los diferentes dispositivos que entran en juego a la hora de manejar un UAV no tripulado. Se encarga de recibir las señales de los dispositivos, procesarlas y enviarlas ya con los valores deseados a través el protocolo MAVLink. Típicamente la estación base incluye interfaces de usuario. Procesa la información que le llega de la emisora RC o del Leap Motion para que, luego, mediante la comunicación inalámbrica se pueda controlar el dron. En la Figura 26 se muestra la interconexión de dispositivos para realizar el vuelo del cuadricóptero o su simulación.

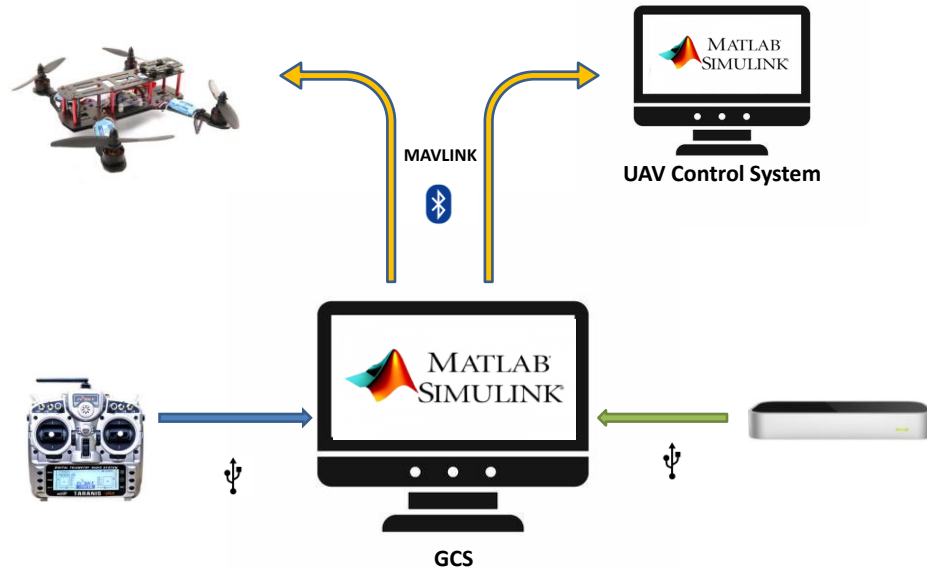


Figura 26: Interconexión de dispositivos

La elección del dispositivo a emplear se realiza activando los bloques correspondientes y desactivando los bloques de Simulink que no se desean (Figura 27). Estas se encuentran dentro del bloque de HARDWARE de Simulink.

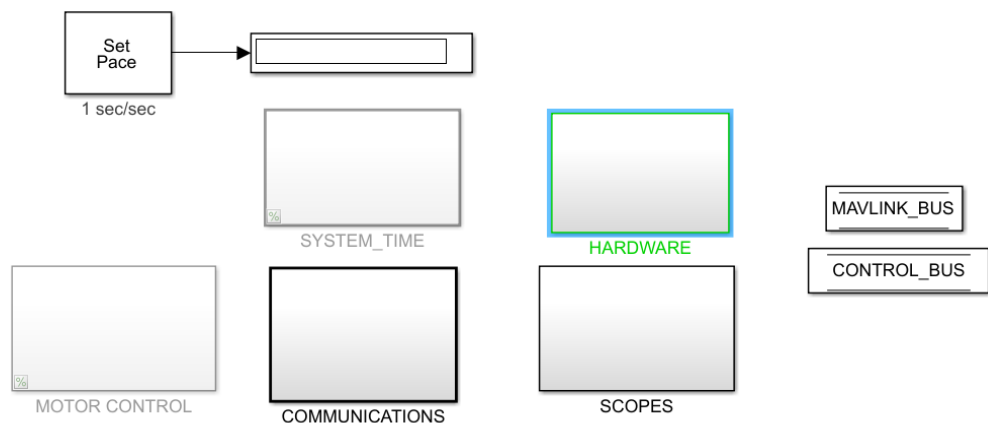


Figura 27: PC Control Station



La forma de comunicación entre la estación base y el cuadricóptero es a través del bluetooth. Por medio de un USB, se conecta el módulo Bluetooth al ordenador, por lo que hay que configurar el puerto COM correspondiente.



Capítulo 4 LEAP MOTION

Leap Motion es un sensor capaz de captar movimientos, ángulos, aperturas y posiciones de las diferentes partes de las manos. En este capítulo se explicará las señales que se han obtenido del sensor y así como el procesamiento que se ha hecho a esta información para poder llegar a controlar nuestro cuadricóptero.

4.1 HARDWARE

Los LEDs emiten rayos infrarrojos y estos se reflejan en los objetos. Según como estén de cerca o lejos los objetos la cámara de infrarrojos los recibirá con diferentes intensidades. En el caso de Leap Motion tiene dos cámaras infrarrojas separadas como se puede ver en la Figura 28. Ambas cámaras tienen unas lentes de gran ángulo que permiten una visión de 120° en el eje Z y 150° en el X (Figura 28). Además, tiene un rango de visión de 80 cm.

El controlador de Leap Motion es un dispositivo periférico de USB, diseñado para ponerlo entre el ordenador y el propio usuario apoyado sobre la misma mesa. Este sensor tiene unas dimensiones muy reducidas en comparación a otros interfaces gestuales en el mercado: mide 75 mm de largo, 25 mm de ancho y 11 mm de alto.

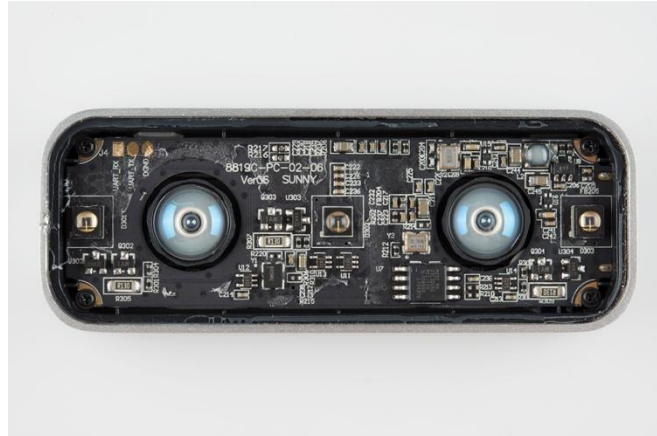


Figura 28: Hardware Leap Motion

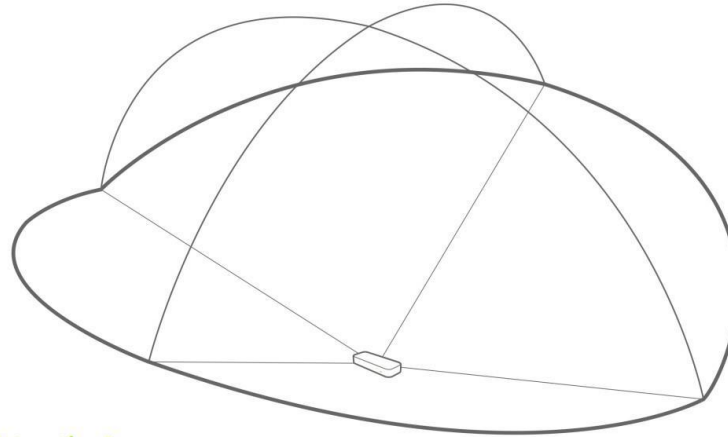
Como se puede observar en la Figura 28, Leap Motion está compuesto por dos sensores de infrarrojos separados y simétricos entre sí y tres LEDs de infrarrojos. Estos LEDs emiten rayos infrarrojos y estos se reflejan en los objetos. Según como estén de cerca o lejos los objetos la cámara de infrarrojos los recibirá con diferentes intensidades.

Las cámaras del Leap Motion son las encargadas de capturar imágenes a través de un sensor monocromático, sensible a la luz infrarroja, con una longitud de onda de 850 nm. Dependiendo de la capacidad del ordenador, puede trabajar a una velocidad de hasta 200 fps.

Los Leds, como es lógico, trabajan en el espectro de luz infrarroja a una longitud de onda de 850 nm. Se consumo eléctrico se ve variado en función de la intensidad luminosa de la zona de trabajo, para asegurar una mejor resolución de la imagen.

Los datos se envían y se reciben al controlador del ordenador a través de dos puertos serie: UART_RX y UART_TX.

En la *interaction box* de la figura (Figura 29) se ve el área de interacción del dispositivo. Como se puede observar, este área se asimila a una semiesfera de aproximadamente de 61 cm de radio.



