



## **INGENIERÍA TÉCNICA INFORMÁTICA EN SISTEMAS**

Curso Académico 2011/2017

Proyecto Fin de Carrera

## **ROBÓTICA AÉREA CON AVIONES**

Autor : José Antonio Fernández Casillas

Tutor : José María Cañas Plaza



# **Proyecto Fin de Carrera**

Robótica aérea con aviones

**Autor :** José Antonio Fernández Casillas **Tutor :** José María Cañas Plaza

La defensa del presente Proyecto Fin de Carrera se realizó el día        de  
de 20XX, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a        de        de 20XX



*Dedicado a  
mi familia / mi abuelo / mi abuela*



# **Agradecimientos**

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.



# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# **Summary**

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Índice general

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>1</b>  |
| 1.1. Robótica Aérea . . . . .  | 1         |
| 1.2. Tipos de Aeronaves . . . . .                                      | 2         |
| 1.3. Aplicaciones . . . . .  | 4         |
| 1.4. Robótica aérea en el laboratorio de Robótica de la URJC . . . . . | 7         |
| <b>2. Objetivos</b>  | <b>11</b> |
| <b>3. Infraestructura utilizada</b>                                    | <b>13</b> |
| 3.1. Hardware . . . . .  | 13        |
| 3.2. Placa estabilizadora . . . . .                                    | 14        |
| 3.3. JdeRobot . . . . .  | 15        |
| 3.3.1. Interfaces . . . . .  | 16        |
| 3.4. MAVLink . . . . .   | 19        |
| 3.5. Python y PyQt5 . . . . .  | 20        |
| 3.6. Mapas y Geo-referenciación . . . . .                              | 21        |
| 3.7. SITL . . . . .  | 21        |
| <b>4. Infraestructura Desarrollada</b>                                 | <b>23</b> |
| 4.1. Sección 1 . . . . .   | 23        |
| <b>5. AUV Commander</b>  | <b>25</b> |
| 5.1. Arquitectura general . . . . .                                    | 25        |
| <b>6. Experimentos</b>   | <b>27</b> |

|   |           |
|---|-----------|
| <b>7. Conclusiones</b>                    | <b>29</b> |
| 7.1. Consecución de objetivos . . . . .   | 29        |
| 7.2. Aplicación de lo aprendido . . . . . | 29        |
| 7.3. Lecciones aprendidas . . . . .       | 29        |
| 7.4. Trabajos futuros . . . . .           | 30        |
| 7.5. Valoración personal . . . . .        | 30        |
| <b>A. Manual de usuario</b>               | <b>31</b> |
| <b>Bibliografía</b>                       | <b>33</b> |

# Índice de figuras

|      |   |    |
|------|---|----|
| 1.1. | Dron grabando en el interior de un crater . . . . . | 5  |
| 1.2. | Dron Yamaha RMAX fumigando . . . . .                | 5  |
| 1.3. | Ventanas de UAV Viewer . . . . .                    | 8  |
| 3.1. | Placa estabilizadora ardupilot . . . . .            | 14 |
| 3.2. | Arquitectura de SITL . . . . .                      | 22 |
| 5.1. | Estructura del parser básico . . . . .              | 26 |



# **Capítulo 1**

## **Introducción**

Los UAV o Drones se han popularizado en los últimos años hasta es punto de formar parte de nuestro día a día con aplicaciones en muchos ámbitos de nuestra vida.

Si bien se están utilizando ya de forma habitual en sectores como el cine o la ingeniería civil, aún se están explorando muchas de las posibles utilidades que estos robots pueden llegar a ofrecer.

El objetivo de este trabajo final es poner en valor y asentar el uso de un tipo de UAV que no está hoy muy representado en el ámbito civil y que aventaja en varios aspectos al más popularizado quadracóptero, se trata del avión.

### **1.1. Robótica Aérea**

Los orígenes de la robótica aerea tienen origen militar y su avance ha estado intrínsecamente ligado a este ámbito durante todo el siglo XX.

Se consideran el origen de los aviones no tripulados los experimentos llevados a cabo a principios del siglo XX durante la 1<sup>a</sup> guerra mundial como el 'Aerial Target' desarrollado por el capitán A. H. Lowpara para su uso como blanco aéreo. Si bien eran vehículos no tripulados (Unmanned Aereal Vehicles) no eran autónomos y eran manejados desde tierra a través de una radio. No es hasta el final del siglo XX cuando bajo el escenario de la guerra de Vietnam y ante la creciente perdida de vidas de los pilotos estos vehículos vuelvan de nuevo a ser objeto de desarrollo y se conviertan en vehículos autónomos.

Desde ese momento y hasta nuestros días se utilizan de forma habitual en el ámbito militar

en misiones de reconocimiento, bombardeos o apoyo sin arriesgar vidas humanas.

A lo largo de los primeros años de este siglo debido al abaratamiento de los componentes electrónicos y a su miniaturización y potencia, la robótica aérea se ha 'desmilitarizado' y esta experimentado un enorme crecimiento en el ámbito de las aplicaciones civiles.

Hoy en día es común encontrar en cualquier juguetería quadracópteros radio-pilotados por poco menos de 30 euros y en tiendas especializadas podemos encontrarlos ya con el hardware y software integrados que les permiten seguir una serie de puntos de control y comportarse de forma autónoma por poco más de 200 €.

En la actualidad el uso de AUV o drones se ha popularizado tanto que es una de las industrias en las que más ha crecido su inversión, y es que según la empresa analista especializada en drones Droneii con sede en Hamburgo en un estudio sobre la inversión en el sector<sup>1</sup> en europa se invirtió en proyectos domésticos en 2016 cerca de 65 millones de dólares incrementándose esta cifra hasta los 314 millones si atendemos al mercado norteamericano. Estas datos se asientan en un mercado cada vez más extendido y con una gran proyección de crecimiento, la publicación BI Intelligence<sup>2</sup> espera que las ventas de drones alcancen los 12.000 millones en 2021.

Con su uso ya ampliamente extendido en sectores como el cine, la televisión, fotografía, agropecuario, forestal, ingeniería civil y presencia en sectores como el de salvamento o seguridad y protección el nicho de mercado de los UAV esta lejos de su cima y se investigan dia a dia nuevos usos en sectores como el de la logística o las telecomunicaciones como veremos mas adelante..

## 1.2. Tipos de Aeronaves

Las aeronaves son la base sobre las que se asienta la inteligencia que permite que nuestro robot vuele de ahí que convenga dedicar unas líneas a entender la base de las mismas y en particular las que son objeto de estudio y desarrollo en este PFC los llamados aerodinos.

Los aerodinos son aquellas aeronaves que vuelan a pesar de pesar más que el aire, son capaces de generar sustentación por sus propios medios a diferencia de los aerostatos como por ejemplo los globos aerostáticos.

---

<sup>1</sup><http://www.droneii.com/drone-investment-trends-2016>

<sup>2</sup><http://www.businessinsider.com/the-drones-report-research-use-cases-regulations-and->

Existen principalmente 2 tipos de aerodinos si atendemos al modo en que generan su sustentación con sus alas, de ala fija y las de ala rotatoria.

Dentro de la tipificación de ala fija tenemos aquellas aerodinos que tienen sus alas fijas al fuselaje, y que comúnmente conocemos como aviones, según la OACI, un avión es un «Aerodino propulsado por motor, que debe su sustentación en vuelo principalmente a reacciones aerodinámicas ejercidas sobre superficies que permanecen fijas en determinadas condiciones de vuelo». Algunos ejemplos de aerodinos de ala fija son los aeroplanos, planeadores/veleros, aladeltas, parapentes, paramotores y ultraligeros.

Este tipo de aerodinos tienen como principal ventaja que la carga de aire que necesitan en sus alas puede ser producida de muchas formas distintas (los veleros no tienen ningún tipo de propulsión). Esta carga es variable en función de la superficie alar del mismo y permite por tanto cargas más grandes que si instalásemos el mismo propulsor en un ala rotatoria. Pongamos como ejemplo el A380 de Airbus, es el avión de pasajeros más grande del mundo y cuenta con 4 motores que producen un empuje de entre 70.000 y 80.000 lbs, unas 32-36 toneladas de empuje cada uno generando por tanto entre los 4 a máximo rendimiento y optimas condiciones alrededor de 144 toneladas de empuje. Este avión tiene un peso máximo al despegue<sup>3</sup> de entre 560 y 590 toneladas. Tenemos por tanto que necesitamos en este caso  $\frac{1}{4}$  del peso total en empuje para despegar este avión. Si hiciésemos este mismo ejercicio con un aerodino de ala rotatoria como el Boing AH-64 o Apache con un peso máximo al despegue de 9,5 toneladas necesitaríamos que la combinación que realizan empuje y palas superase esos 9,5 toneladas para siguiendo levantar del suelo. Este tipo de aerodinos son por tanto más eficientes, rápidos, con mayor carga de pago, mayor alcance debido a su menor consumo y más estables.

Dentro de la tipificación de ala rotatoria tenemos aquellas aerodinos que producen su sustentación con el movimiento (rotación) de sus alas. En este tipo de aerodinos las alas, también llamadas 'palas' en este tipo de aerodinos, giran en torno a un eje produciendo con este giro la sustentación necesaria para despegar del suelo. Algunos ejemplos de este tipo de aerodinos son los helicópteros, autogiros, convertibles o los ampliamente conocidos en robótica aérea los quadracópteros. Este tipo de aerodino tiene como principal ventaja frente a los ala fija en su versatilidad a la hora de realizar las maniobras de despegue y aterrizaje que pueden realizarse

---

<sup>3</sup>Peso máximo que es capaz de soportar un avión en su maniobra de despegue

de forma vertical (VTOL<sup>4</sup>) además de la capacidad de realizar vuelo estacionario<sup>5</sup> que le hacen imprescindible en escenarios poco accesibles o donde nos es posible aterrizar como el rescate marítimo.

### 1.3. Aplicaciones

La robótica aérea ha experimentado un crecimiento exponencial en los últimos años, se ha popularizado su uso y se ha extendido la comercialización de drones.