Interaction Area
2 feet above the controller, by 2 feet wide on each side
(150° angle), by 2 feet deep on each side (120° angle)

Figura 29: Interaction Box

Tanto el ángulo de visión horizontal como el vertical son de $150,92^\circ$. La zona de interacción se verá delimitada por estos ángulos. El volumen de la caja de interacción consta de 110.55 mm de altura, 110.55 mm de anchura, 69.43 mm de profundidad, aunque las dimensiones pueden variar en función del entorno de trabajo en el que se encuentre. Esta es la zona en la que se marca el centro del sistema de coordenadas cartesiano de Leap Motion.

4.2 SOFTWARE-MATLEAP

Para poder empezar a usar Leap Motion es necesario instalarse la aplicación propia que incorpora drivers para funcionar en PC.

En [5] se ha visto que este dispositivo no puede ser controlado por el Image Acquisition Toolbox de Matlab. Por lo que se han encontrado unas librerías de mex-files que pueden ser ejecutadas en Matlab y proporcionan ciertas funciones. Las mex-files son un conjunto de archivos escritos en C que crean funciones que



luego se pueden utilizar en el entorno de Matlab y con su lenguaje. Este conjunto de Archivos de Matlab se denomina Matleap.

El dispositivo utiliza en el sistema de referencias cartesiano y tiene su origen (0, 0, 0) de los ejes cartesianos (x, y, z) en el centro del dispositivo (Figura 30).

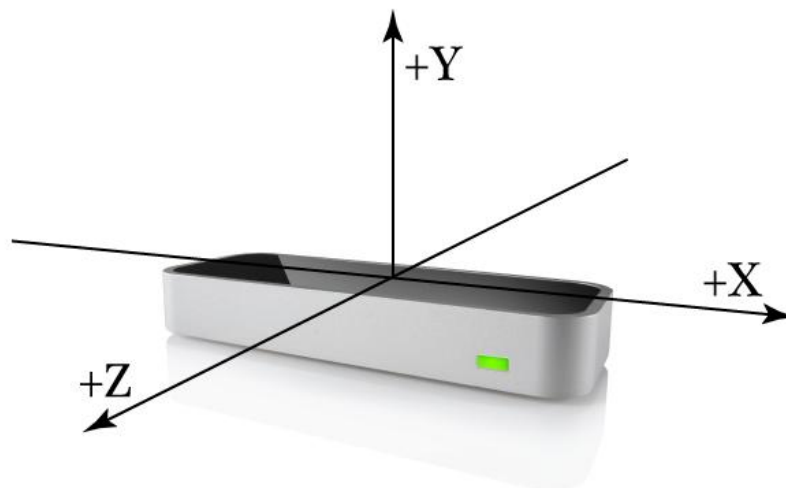


Figura 30: Ejes del Leap Motion

En la figura se muestran los ejes cartesianos. El eje X corresponde al eje horizontal, el eje Y a la altura en el eje vertical y el eje Z, a la distancia frontal del sensor.

4.3 LECTURA DEL DISPOSITIVO

Para poder hacer la lectura de este dispositivo en Simulink lo que se ha creado en [8] ha sido una S-Function de segundo nivel en Matlab. Las mex-files encontradas no sirven para Simulink, solamente para el entorno de Matlab. Las s-functions permiten crear un código en lenguaje Matlab y ejecutarlo dentro de un script de Simulink (Figura 31).

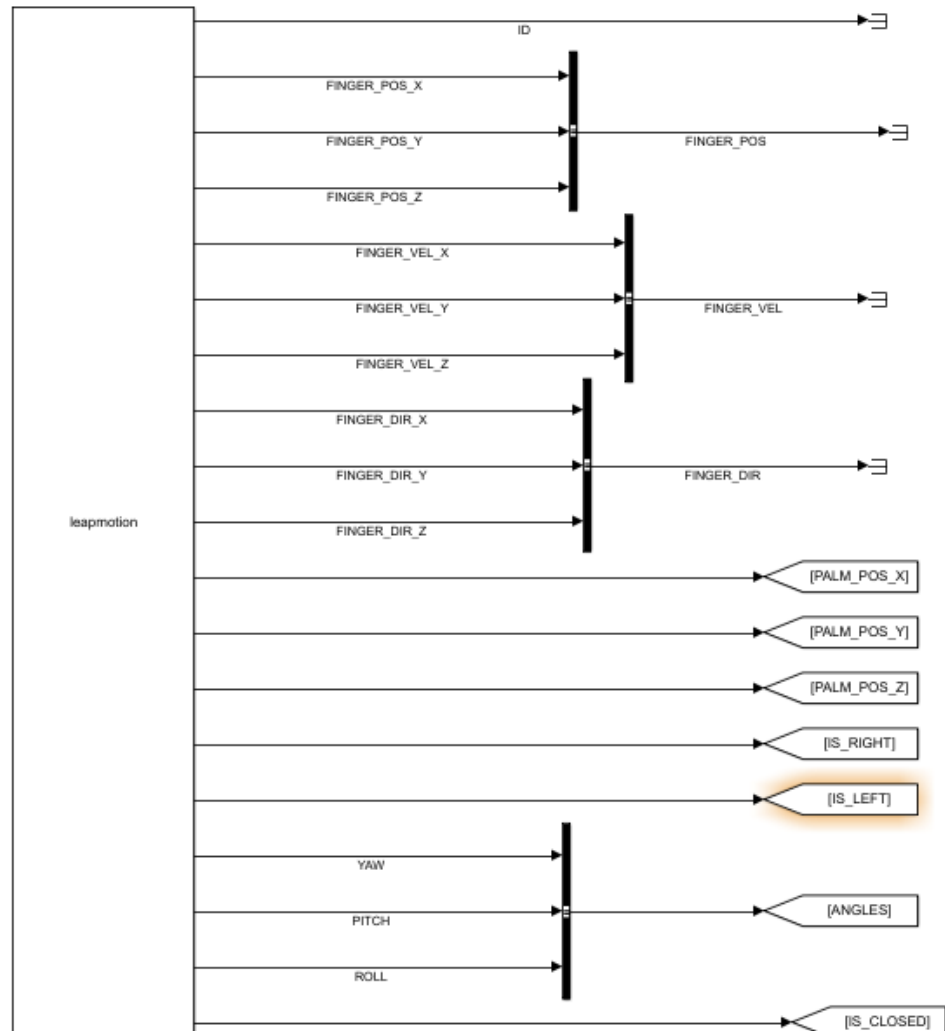


Figura 31: Bloque Leap Motion en Simulink

En la Figura 31 se han colocado un *Terminator* en aquellas variables que no se le han dado utilidad en este proyecto.

Las variables de entrada que se obtienen del dispositivo Leap Motion al hacer la lectura de las manos son:

- FINGER_POS: hace una lectura de la posición en milímetros de los 10 dedos de las manos en los distintos ejes (x, y, z)

- FINGER_VEL: da la velocidad de los 10 dedos en los distintos ejes del sistema de referencias (x, y, z). La derivada de las variables de la *Position* respecto al tiempo.
- FINGER_DIR: llega como entrada el valor de la dirección de hacia dónde apuntan los distintos dedos de ambas manos.
- PALM_POS: posición en cm del centro de la palma de la mano en el eje x, y, z. Como se mostrará a continuación, será de gran utilidad.
- IS_RIGHT: hace una lectura binaria de la mano derecha. Da un 1 si lee la mano derecha y 0 si no la lee.
- IS_LEFT: hace una lectura binaria de la mano izquierda. Si lee la mano izquierda da un 1 y 0 si no lee nada.
- ANGLES: de aquí se obtienen las medidas en grados de los distintos ángulos de la mano (guiñada, cabeceo y alabeo) como se demuestra gráficamente en la siguiente imagen (Figura 32). Como se verá posteriormente, estos ángulos serán de gran utilidad para manejar el cuadricoptero.

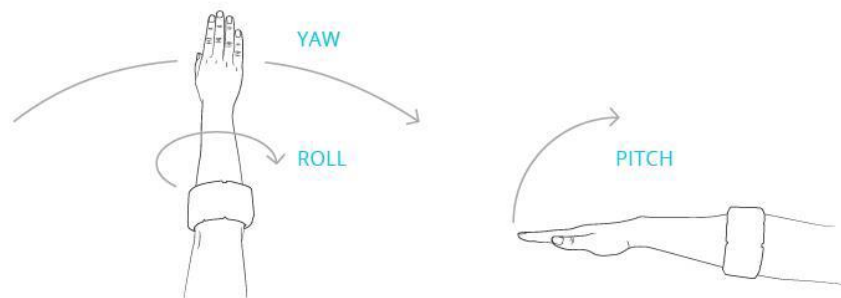


Figura 32: Ángulos de la mano

- IS_CLOSED: hace referencia a las aperturas de las manos. Cabe destacar que no es valor continuo y no binario, pero si delimitado por un mínimo 0 (totalmente abierta) y un máximo de 1 (totalmente cerrada) y admite



valores intermedios con la mano medio cerrada y medio abierta. Esta variable será clave para desarmar los motores.