El sector donde más rápida acogida ha tenido la robótica aérea ha sido el sector audiovisual, se usa de forma habitual en cine, grabación de espectáculos en directo televisión y fotografía. El porqué de tal acogida se basa principalmente en 2 factores, los costes y la viabilidad técnica. Los costes de realizar una toma aérea en una producción antes pasaban por el alquiler de un helicóptero, dependiendo de la toma, así como del material y la contratación de los medios humanos a bordo para realizarlas podía ascender a entre 4000 y 6000 euros la hora. Hoy en día basta con un pequeño dron de entre 400 y 1800 euros<sup>6</sup> por jornada e incluirían en el tramo más elevado piloto y operador de cámara. Como se puede constatar fácilmente el ahorro incurrido no es para nada desdeñable y ofrece a pequeñas productoras acceder a éste tipo de grabaciones que de otro modo serían privativas para éstas. Aunque aspecto económico es especialmente importante para decantarse por el uso de este tipo de medios, existen en el mundo audiovisual trabajos que no hubiesen sido posibles hasta este momento gracias a los drones. Grabaciones en las que el riesgo humano y material que habría que asumir es tan alto que tan solo de éste modo, debemos por tanto a los drones fotografías como ésta que si bien nos se tomó como parte de ningún proyecto cinematográfico nos hace una idea de lo que la robótica aérea puede ofrecernos.

---

<sup>4</sup>Vertical take off and landing

<sup>5</sup>Mantenerse estáticamente en un punto elevado

<sup>6</sup>Precios aproximados extraídos de la empresa especializada World Aviation Helicopters <https://www.worldaviation.es/es/servicio-drones.aspx>



Figura 1.1: Dron grabando en el interior de un crater

Otro sector que ha adaptado rápidamente el uso de éstas pequeñas aeronaves es el agropecuario donde se utiliza para medir la condiciones del terreno, con el fin de recoger información sobre la hidratación, la temperatura o el ritmo de crecimiento de los cultivos. Controlan el riego e incluso esparcen los pesticidas de manera eficiente siendo un arma eficaz contra las plagas, se utilizan incluso como espantapájaros. La aplicación de drones en este sector se remonta a 1983 cuando el Ministerio de Agricultura de Japón preocupado por el envejecimiento de su población rural encargó a Yamaha el desarrollo de una aeronave no tripulada capaz de realizar varias tareas de las anteriormente descritas a fin de atraer más gente al medio rural. En 1990 se entregaron las primeras unidades del Yamaha RMAX y actualmente el 40 % de los arrozales japoneses cuentan con un dron sobrevolándolos.



Figura 1.2: Dron Yamaha RMAX fumigando

Son utilizados también por los servicios de rescate tras una catástrofe para evaluar los daños

experimentados y ayudas a encontrar supervivientes entre los escombros. Algunos son capaces de enviar a los supervivientes paquetes de supervivencia con salvavidas, alimentos o agua mientras esperan su rescate<sup>7</sup>

E incluso se han empezado a utilizar por el Ministerio de Hacienda de España con fines recaudatorios, sobrevuelan las viviendas o fincas localizando edificaciones no registradas por el catastro con el fin de regularizar su situación.

El uso de drones es especialmente importante en el ámbito científico para por ejemplo tomar medidas de temperatura y CO<sub>2</sub> en zonas peligrosas<sup>8</sup>, estudiar las nubes volcánicas<sup>9</sup> o volar dentro de una grieta en un glaciar<sup>10</sup>.

El futuro del uso de los drones se está escribiendo en este mismo momento y es que poco a poco se investigan con nuevos usos o cómo mejorar los ya actuales.

Uno de los usos más prometedores que esta siendo investigado es el uso de drones para la logística y paquetería, Amazon estudió en 2005 la viabilidad de utilizar drones para el reparto, especialmente en zonas de difícil acceso o bien alejadas de las zonas habituales de reparto. A finales de dicho año realizó pruebas de entregas en el reino unido y desarrollo un prototipo de alrededor de 25Kg de peso. El objetivo de Amazon, entregas de menos de 3Kg en 30 minutos y una toma de contacto en la entrega de paquetes en zonas pobladas. Esto a dado pie a las principales empresas de logística y paquetería a realizar sus propios desarrollos y constuir sus propios prototipos. DHL ha realizado sendas pruebas con drones de cerca de 5 Kg para la entrega de paquetes de hasta 1,2 Kg. La empresa francesa GeoPost se ha asociado con la empresa Atechsys, especializada en el desarrollo de sistemas autónomos para aeronaves no tripuladas, que ha dado como resultado un dron que cuenta con seis rotores eléctricos y estructura de fibra de carbono, con capacidad para llevar paquetes de un peso aproximado de hasta 2 kilos. UPS ha realizado un experimento en tampa con un octocóptero que despegaría desde el techo de la furgoneta de reparto para entregar los paquetes en zonas rurales y ahorrar en kilometraje el dron tenía una carga de pago de unos 4,5Kg.

Otro de los estudios que más llama la atención es el de Google y Facebook que compiten

---

<sup>7</sup><http://www.elmundo.es/economia/2015/09/15/55f8239546163fc6598b45c3.html>

<sup>8</sup><http://www.igepn.edu.ec/servicios/noticias/1395-medidas-de-temperatura-y-co2-de-las->

<sup>9</sup><https://www.nasa.gov/topics/earth/earthmonth/volcanic-plume-uavs.html>

<sup>10</sup>[http://tn.com.ar/tecnologia/recomendados/increible-un-drone-volo-dentro-de-un-glaciar\\_648661](http://tn.com.ar/tecnologia/recomendados/increible-un-drone-volo-dentro-de-un-glaciar_648661)

en llevar internet a zonas aisladas a través de una red de drones y satélites. Facebook presentó en su F8 en Marzo de 2015 su prototipo de dron a tal efecto, Aquila, fruto de la adquisición de la empresa especializada en robótica aérea Acenta. Este proyecto se enmarca dentro del plan internet.org y liderado Connectivity Lab<sup>11</sup> que pretende ofrecer internet con un coste reducido a todo aquel que no lo tenga.

## 1.4. Robótica aérea en el laboratorio de Robótica de la URJC

Dentro del laboratorio de robótica de la Universidad Rey Juan Carlos cabe destacar los trabajos realizados para profundizar investigar y experimentar con ellos.

Trabajos como el de Alberto Martínez Florido permitió controlar un AR-Drone real desde una aplicación cliente, desarrollando para ello un driver para el dron, ardrone\_server y una aplicación cliente UAV Viewer. El driver ardrone\_server es capaz de conectar con el AR-DRONE a través de comandos AT tanto para obtener los datos de todos sus sensores como para controlarlo y expone todo ello en interfaces JdeRobot para su interconexión con todo el software disponible en el framework. El AR-Drone es un quadracóptero que en su versión actual cuenta con acelerómetros, giróscopos y magnetómetros en 3 ejes que determinan junto con su barómetro y un sensor de ultrasonidos la actitud del mismo, dispone además de 2 cámaras, una ventral y otra frontal y algunos modelos traen sensor GPS. Alberto desarrolló también el software de control UAV Viewer, esta aplicación cliente es capaz de controlar con cualquier dron y mostrar de forma visual su actitud, lo que captan sus cámaras.

---

<sup>11</sup>Un equipo formado por 50 expertos en aeronáutica y ciencia espacial

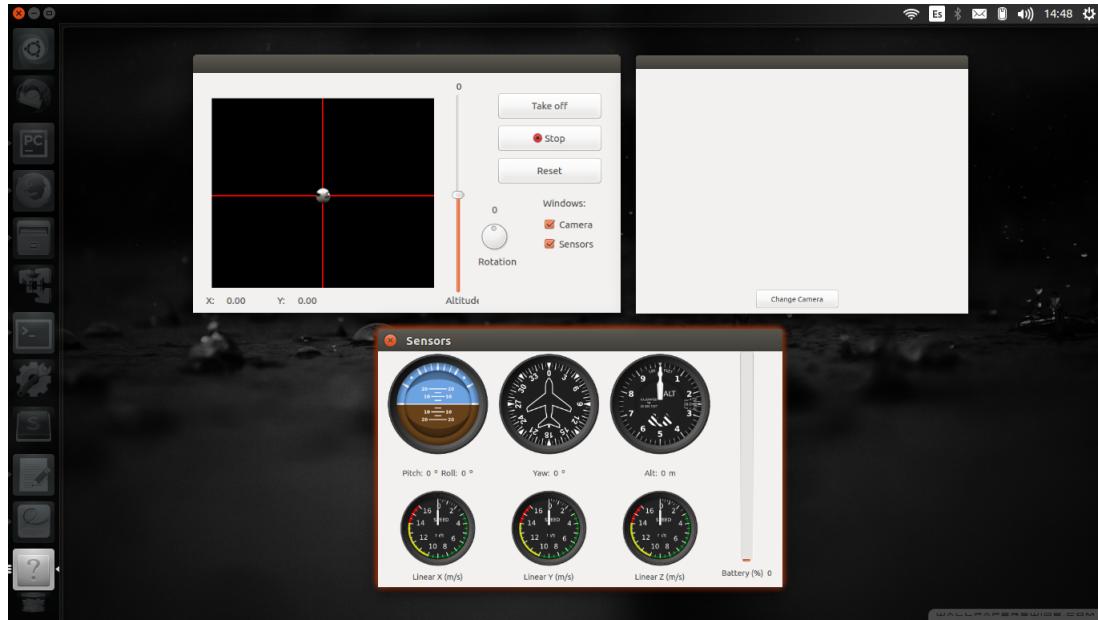


Figura 1.3: Ventanas de UAV Viewer

Daniel Yague desarrolló un driver para poder simular el comportamiento del AD-Dronde en el simulador Gazebo, de referencia en el departamento de robótica de la universidad y contenido dentro de la suite JdeRobot. Con este desarrollo se hizo posible desarrollar, probar anticipar los problemas que puedan surgir en el vuelo del dron antes siquiera de volarlo, de esta forma es posible el desarrollo de aplicaciones de navegación complejas sin disponer de el, sin arriesgarlo e independientemente de factores externos como la climatología. Para mostrar las posibilidades de desarrollo que se abrían desarrolló varias aplicaciones en el ar-drone simulado como un gatarrón donde un dron teleoperado trataba de huir de otro que le seguía o una aplicación donde el dron seguía una carretera.