4.4 COMPROBACIONES Y PRUEBAS

Con el objetivo de comprobar el funcionamiento del Leap Motion, se han realizado ensayos visualizándolos en *Scopes* de Simulink de los diferentes ángulos y posiciones de la mano para que, posteriormente, se pueda procesar los datos de la manera más conveniente.

El primer ensayo realizado consiste en mover las manos a lo largo del eje X. Para ello visualizamos en un *Scope* (Figura 33) la variable PALM_POS_X. La mano derecha se representa de color rojo y la izquierda de color azul. En la gráfica, el eje de abscisas corresponde a un tiempo de muestreo de 30 segundos y el de ordenadas a la posición en milímetros.

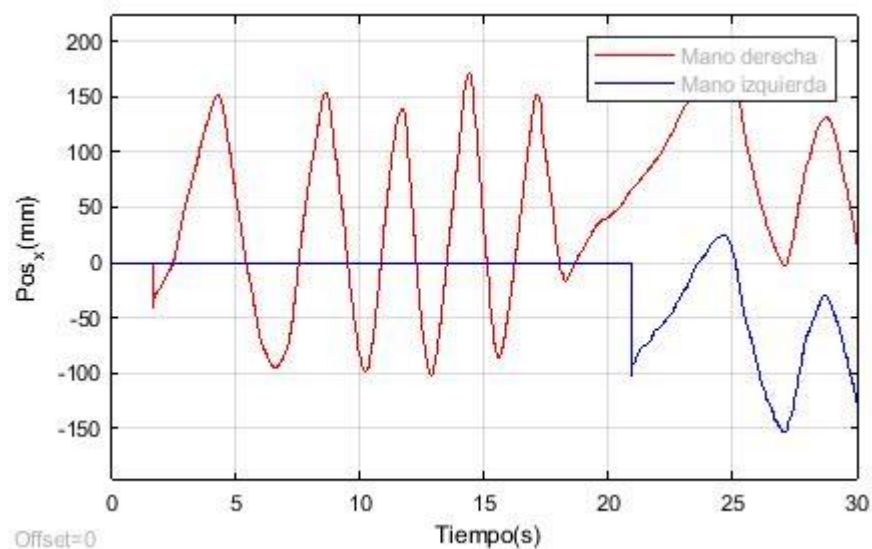


Figura 33: Posición manos eje X

Como se puede observar en la anterior imagen, en una primera instancia se movió la mano derecha continuamente a lo largo del eje de derecha a izquierda. Posteriormente sobre el segundo 21 se levantó la mano izquierda. Ambas manos realizaron el mismo movimiento pero desplazados varios milímetros en el eje X.

El siguiente ensayo realizado ha sido los movimientos de guiñada, cabeceo y alabeo realizados con las inclinaciones de la mano (Figura 32). La gráfica (Figura 34) muestra los ángulos en grados al producir el movimiento en un tiempo de registro de 30 segundos.

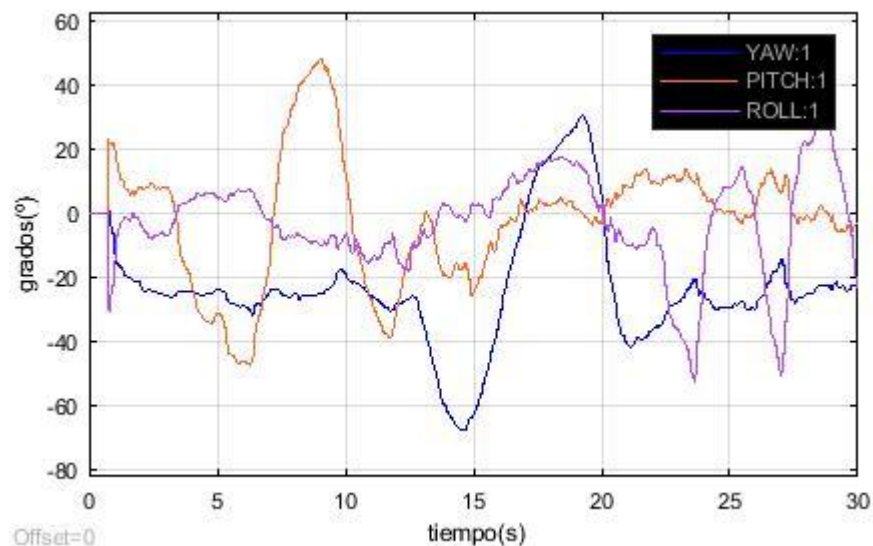


Figura 34: Ángulos de la mano

Por último, se muestra el ensayo de la altura de la mano en el eje Y respecto al Leap Motion (PALM_POS_Y). Esta variable será clave para controlar el PWM de los motores. En la siguiente imagen (Figura 35) se ve la altura en mm que se ha elevado la mano.

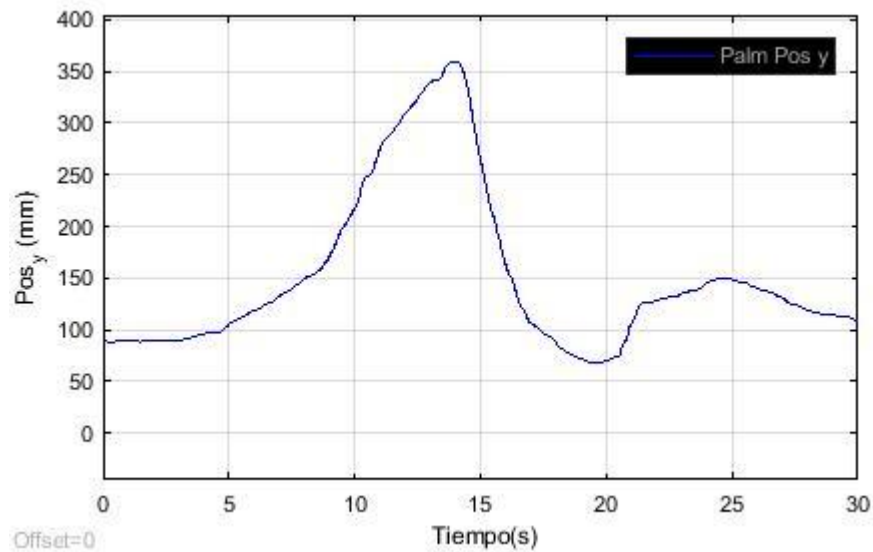


Figura 35: Altura mano

4.5 PROCESAMIENTO DE LAS SEÑALES

Después de recibir las señales de entrada llega una etapa de procesamiento de señales. Con la cantidad de variables de entrada que se presentan, se abren infinitas combinaciones de movimientos y gestos corporales para hacer la navegación del cuadricóptero. En consecuencia, se eligen las variables más convenientes y sencillas para conseguir los objetivos (Figura 36). Se ha decidido, para simplificar, no utilizar los inputs que corresponden a los dedos de las manos que son: *Position*, *Velocity* y *Direction* debido a la complejidad que conllevan. Es preferible utilizar las posiciones y ángulos de la palma de la mano.

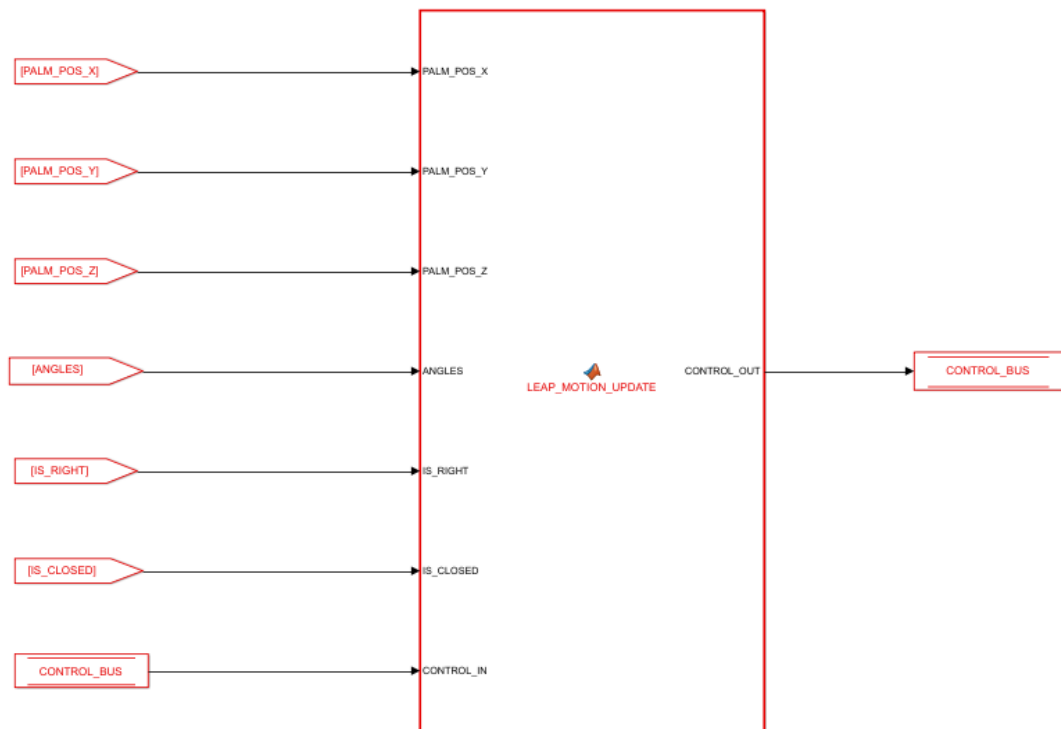


Figura 36: Leap Motion Update en Simulink

Para el armado y desarmado de los motores se han usado las siguientes variables:

- Armar motores: se ha utilizado la variable IS_RIGHT(2). Como se ha dicho anteriormente, esta variable da el valor de 1 cuando se levanta primero la mano izquierda y a continuación la derecha. Es una posición de seguridad para armar los motores sin tener problemas.
- Desarmar motores: bastará con cerrar el puño de la mano derecha. Se usa la variable IS_CLOSED para realizar esta orden.