Jorge Cano construyó propio su dron utilizando como base un quadracóptero con un Intel Compute Stick (ICS) STCK1A8LFC como ordenador de abordo y una placa Pixhawk como placa estabilizadora/piloto automático. Esta placa utiliza como protocolo de comunicación MAVLink<sup>12</sup> y dispone de acelerómetros, giróscopos y magnetómetros para determinar la actitud. Además de la construcción del drone Jorge desarrolló el driver MAVLinkServer donde apoyándose en MAVProxy adaptó los comandos MAVLink a interfaces JdeRobot permitiendo acceder a la actitud y controlar el drone con comandos tipo GotoXY enviados a través del interfaz Po-

<sup>12</sup>Micro Air Vehicle Link <https://en.wikipedia.org/wiki/MAVLink>

se3D<sup>13</sup>



(a) Intel

(b) camaras

En la actualidad se están desarrollando nuevos trabajos sobre este tipo de placas estabilizadoras que tiene como software base ArduCopter/Ardupilot o son compatibles.

Diego Jimenez trabaja en controlar un Solo Drone de la empresa 3DR<sup>14</sup> mediante el interfaz de velocidades CMDVel<sup>15</sup>, este quadracóptero tiene como placa de control una Pixhawk como la utilizada por Jorge Vela en el dron que se construyó y utiliza también comandos MAVLink.

Jorge Vela se encuentra desarrollando como realizar la maniobra de aterrizaje de forma automática al localizar un patrón o baliza localizado visión, como dron están utilizando un Solo Drone con una Intel Stick como ordenador de abordo.

---

<sup>13</sup>Ver capítulo 3 Arquitectura utilizada

<sup>14</sup>Empresa norteamericana con sede en California especializada en robótica aérea. Se sitúa en 2017 como la 3<sup>a</sup> empresa del sector

<sup>15</sup>Ver capítulo 3 Arquitectura utilizada



# **Capítulo 2**

## **Objetivos**

[PRELIMINAR SOLO IDEAS] Los objetivos de este PFC son arrojar luz sobre el uso de drones de ala fija poniendo en valor éstos en situaciones donde la carga de pago, la autonomía, velocidad o consumo desaconsejan el uso de los más extendidos quadracópteros.

Para ésto hemos adaptado un avión de radio-control comercial añadiéndole toda la aviónica necesaria para que puede operar como drone. Hemos desarrollado un driver para JdeRobot del interfaz MAVLink que nos permite recoger actitud, velocidades lineales y angulares, altura, posición y otros parámetros como la batería restante. Nos permite también la actuación con el seguimiento de misiones que no sólo incluyen el paso por ciertos puntos de control o waypoints sino que también soporta las maniobras de despegue y aterrizaje en drones de ala fija. Hemos desarrollado también un software de control que da soporte a este tipo de actuación de más alto nivel, las misiones.



# Capítulo 3

## Infraestructura utilizada

### 3.1. Hardware

Este PFC se apoya principalmente en 2 piezas hardware:

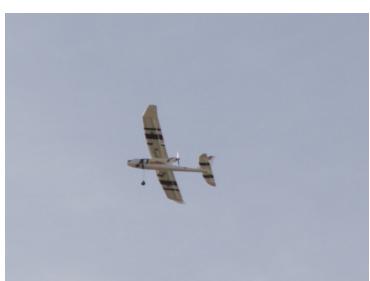
- El avión a radio-control. Para este desarrollo hemos elegido el avión Bix3 distribuido por la empresa china [www.hobbyking.com](http://www.hobbyking.com), hemos elegido este modelo por ser un avión muy estable debido principalmente a sus cualidades como velero y su alta superficie alar. Otro aspecto importante de la construcción del avión es que el propulsor no se encuentra en el frontal del avión lo que nos permitirá aprovechar al máximo esa zona pudiendo poner incluso una cámara frontal.



(a) Bix3



(b) Vista de cerca



(c) Bix3 en vuelo

Para poder cargar con el equipo necesario nos ha sido necesario cambiar el motor, el variador y las baterías de serie por otras de mayor rendimiento que nos permitan volar con tanta carga de pago.

- Una raspberry PI3 que es nuestro ordenador de abordo y en el que instalaremos la infraestructura necesaria para que podamos dotarle de inteligencia. Hemos elegido éste dispositivo debido al compromiso peso/potencia que nos otorga así como porque se trata de un hardware muy asequible y extendido. Conectado a esta PI3 irá una PiCam que nos permitira ver lo que el avión vea.

### **3.2. Placa estabilizadora**

El estabilizador/piloto automático, para este desarrollo hemos optado por un Ardupilot Mega<sup>1</sup>. Este dispositivo tiene como base una placa Arduino Mega a la que se le han incorporado, giróscopos y acelerómetros en 3 ejes para su estabilización y que trae de serie un receptor GPS con brújula. El kit que adquirimos incluye también esta placa ataca directamente sobre los actuadores del avión (los servos y el motor) a través de señales PWM.

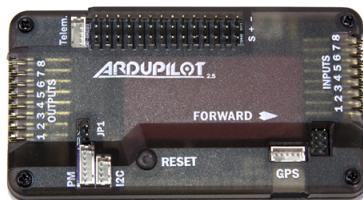


Figura 3.1: Placa estabilizadora ardupilot

Esta placa ha de flashear con un firmware en función del tipo de aeronave que utilicemos como base, Arducopter para alas rotatorias o Ardupilot para alas fijas.

Esta y otras placas similares como PixHawnk ofrecen un interfaz que se apoya en comandos llamado MAVLink. A través de comandos MAVLink se puede acceder a los sensores, a los que incorpora la placa o a los que le añadamos a la misma como el GPS o sensores de velocidad del aire. A través de estos comandos se le puede también enviar ordenes al piloto automático quien las ejecutará más adelante trataremos el protocolo MAVLink en profundidad. Cabe sin embargo destacar ya en este punto no sólo que algunos de los comando que mandemos tendrán distinto comportamiento en la placa en función del firmware escogido sino que hay set de comandos que no pueden ser utilizados en uno o en otro.

<sup>1</sup><http://www.ardupilot.co.uk/>

### 3.3. JdeRobot

JdeRobot es un framework desarrollado por el laboratorio de robótica de la Universidad Rey Juan Carlos, para el desarrollo de aplicaciones de robótica. Su última realease la 5.5 se liberó el 15 de Marzo de 2017 pudiendo ver los detalles de ésta en el github oficial<sup>2</sup>. JdeRobot se compone de interfaces, drivers, utilidades y aplicaciones para el desarrollo de cualquier proyecto de robótica, se apoya en estos interfaces, algunos de ellos los veremos en profundidad a continuación, para interconectar entre sí todos los aplicativos del mismo y en Zeroc ICE para la comunicación entre ellos. Algunos de los driver mas importantes que contiene serían:

1. Camer(server. Se trata de un driver para enviar imágenes y video a través del interfaz camera
2. Gazebo(server. Driver desarrollado para conectar la herramienta de simulación Gazebo con JdeRobot y así poder simular los desarrollos.
3. MAVLinkServer. Desarrollado para intercomunicar JdeRobot con placas que utilicen el protocolo de comunicación MAVLink.
4. Ardrone\_server. Driver que conecta el Parrot Ar-Drone a JdeRobot. Este driver escrito en c++ transforma el set de comandos AT del drone en interfaces y viceversa, implementa los interfaces camera, cmdvel, navdata, extra y pose3D y permite acceder a la actitud del drone así como a sus 2 cámaras. Sirve también datos como el nivel de la batería y permite grabar vídeo o tomar fotos.

Algunas de las aplicaciones desarrolladas más importantes serían:

1. Cameraview. Se trata de una aplicación desarrollada en c++ capaz de recibir vídeo a través del interfaz camera.
2. UAV viewer. Aplicación desarrollada como ground control de robots aéreos. Esta aplicación permite teleoperar cualquier tipo de robot aéreo y ofrece de forma visualmente atractiva datos como la actitud, velocidades lineales y angulares, ofrece también la posibilidad de visualizar videos servidos por el interfaz camera.

---

<sup>2</sup><https://github.com/JdeRobot/JdeRobot/wiki/JdeRobot-5.5.0>

### 3.3.1. Interfaces

JdeRobot expone más de 30 interfaces pero en este capítulo explicaremos los que durante nuestro desarrollo hemos implementado:

- Pose3D. Utilizado para recoger los datos de actitud y la posición de la aeronave.

Pose3DData

```

{
    float x; /* x coord */
    float y; /* y coord */
    float z; /* z coord */
    float h; /* */
    float q0; /* qw */
    float q1; /* qx */
    float q2; /* qy */
    float q3; /* qz */
};
```

- Camera. Utilizado para servir imágenes.

```

class CameraDescription
{
    string name;
    string shortDescription;
    string streamingUri;
    float fdistx;
    float fdisty;
    float u0;
    float v0;
    float skew;
    float posx;
    float posy;
```

```
    float posz;  
    float foax;  
    float foay;  
    float foaz;  
    float roll;  
};
```

- NavData. Utilizado para servir datos secundarios de actuación como velocidades lineales o angulares o el estado de la batería.

```
class NavdataData  
{  
    int vehicle; //0-> ArDrone1, 1-> ArDrone2  
    int state; // landed, flying,...  
    float batteryPercent; //The remaing charge of baterry %  
  
    //Magnetometer Ardrone2.0  
    int magX;  
    int magY;  
    int magZ;  
  
    int pressure; //Barometer Ardrone2.0  
    int temp; //Temperature sensor Ardrone2.0  
    float windSpeed; //Estimated wind speed Ardrone2.0  
  
    float windAngle;  
    float windCompAngle;  
  
    float rotX; //rotation about the X axis  
    float rotY; //rotation about the Y axis  
    float rotZ; //rotation about the Z axis
```

```

int altd; //Estimated altitude (mm)

//linear velocities (mm/sec)
float vx;
float vy;
float vz;

//linear accelerations (unit: g) ¿Ardron2.0?
float ax;
float ay;
float az;

//Tags in Vision Detectoion
//Should be unsigned
int tagsCount;
arrayInt tagsType;
arrayInt tagsXc;
arrayInt tagsYc;
arrayInt tagsWidth;
arrayInt tagsHeight;
arrayFloat tagsOrientation;
arrayFloat tagsDistance;

float tm; //time stamp
};

```

- **Extra.** Utilizado principalmente para las órdenes de despegue y aterrizaje.

```

void land() - land drone.
void takeoff() - takeoff drone.
void reset()

```

```

void recordOnUsb(bool record)
void ledAnimation(int type, float duration, float req)
void flightAnimation(int type, float duration)
void flatTrim()
void toggleCam() - switch camera.