Se han creado tres modos tres modos distintos de navegación por gestos de un cuadricóptero en función de las preferencias y del nivel de habilidad que tenga el usuario. Se ofrecen así tres alternativas para controlar el UAV: Control Básico por Posición, Control Básico por Ángulos, Control Avanzado por Posición. La

diferencia entre el modo Básico y el Avanzado es que el primero está basado en variables discretas y el segundo en continuas.

- Independientemente de control utilizado, se empleará la guiñada de la mano (YAW) para hacer orientar el cuadricóptero una vez que ya ha despegado (Figura 37). Esta variable tiene tres estados:
 - Giro a velocidad angular constante en dirección de las agujas del reloj (positivo)
 - Giro a velocidad angular constante en dirección opuesta de las agujas del reloj (negativo)
 - No hacer movimientos de rotación

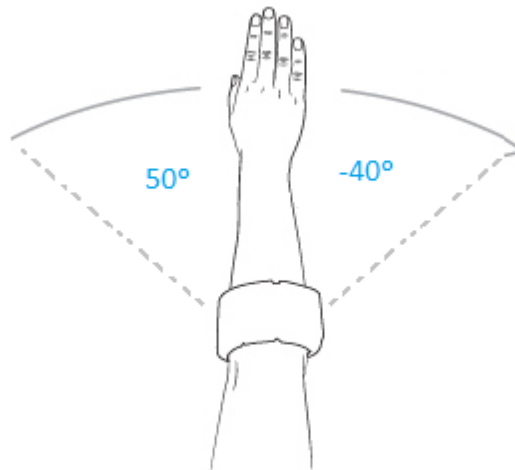


Figura 37: Yaw

4.5.1 CONTROL BÁSICO POR POSICIÓN

Como se ha mencionado anteriormente, este modo consiste en trabajar con variables discretas. Esto quiere decir que para este modo se emplean umbrales y bandas muertas. No interesa la medida exacta, sino si ese valor está dentro o fuera del umbral para luego dar la orden correspondiente. En ese espacio intermedio o banda muerta la orden que recibe el dron es la de conservar su posición actual. Es más sencilla si se uno mismo se imagina un cubo imaginario (Figura 38) en el que

si la mano derecha permanece dentro no ocurre nada, pero si esa misma mano se saca del cubo en una dirección de un eje, el dron recibe la orden de moverse se a velocidad contante en esa misma dirección ($V_{y\max}$).

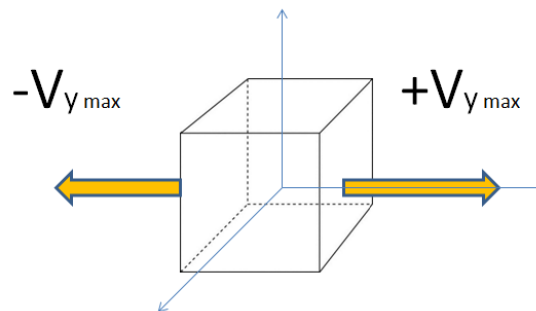


Figura 38: Cubo en el sistema de coordenadas

En todas las diferentes variables habrá hasta tres órdenes para cada eje en función de la posición de la mano: orden de velocidad constante hacia una dirección, hacia la dirección contraria o permanece en la posición actual. A continuación se muestra las variables utilizadas para cada movimiento del UAV:

- Pitch (Cabeceo): en función de la posición de la mano derecha en el eje Z (PALM_POS_Z), se le da la orden de avance (positivo) o retroceso (negativo).
- Throttle: para manejar la altura del cuadricóptero se ha empleado la medida de la posición de la palma de la mano en el eje Y (PALM_POS_Y). Es de gran utilidad para el despegue y aterrizaje del cuadricóptero.
- Roll (Alabeo): en función de la posición de la mano derecha en el eje X (PALM_POS_Z), se le da la orden girar a la derecha (positivo) o a la izquierda (negativo).



4.5.2 CONTROL BÁSICO POR ÁNGULOS

Como el control anterior, este también se basa en variables discretas. El planteamiento de este control de navegación se basa en la idea del cubo del anterior control pero sustituyendo la posición de la mano por la inclinación de esta en los distintos ejes. Para cada ángulo de la mano hay un valor que hace de frontera. Si los grados de inclinación de la mano sobrepasan esa frontera, el dron recibe la orden de moverse a velocidad constante sobre ese eje en esa dirección. Si no sobrepasa ese valor, el dron permanecerá en la misma posición.

- Pitch (Cabeceo): En función del ángulo del cabeceo de la mano derecha le daremos la orden de avance (negativo) o de retroceso (positivo) como se muestra en la siguiente ilustración (Figura 39).

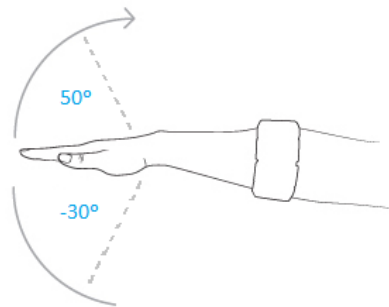


Figura 39: Pitch

- Roll (Alabeo): Para girar el cuadricóptero se empleará el alabeo de la mano derecha: giro a la derecha con ángulo positivo y a la izquierda con ángulo negativo (Figura 40).

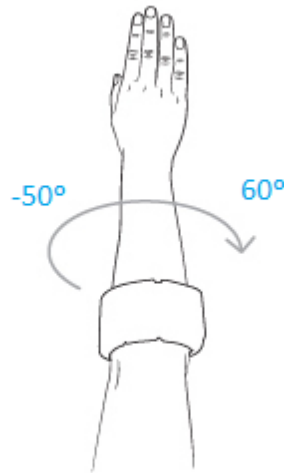


Figura 40: Roll

- Throttle: para manejar la altura del cuadricóptero se ha empleado la medida de la posición de la palma de la mano en el eje Y (PALM_POS_Y). El throttle se manejará con la altura de la mano y no con ángulos.

4.5.3 CONTROL AVANZADO POR POSICIÓN

En los casos anteriores se utiliza el Leap Motion para hacer movimientos con las manos para dar órdenes al dron. En este caso la idea principal de este control es que se basa en variables continuas y ahora sí que importa dónde este colocada la mano derecha. No se la una orden, sino una posición, que posteriormente será transformado en una velocidad concreta. En este caso, la velocidad del cuadricóptero es directamente proporcional al posicionamiento de la mano. Se parte de la colocación de la mano en un punto en el que la posición del dron permanece constante que es el (0, 200, 0) mm. La velocidad que adquiere el dron irá en aumento en función de la posición de la mano derecha respecto al punto de partida (Figura 41).