```

## 3.4. MAVLink

### labelsec:mavlink

MAVLink siglas de Micro Air Vehicle Link es un protocolo de comunicación desarrollado para comunicar las placas estabilizadoras con piloto automático a los GCS o Ground control station, las aplicaciones desde las que se podía enviar misiones y seguir el cumplimiento de las mismas desde tierra. MAVLink se publicó en 2009 por Lorenz Meier, publicado bajo licencia LGPL aspira a convertirse en el protocolo standard en robótica aérea y se ha probado su funcionamiento en PX4, PIXHAWK, APM<sup>3</sup> y Parrot AR.Drone.

Un ejemplo de comando MAVLink sería:

```

type GpsStatus struct {
    SatellitesVisible uint8      Número de satélites visibles
    SatellitePrn     [20]uint8   Id Global de cada satélite
    SatelliteUsed    [20]uint8   Lista con el uso de cada satélite
    SatelliteElevation [20]uint8  Elevación, nos da el ángulo sobre el horizonte
    SatelliteAzimuth  [20]uint8  Dirección del satélite, 0: 0 grados, 255: 360
    SatelliteSnr      [20]uint8  Señal/ruido de cada uno de los satélites
}

```

Este mensaje trae la información del enlace actual con el GPS y se envía periódicamente en ciclos que decidimos en parámetros de conexión con el dispositivo. Otro parámetro, esta vez vinculado a la actuación sería:

```

type MissionItem struct {
    Param1          float32    parámetro variable en función del comando.
    Param2          float32    parámetro variable en función del comando.

```

---

<sup>3</sup>Ardupilot Mega

```

Param3          float32    parámetro variable en función del comando.
Param4          float32    parámetro variable en función del comando.
X              float32    latitud
Y              float32    longitud
Z              float32    altitud
Seq             uint16    Número del item en la misión
Command         uint16    Tipo de comando de navegación.
TargetSystem    uint8     ID del sistema
TargetComponent uint8
Frame           uint8     Sistema de coordenadas que se utiliza.
Current          uint8    Misión actual no:0, si:1
Autocontinue    uint8    Autocontinuar al siguiente objeto de misión.
}

}

```

### 3.5. Python y PyQt5

Python es un lenguaje de programación interpretado y multiplataforma que nació en los años 80 en los países bajos con idea de hacer más legible el código. El lenguaje de programación que inicialmente se utilizaba principalmente para scripting, ha sabido crecer con los años y con la publicación de Python3 en 2009 ha recibido el impulso que necesitaba para ser hoy en día el 5º lenguaje más utilizado por encima de PHP, .NET y Javascript que baja hasta el 8º puesto según TIOBE en un estudio de Abril de 2017.

El porqué de utilizar Python, muy sencillo mantiene el carácter multiplataforma de JdeRobot, su código es simple y legible y trabaja muy bien con dependencias muy utilizadas en robótica como OpenCV.

Para nuestro desarrollo hemos utilizado PyQt5 para desarrollar el interfaz gráfico. PyQt5 es un binding de Qt5 en forma de librería python que nos permite acceder a toda la funcionalidad de Qt5. Qt, propiedad de Nokia, es un conjunto de librerías escritas en C++ para interfaces gráficas.

## 3.6. Mapas y Geo-referenciación

En nuestro desarrollo hemos tenido que trabajar con mapas geo-referenciados. Estos mapas, que tanto se han popularizado gracias a Google, son una herramienta imprescindible en robótica aérea si se quiere trabajar con largas distancias.

Un mapa geo-referenciado es aquel en el que conocemos o podemos calcular la posición en el planeta que representa cada pixel del mismo.

La forma mas común de obtener estos mapas geo-referenciado es a través de WMS, siglas de Web Map Service. Un WMS nos es mas que un servicio web que recibe como entrada unas coordenadas y una serie de parámetros y devuelve una imagen encuadrada en los datos enviados. En nuestro desarrollo hemos utilizado 2 WMS el del IGN<sup>4</sup> y el de Google, a continuación vamos a desgranar un WMS, el de PMOA del IGN.

El WMS PMOA requiere como entrada 4 atributos principales:

1. La posición GPS.
2. Bounding Box. El Bounding box son 2 puntos que corresponden con la posición GPS que queremos que sea el extremo inferior izquierdo de nuestra imagen y con el punto superior derecho de la misma.
3. Datum. El datum es el sistema de referencia o proyección de la tierra a utilizar. En el caso de IGN pese a que soporta varias utilizamos WGS84 que es el estandar.
4. Tamaño de la imagen, el tamaño que queremos que tenga la imagen obtenida.

El resultado un mapa del que conocemos el punto central, y los extremos inferior izquierdo y superior derecho y los pixeles que tiene a lo ancho y alto del mismo, un mapa geo-referenciado.

## 3.7. SITL

SITL siglas de Software in the loop es un simulador que te permite a cualquier programa que envíe o reciba comandos MAVLink ejecutar pruebas sin necesidad de tener ninguna placa estabilizadora y evitando la pérdida de la aeronave en caso de error del mismo. SITL se conecta con JSBSim, un simulador de vuelo de software libre para ejecutar el aspecto físico de la

---

<sup>4</sup>Instituto Geográfico Nacional de España

simulación y con MAVProxy para el envío de commandos y seguimiento de misiones. Se trata de una compilación en C++ de ardupilot y se podría asemejar a nuestro gazeboserver. SITL puede conectarse también con Gazebo o a FlightGear para hacer mas completa su simulación mostrando el vuelo en un entorno desarrollado para pruebas con elementos del mundo real.

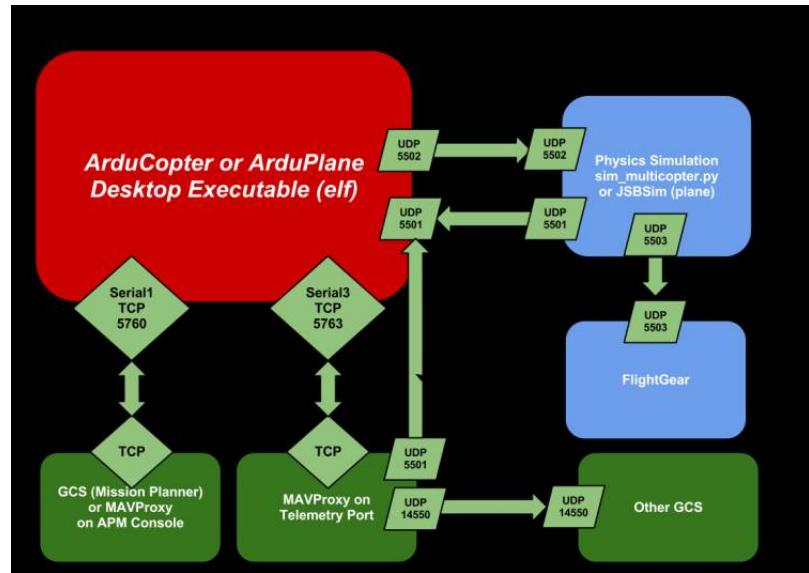


Figura 3.2: Arquitectura de SITL

# **Capítulo 4**

## **Infraestructura Desarrollada**

Descripción de las tecnologías que utilizas en tu trabajo. Con dos o tres párrafos por cada tecnología, vale.

Puedes citar libros, como el de Bonabeau et al. sobre procesos estigmérgicos [1].

También existe la posibilidad de poner notas al pie de página, por ejemplo, una para indicarte que visite la página de LibreSoft<sup>1</sup>.

### **4.1. Sección 1**

---

<sup>1</sup><http://www.libresoft.es>



# **Capítulo 5**

## **AUV Commander**

### **5.1. Arquitectura general**

figura 5.1.

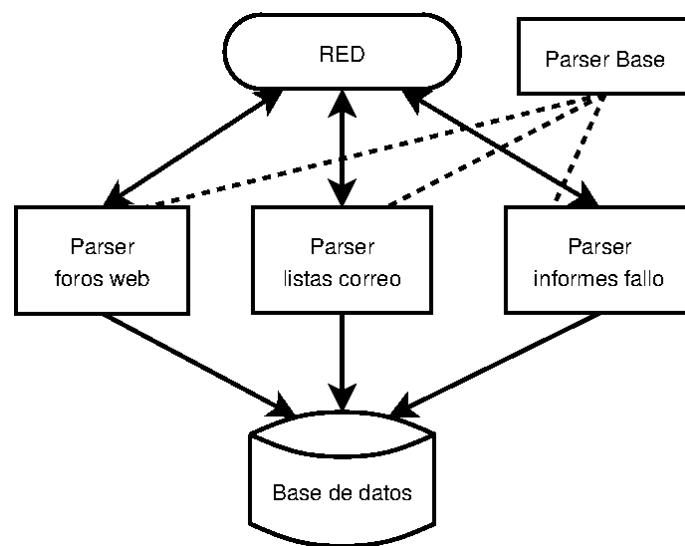


Figura 5.1: Estructura del parser básico

# **Capítulo 6**

## **Experimentos**



# **Capítulo 7**

## **Conclusiones**

### **7.1. Consecución de objetivos**

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemás se han encontrado y qué medidas se han tomado para mitigar esos problemás.

### **7.2. Aplicación de lo aprendido**

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

### **7.3. Lecciones aprendidas**

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. a

2. b

## **7.4. Trabajos futuros**

Ningún software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFMs.

## **7.5. Valoración personal**

Finalmente (y de manera opcional), hay gente que se anima a dar su punto de vista sobre el proyecto, lo que ha aprendido, lo que le gustaría haber aprendido, las tecnologías utilizadas y demás.

# **Apéndice A**

## **Manual de usuario**



# Bibliografía

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Inc., 1999.