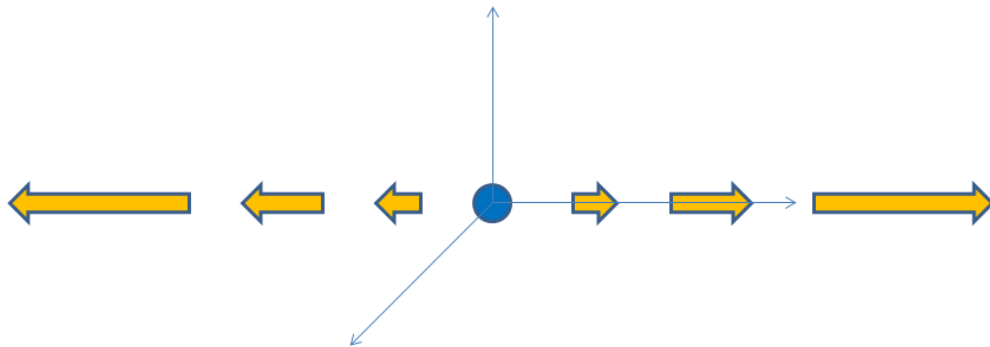


Figura 41: Control Avanzado

- Throttle: para manejar la altura del cuadricóptero se ha empleado la medida de la posición de la palma de la mano en el eje Y (PALM_POS_Y). para una altura de 200 mm el cuadricóptero permanecerá en la misma posición. Si es superior o inferior a ese valor, el cuadricóptero subirá o bajará respectivamente.
- Pitch (Cabeceo): en función de la posición de la mano derecha en el eje Z (PALM_POS_Z), se le da la orden de avance (positivo) o retroceso (negativo).
- Roll (Alabeo): en función de la posición de la mano derecha en el eje X (PALM_POS_Z), se le da la orden girar a la derecha (positivo) o a la izquierda (negativo), con distinta velocidad.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Leap Motion

Capítulo 5 SIMULACIÓN Y VISUALIZACIÓN

En este capítulo se muestran los resultados realizados en diferentes ensayos. Debido a la gran complejidad del proyecto, se irán estableciendo objetivos a corto plazo para conseguir depurar errores con mayor facilidad. En consecuencia, antes de realizar la navegación sobre el cuadricóptero en condiciones reales, se efectuarán pruebas sobre simuladores.

En un primer lugar se ha visto las gráficas del posicionamiento del dron en vuelo, luego en un simulador de vuelo en 3D llamado FlightGear.

5.1 VISUALIZACIÓN EN SIMULINK

En la siguiente imagen (Figura 42) se muestra la conexión de los distintos componentes hardware y software para realizar el control de navegación del dron.

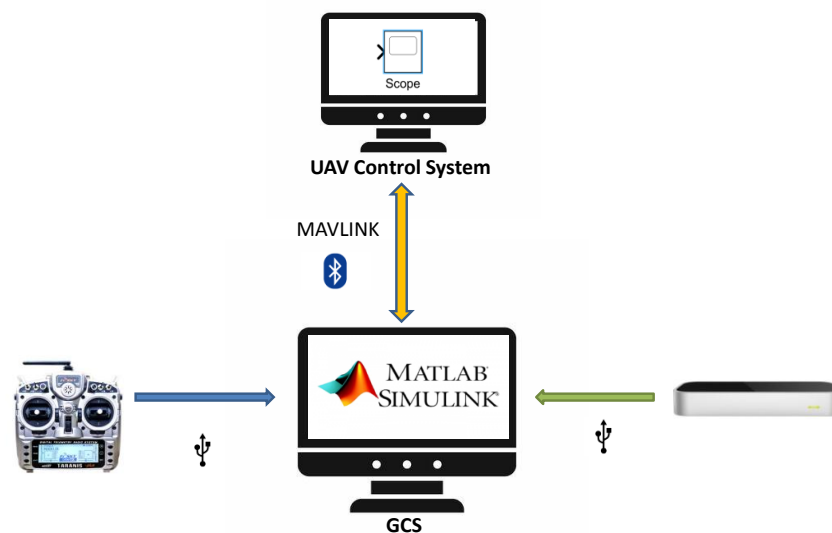


Figura 42: Esquema simulación dron

En el PC Control Station se realiza la implementación del Leap Motion. Luego con Bluetooth utilizando el protocolo de comunicación MAVLINK se envía las variables de las distintas variables necesarias para controlar el dron. El UAV Control System recibe la información de la estación base (GCS) y emula el funcionamiento de un cuadricóptero real. A través de unos *scopes* se pueden visualizar el PWM de los motores, los ángulos, la posición...

Como se ha dicho en el anterior capítulo, se ha decidido controlar el dron ajustando las referencias de velocidad para los tres modos de control de navegación.

El eje de coordenadas empleado para aviones o cuadricópteros es diferente al empleado con anterioridad para el Leap Motion (Figura 30). Para entender con mayor facilidad las gráficas que se mostrarán posteriormente, a continuación se muestra los ejes utilizados para las simulaciones del dron (Figura 43).

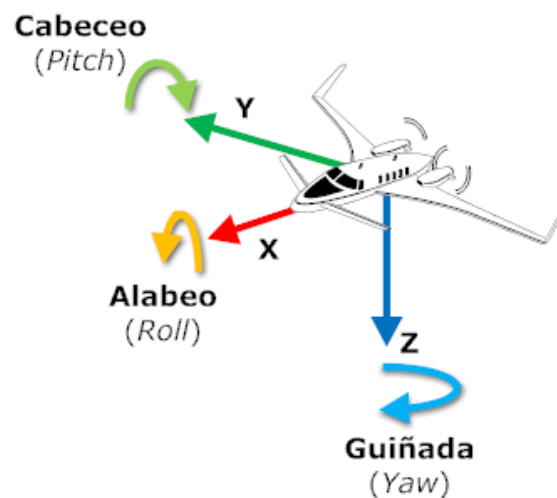


Figura 43: Ejes de coordenadas aviación

A continuación se mostrarán los ensayos realizados en los distintos modos de control creados para este proyecto. Para una mejor comprensión del lector se mostrará solamente el eje Z correspondiente a la altura en metros del UAV en las gráficas siguientes, en concreto de los despegues.



5.1.1 CONTROL BÁSICO POR POSICIÓN

A continuación se muestra una gráfica de la integral de la velocidad del UAV en metros en el eje Z (Figura 44). El tiempo de registro es aproximadamente de 50 segundos. La señal naranja corresponde a la Referencia y la azul al Output. En el segundo 5 se le ha dado la orden al dron de despegar hasta 1 metro de altura. A partir del segundo 15 comenzó el control de navegación por gestos.

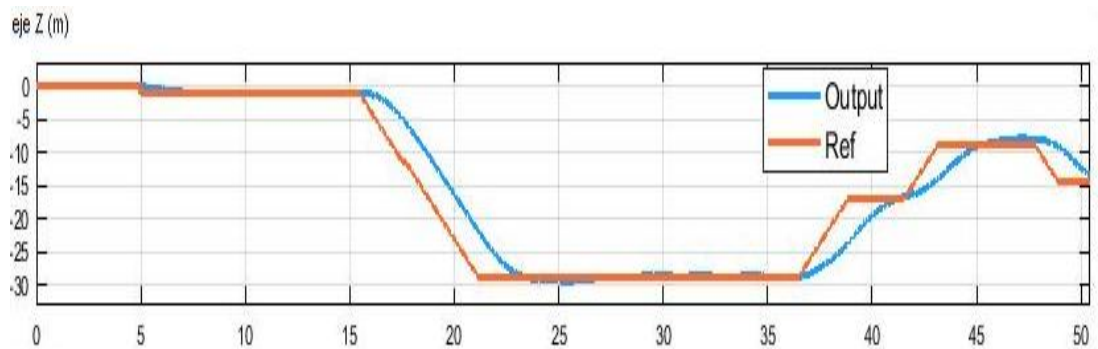


Figura 44: Posicionamiento UAV en Control Básico por Posición

Como se puede observar en la imagen, a partir del segundo 15 la mano derecha ha sido llevada fuera de la banda muerta y el cuadricóptero ha recibido la orden de moverse a una velocidad constante utilizando variables discretas. En el segundo 21 la mano ha vuelto a la banda muerta, por lo que el UAV recibe la orden de no moverse.

5.1.2 CONTROL BÁSICO POR ÁNGULOS

El Control Básico por Ángulos no difiere en nada en simulación Control Básico por Posición.



En la Figura 45 se ha utilizado un tiempo de registro de 60 segundos. La señal naranja corresponde a la Referencia y la azul a la del Output. Las rampas de velocidad se le dan al UAV son exactamente las mismas que el anterior control.

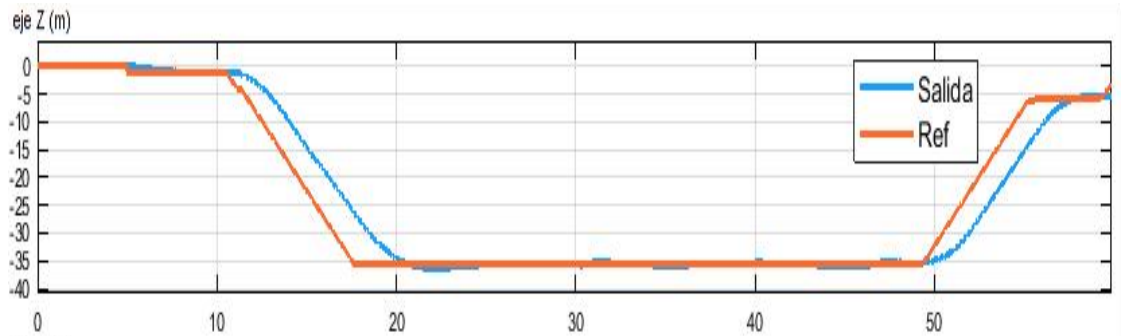


Figura 45: Posicionamiento UAV en Control Básico por Ángulos

5.1.3 CONTROL AVANZADO POR POSICIÓN

En los casos anteriores se utiliza el Leap Motion para hacer movimientos con las manos para dar órdenes al UAV con pendientes constantes. La idea es que para cada posición concreta de la mano está vinculada una referencia de velocidad específica se puede entender sencillamente comparando esta gráfica con las anteriores.

Para esta ilustración (Figura 46) se ha utilizado un tiempo de registro de 90 segundos. La señal naranja corresponde a la Referencia y la azul a la del Output.

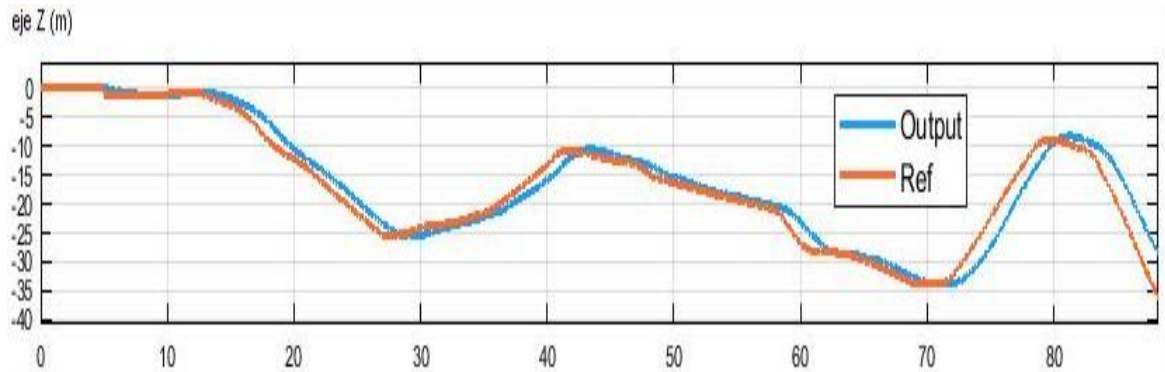


Figura 46: Posicionamiento UAV en Control Avanzado por Posición

En la imagen se puede ver que son valores continuos y no son discretos como en los dos modos anteriores.

5.2 FLIGHTGEAR

El FlightGear es un simulador de vuelo multiplataforma y libre. Es un proyecto de simulador de vuelo gratuito desarrollado por voluntarios de todo el mundo pensado para usos principalmente académicos. La ventaja del FlightGear frente a otros simuladores de vuelo comerciales es su extensibilidad y flexibilidad. Su código es libre y no esconde su funcionamiento interno. Este simulador nació a raíz de las incompatibilidades de otros simuladores de vuelo comerciales que no podían ser modificados como deseara el usuario, debido a su privacidad. Se pone a la alcance de los usuarios la capacidad de aprender, investigar y contribuir en la creación y extensión de este simulador. Les da la oportunidad de plasmar sus ideas a diferencia de otras plataformas.

Utiliza la interfaz OpenGL ES que define una API multilenguaje y multiplataforma para reproducir gráficos en 3D y 2D. Cabe mencionar del FlightGear que, a pesar de su bajo nivel gráfico comparado con otros simuladores, destaca por su realismo en la navegación y control de vehículos voladores y



también por su gran jugabilidad. Ha sido desarrollado con un alto perfil técnico y científico.

Su amplia variedad de naves, permite escoger un cuadricóptero (Figura 47):

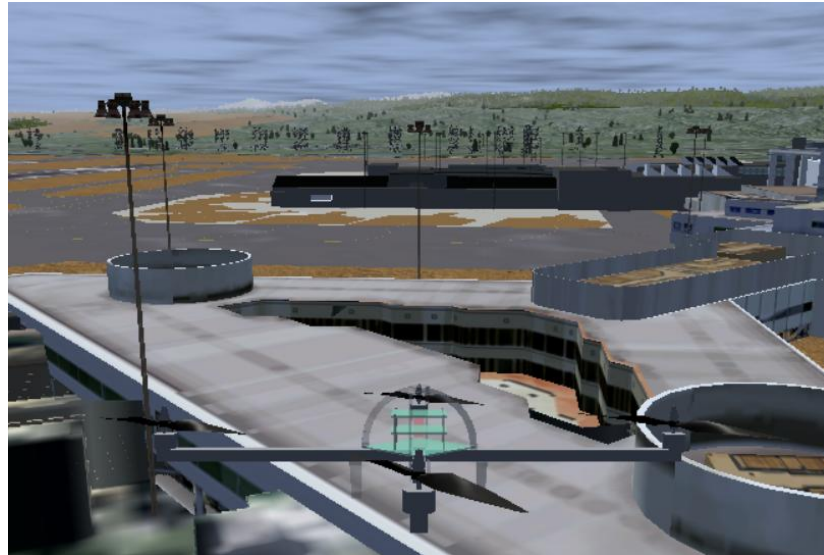


Figura 47: Arducopter en FlightGear

En este proyecto resulta una herramienta fundamentalmente por las siguientes ventajas:

- Compatibilidad con Matlab y Simulink: permite configurar la dinámica del cuadricóptero y sus respectivos cálculos en un modelo de Simulink y utilizar simultáneamente el Software del modelo y el Software del simulador de vuelo como interfaz gráfica del comportamiento del dron. Se reproducirá con plena exactitud el funcionamiento de un cuadricóptero real (Figura 48).

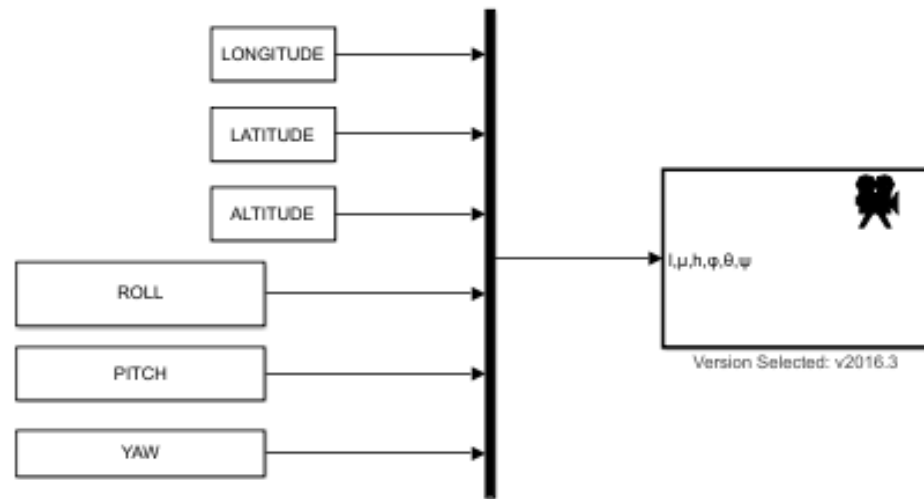


Figura 48: Bloque FlightGear en Simulink

- Interfaz gráfica: la versatilidad del programa para enviar los datos de la dinámica del avión no tripulado, en concreto las coordenadas GPS, la altura y los ángulos de Euler.

El gran inconveniente del Flightgear es la cantidad de recursos informáticos que consume. Con un equipo potente, es posible ejecutar los cálculos y la interfaz al mismo tiempo y sin retardos. En cambio, si no se dispone de equipos potentes, se pueden conectar dos equipos en red, de tal forma que uno de los equipos ejecuta Matlab y el Simulink, y éste envía los resultados calculados por protocolo MAVLINK en UDP o UART al otro equipo. Este equipo, ejecutaría el simulador de vuelo, recibiendo los cálculos y mostrando los movimientos del cuadricóptero.

En la siguiente imagen (Figura 49) se muestra la conexión de los distintos componentes hardware y software para ejecutar la simulación en FlightGear. La simulación del FlightGear se realiza en el mismo ordenador que el PC Control Station. Mediante el protocolo MAVLINK se envía a la estación base los registros de los ángulos y el posicionamiento del cuadricóptero para poder visualizarlo. En la comunicación el retardo es tan pequeño que no afecta al control.

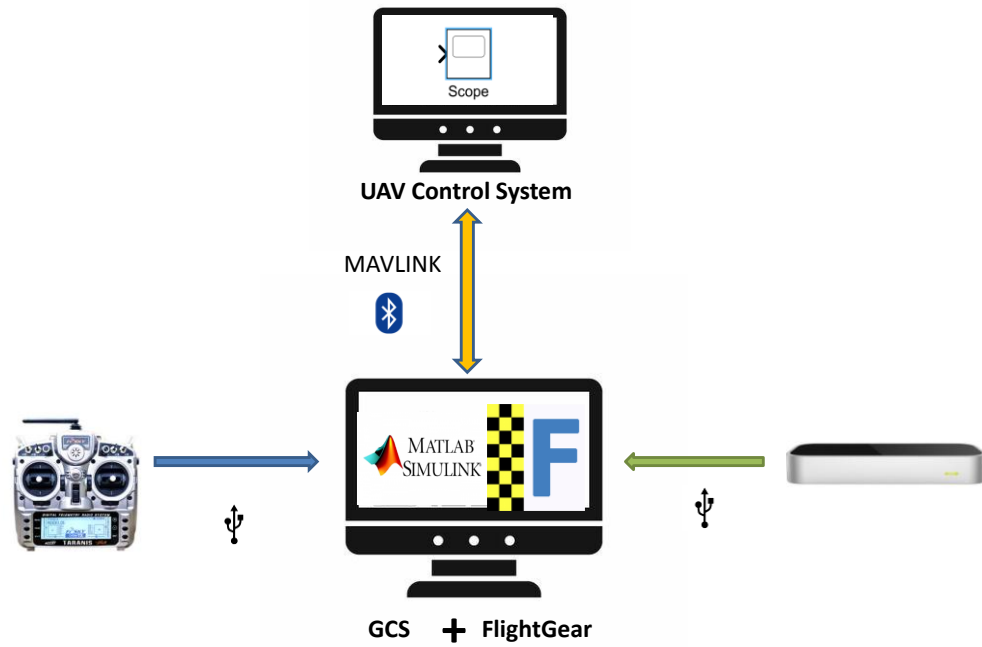


Figura 49: Esquema conexión FlightGear



Capítulo 6 RESUMEN Y CONCLUSIONES

En este capítulo se resumen las tareas realizadas y se incluye la conclusión del proyecto

6.1 RESUMEN TAREAS REALIZADAS

En un primer momento se ha tenido que estudiar el funcionamiento del Leap Motion para poder implementarlo en Matlab y Simulink y procesar correctamente los datos captados. Luego se procedió a su integración con la Estación Base (GCS).

Después se procedió al control del cuadricóptero y cómo conseguir la sustitución total del Leap Motion con la emisora RC. Para emular el funcionamiento de un UAV, se precisó de otro ordenador al cual se le enviarían los registros necesarios para su manejo a través de MAVLINK, que es un protocolo de comunicación muy extendido en el sector aeronáutico. Ya establecido la conexión UART por Bluetooth, se realizaron ensayos y pruebas en simulación. El resultado fue visualizado a través del simulador de vuelo en 3D llamado FlightGear.

Para ofrecer mayores alternativas al usuario y mayores comodidades se realizaron tres modos distintos de manejar la navegación del UAV: Controles Básicos (ángulos y posición) y Avanzado (posición).

Una vez comprobado que la interfaz intuitiva creada en este proyecto funcionaba correctamente, se procedió a realizar pruebas en condiciones reales. Para ello se construyó de cero un cuadricóptero pequeño y sencillo de manejar para reducir en la mayor parte de lo posible las pérdidas por choques fortuitos.



6.2 CONCLUSIONES

A la vista de los resultados, se puede afirmar que se ha conseguido con gran éxito el manejo del cuadricóptero de manera más intuitiva que la emisora RC. Su implementación en Matlab/Simulink y la integración en el simulador de vuelo en 3D FlightGear avalan el proyecto. El Leap Motion permite manejar un UAV sin tener experiencia previa. Se puede considerar un logro, pero siempre pensando que es posible realizar mejoras y nuevas aportaciones.

En lo que se refiere al UAV, se ha construido y configurado correctamente para su vuelo. Su pequeño tamaño permite que los posibles golpes no sean realmente un problema. Sin embargo, por falta de tiempo, no se ha llegado a realizar la navegación del cuadricóptero, ya que no estaba diseñado el control de estabilización y navegación, siendo este apartado responsabilidad de otro proyecto. Por este motivo el dron está habilitado para futuros desarrollos del laboratorio para el curso venidero.



Capítulo 7 FUTUROS DESARROLLOS

Existen posibles desarrollos futuros para ampliar este proyecto debido a la extensión y a la falta de tiempo. En este proyecto se ha creado una interfaz intuitiva para el usuario medio para el control de navegación de un cuadricóptero mediante gestos corporales.

A partir de este proyecto se pueden sacar tres líneas diferentes de mejora.

La primera sería una ampliación de este proyecto para conseguir hacerlo lo más realista posible:

- A través de las bases de este proyecto se puede cambiar los controles a los gestos de las manos que más prefiera el usuario. En este trabajo se han creado tres modos de control de navegación, pero caben infinidad de combinaciones para realizar la navegación. Un ejemplo sería modificar el eje de coordenadas del Leap Motion por el eje que captara inicialmente de la posición de la mano.
- Estudiar y aplicar ecuaciones cinemáticas corporales para que los gestos se imiten de forma más intuitiva.

La segunda trata entorno al Leap Motion con otras finalidades distintas al control de un UAV:

- Ya se ha demostrado que se puede controlar con los gestos un UAV. Sin embargo, hay cantidad de aplicaciones que se puede controlar con el Leap Motion: control de brazos robóticos, de la luminosidad de un habitáculo, aplicaciones domóticas...
- Hacer una aproximación a la realidad virtual empleando el Leap Motion con el Oculus Rift. En la realidad virtual se pueden crear innumerables escenarios en los que también tienen cabida los UAV.



Por último, gracias a las bases asentadas en este proyecto, tales como la interfaz genérica, es posible desarrollar otros proyectos relacionados con el control de navegación de UAV con gestos corporales pero con otros sensores, como el Myo Armband, del que se habló en el capítulo 2. El Myo Armband está muy bien valorado por sus usuarios y es muy accesible en el mercado. Un posible estudio sería la investigación de este sensor y ver si su resolución y las señales captadas son superiores o inferiores que el Leap Motion. El brazalete es compatible con el microcontrolador Raspberry Pi que ha sido utilizado para este proyecto.



BIBLIOGRAFÍA

- [1] El origen y la historia de los drones (2016).
- [2] Obtenido de: <https://hemav.com/el-origen-y-la-historia-de-los-drones/>
- [3] Drones que se manejan como un 'jedi'. La Voz de Galicia. César Rodríguez (Enero 2018).
- [4] Mueve el esqueleto; Move, Kinect y Wii a examen (2012)
- [5] Obtenido de: <http://www.fsgamer.com/mueve-el-esqueleto-move-kinect-y-wii-a-examen-20120704.html>
- [6] Leap motion app store | leap motion. Obtenido de:
- [7] Apps.leapmotion.com. Available at: <https://apps.leapmotion.com/>
- [8] Menéndez botella, Marta. Proyecto fin de grado: Control de navegación mediante gestos. Escuela Técnica Superior de Ingeniería-Icai.
- [9] Microsoft's project natal: what does it mean for game industry? (Junio 2009) Obtenido de: <https://www.cnet.com/news/microsofts-project-natal-what-does-it-mean-for-game-industry/>
- [10] Orland, Kyle (Febrero 2011). "News - Microsoft Announces Windows Kinect SDK For Spring Release".
- [11] Edwin Duque Blog. Diferencias entre Kinect V1 y Kinect V2 (Febrero 2015). Obtenido de: <https://edwinnui.wordpress.com/2015/02/05/diferencias-entre-kinect-v1-y-kinect-v2-2/>
- [12] Matthew Panzarino. Leap Motion launches software developer program and starts sending test units of its 3d controller (Octubre 2012). Obtenido de: <https://thenextweb.com/apple/2012/10/29/leap-motion-launches-software-developer-program-and-starts-sending-test-units-of-its-3d-controller/>
- [13] Lyra VR puts music synthesis within virtual reach. *Www.gizmag.com*. Retrieved 2016-04-18.
- [14] "leap motion announces orion for faster, more accurate vr hand tracking vrfocus". *Vrfocus* (Abril 2016).
- [15] Electromiografía y electroneurografía. Diagnóstico en la clínica. Universidad de navarra.
- [16] Myo Armband. Obtenido de: www.myo.com
- [17] MIT Technology Review. Un brazalete para simplificar el control por gestos (Julio 2013). Obtenido de: <https://www.technologyreview.es/s/3685/un-brazalete-para-simplificar-el-control-por-gestos>



- [18] Flying drones with myopilot <http://developerblog.myo.com/flying-drones-with-myopilot/>
- [19] MAVLINK Common Message Set Specifications
- [20] RC Receiver. Sbus Protocol.
- [21] The FlightGear Manual. Michael Basler. Getting Start (Enero 2017)



Parte II PRESUPUESTOS



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Bibliografía



Capítulo 1 RECURSOS EMPLEADOS

En esta sección se ha calculado la cantidad de recursos que se han dedicado al proyecto, tanto elementos materiales como de equipo, software o humanos.

1.1 COMPONENTES PRINCIPALES

Componentes	Cantidad
Leap Motion 2.0	1
Emisora RC	1
Transmisor Bluetooth	1
Receptor Bluetooth	1
Estructura de cuadricóptero con PCB 1	1
Motores Dragonfly	4
Batería LiPo	1
Hélices 5030 10	4
Tarjeta de control HKPilotMega 2.7 1	1
Raspberry Pi	1

Tabla 3: Recursos Empleados de Componentes Principales



1.2 HERRAMIENTAS Y SOFTWARE

Elemento/programa	Cantidad	Horas de uso en el proyecto	Horas de uso al año
Ordenador	1	450	1500
Cargador de baterías	1	10	20
Cable USB-microUSB	1	5	100
Matlab/Simulink 2018a	1	320	1300
Microsoft Office	1	150	1000
FlightGear	1	50	150
Osciloscopio	1	5	100

Tabla 4: Recursos Empleados de Herramientas y Software

1.3 MANO DE OBRA

Actividad	Horas
Búsqueda de información	40
Análisis y diseño del control	50
Implementación y adaptación de los sensores al control	200
Ensayos, pruebas y depuración	200
Redacción de documentos	80

Tabla 5: Recursos Empleados de Mano de obra



Capítulo 2 COSTES UNITARIOS

En este apartado se detallan los costes y precios de cada uno de los componentes que se han analizado previamente para llevar a cabo este proyecto.

2.1 COMPONENTES PRINCIPALES

Componentes	Coste (€/ud)
Leap Motion 2.0	75
Emisora RC	150
Transmisor Bluetooth	8
Receptor Bluetooth	8
Estructura de cuadricóptero con PCB 1	15
Motores Dragonfly	4
Batería LiPo	20
Hélices 5030 10	0,75
Tarjeta de control HKPilotMega 2.7 1	64
Raspberry Pi	50

Tabla 6: Costes Unitarios de Componentes Principales



2.2 *HERRAMIENTAS Y SOFTWARE*

Elemento/programa	Coste (€/ud)
Ordenador	1000
Cargador de baterías	40
Cable USB-microUSB	2
Matlab/Simulink 2018a	6000
Microsoft Office	100
FlightGear	0
Osciloscopio	1500

Tabla 7: Costes Unitarios de Herramientas y Software

2.3 *MANO DE OBRA*

Actividad	Coste (€/h)
Búsqueda de información	10
Análisis y diseño del control	40
Implementación y adaptación de los sensores al control	60
Ensayos, pruebas y depuración	50
Redacción de documentos	20

Tabla 8: Costes Unitarios Mano de obra



Capítulo 3 SUMAS PARCIALES

En este apartado se ha calculado a partir de las mediciones y los precios unitarios, el coste total de cada uno de los recursos empleados.

3.1 COMPONENTES PRINCIPALES

Componentes	Cantidad	Coste (€/ud)	Coste total (€)
Leap Motion 2.0	1	75	75
Emisora RC	1	150	150
Transmisor Bluetooth	1	8	8
Receptor Bluetooth	1	8	8
Estructura de cuadricóptero con PCB 1	1	15	15
Motores Dragonfly	4	4	16
Batería LiPo	1	20	20
Hélices 5030 10	4	0,75	3
Tarjeta de control HKPilotMega 2.7 1	1	64	54
Raspberry Pi	1	50	50
		TOTAL	399

Tabla 9: Sumas Parciales de Componentes Principales



3.2 HERRAMIENTAS Y SOFTWARE

Elemento/programa	Cantidad	Horas de uso en el proyecto	Horas de uso al año	Amortización anual	Coste (€/ud)	Coste total (€)
Ordenador	1	450	1500	25%	1000	75
Cargador de baterías	1	10	20	25%	40	5
Cable USB-microUSB	1	5	100	15%	2	0,02
Matlab/Simulink 2018a	1	320	1300	15%	6000	221,6
Microsoft Office	1	150	1000	15%	100	2,25
FlightGear	1	50	150	15%	0	0
Osciloscopio	1	5	100	25%	1500	18,75
					TOTAL	322,62

Tabla 10: Sumas Parciales de Herramientas y Software

3.3 MANO DE OBRA

Actividad	Horas	Precio (€/h)	Precio total (€)
Búsqueda de información	40	10	400
Análisis y diseño del control	50	40	2000
Implementación y adaptación de los sensores al control	200	60	12000
Ensayos, pruebas y depuración	200	50	10000
Redacción de documentos	80	20	1600
TOTAL			26000

Tabla 11: Sumas Parciales de Mano de obra



Capítulo 4 PRESUPUESTO GENERAL

En este apartado se muestra el presupuesto total de este proyecto:

Recurso	Coste (€)
Componentes	399
Herramientas y software	322,62
Mano de obra	26000
TOTAL	26721,62

Tabla 12: Presupuesto General del proyecto



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Presupuesto General



Parte III CÓDIGO FUENTE



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Código fuente 1



CÓDIGO FUENTE 1

En este anexo se recoge el código y procesamiento de datos del bloque de la matlab-function LEAP_MOTION_UPDATE. En este archivo se muestra la diferencia en el procesamiento de datos para los distintos controles de navegación creados en este proyecto:

```
function CONTROL_OUT =  
LEAP_MOTION_UPDATE(PALM_POS_X,PALM_POS_Y,PALM_POS_Z,ANGLES,IS_RIGH  
T,IS_CLOSED,CONTROL_IN)  
CONTROL_OUT = CONTROL_IN;  
  
persistent initialize armed  
  
if isempty(initialize)  
    initialize = 1;  
    armed = 0;  
end  
  
%THROTTLE  
  
if PALM_POS_Y(1) >= 300  
    Palmposition_y = 0.9  
elseif PALM_POS_Y(1) <= 100  
    Palmposition_y = 0.04  
else  
    Palmposition_y = 0.5  
end  
  
if IS_CLOSED(1) >= 0.65  
    is_closed = 1  
  
else  
    is_closed = 0  
end  
  
if (is_closed == 0 && IS_RIGHT(2)==1)  
    armed = 1;  
elseif (is_closed == 1 && IS_RIGHT(2) == 0)  
    armed = 0;  
end  
  
%YAW-GUIÑADA
```



```
    if ANGLES(1) >= 30
        yaw = 1;
    elseif ANGLES(1) <= -30
        yaw = -1;
    else
        yaw = 0;
    end

switch CONTROL_OUT.INPUT.LEAP_MOTION.CONTROL_MODE % MODE SELECTION

case 0 %Control básico por posición

    %THROTTLE
    if PALM_POS_Y(1) >= 300
        Palmposition_y = 0.9
    elseif PALM_POS_Y(1) <= 100
        Palmposition_y = 0.04
    else
        Palmposition_y = 0.5
    end

    %ROLL-ALABEO
    if PALM_POS_X(1) >= 60
        roll = 1;
    elseif PALM_POS_X(1) <= -60
        roll = -1;
    else
        roll = 0;
    end

    %PITCH-CABECEO
    if PALM_POS_Z(1) >= 60
        pitch = 1;
    elseif PALM_POS_Z(1) <= -50
        pitch = -1;
    else
        pitch = 0;
    end

case 1 %Control básico por ángulos

    %THROTTLE
    if PALM_POS_Y(1) >= 300
        Palmposition_y = 0.9
    elseif PALM_POS_Y(1) <= 100
        Palmposition_y = 0.04
    else
        Palmposition_y = 0.5
    end

    %PITCH
    if ANGLES(3) >= 20
```




```
        pitch = 1;
elseif ANGLES(3) <= -10
    pitch = -1;
else
    pitch = 0;
end

%ROLL
if ANGLES(5) >= 25
    roll = -1;
elseif ANGLES(5) <= -25
    roll = 1;
else
    roll = 0;
end

case 2 %Control avanzado por posición

    pitch = PALM_POS_Z(1);
    roll = PALM_POS_X(1);
    Palmposition_y = PALM_POS_Y(1);

otherwise
    pitch = 0;
    roll = 0;
    Palmposition_y = 0.04;
end

LEAP_MOTION = CONTROL_OUT.INPUT.LEAP_MOTION;
LEAP_MOTION.YAW_RATE_TARGET_PILOT = single(yaw);
LEAP_MOTION.PITCH_TARGET_PILOT = single(pitch);
LEAP_MOTION.ROLL_TARGET_PILOT = single(roll);
LEAP_MOTION.THR_TARGET_PILOT = single(Palmposition_y);
LEAP_MOTION.ARMED_MOTORS = single(armed);
CONTROL_OUT.INPUT.LEAP_MOTION=LEAP_MOTION;
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Código fuente 1



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL
