Capítulo 1

Infraestructura utilizada.

En esta parte voy a hablar sobre los diferentes elementos y aplicaciones para el desarrollo de este proyecto.

En primer lugar hay que hablar del **Parrot AR.Drone 2.0**, dron utilizado para realizar las pruebas y poder comprobar de esta manera que el desarrollo del programa era el correcto.

La *batería* de este era de 1500mah, lo que en algunas ocasiones nos ha llevado a problemas, pues el tiempo de vuelo con esto es unos 12 minutos aproximadamente. Esto conllevaba que algunas pruebas en ocasiones eran muy difíciles de realizar, ya que si no salia el resultado esperado en los primeros intentos muchas veces había que parar la prueba durante bastante rato para continuar. Si las pruebas no eran de vuelo sino que era de imagen con este tiempo era suficiente, ya que el dron no gastaba tanta energía al relizar solo la retransmisión de los datos que obtenía la cámara. Un detalle curioso en este punto es el cambio de funcionamiento que se notaba una vez la batería estaba por debajo del 30/100, pues el despegue se hacia con menos fuerza y de forma más inestable, además las instrucciones de movimiento que se le mandaban las ejecutaba con menos fuerza y por lo tanto de forma más lenta. La única solución posible para remediar este problema fue trabajar a la vez con 3 baterias, pero he de decir que en momentos que las pruebas eran muy continuas durante mucho tiempo, llegaba un momento que había que parar porque el tiempo de carga no era suficientemente rapido comparado con el de las 3 descargas.

El *alcance* que tiene éste con la estación tierra es de 50 metros, suficiente para las pruebas que han sido realizadas en este proyecto.

Este dron tiene una *placa base* ARM Cortex A8 de 1Ghz y un DSP de vídeo de 8Ghz. También posee una memoria RAM DDR2 de 1GB y 200Mhz. Este chip es el encargado de levantar una red WiFi, a la cual nos podemos conectar mediante el smartphone (hay una aplicacion específica para ello), o como en este caso, mediante el ordenador, y de esta forma poder controlarlo. Esta placa base funciona con Linux.

Además, dicho dron cuenta con dos *cámaras*, una horizontal y otra vertical, lo que nos permite ver diversos planos en todo momento. Estas cámaras están conectadas a la placa base, lo que permite en todo momento la transmisión de imágenes en directo, gracias a las cuales podremos trabajar, sera la información gracias a la cual podremos darle instrucciones al dron. Cabe destacar la diferencia de calidad que hay entre las cámaras, algo que se notaba al realizar pruebas con una cámara, y cuando todo era correcto y querias cambiar de cámara para realizar otra prueba se podía observar como los colores perdian nitidez y se despreciaban algunos detalles, lo que te llevaba a cambiar cosas de los algoritmos que en un principio no se esperaban.

Ya para terminar, la *velocidad* de este drone puede llegar a alcanzar los 18km/h, algo que en este proyecto nunca he llegado a probar, pero sería algo poco aconsejable teniendo en cuenta la distancia de alcance, pues los 50 metros que tiene, si la estación terrena de comunicación esta en un punto fijo, el dron tardaría 10 segundos en salirse de este rango.

Tras este dron, tengo que hablar del dron **3DR solo**. En un principio era este sobre el que iba a funcionar el software realizado en el proyecto, pero al final hubo diversos problemas debido al software del dron, pues no lleva una cámara integrada, sino que es una GoPro que va por otra parte la que se puede a?adir para utilizar, se estuvieron valorando diversas opciones para que este tuviera una cámara de visión que contaré más adelante, pero dado que era un dron con unas muy buenas característcas y que se trabajaba de forma excelente con él, voy a comentar algunos detalles suyos.

Lo primero es que cuenta con *dos procesadores*, uno en el drone y otro en el mando de control. Destacar que la del mando del control se trata de una Pixhawk de 3DRobotics. Es importante decir que el drone y el mando se comunican mediante radiofrecuencia, y es el mando el que levantaba la red WiFi para poder conectarte al sistema. Esto es otro detalle negativo que tuvimos al trabajar con este dron, pues no podíamos prescindir del mando si en algun momento el proyecto llega a tal punto, lo que lleva al software a tener una limitacion de distancia, aunque ésta sea considerable. Desde el mando era el que se le enviaban las instrucciones al dron, pero con software específico se le podía enviar las instrucciones al mando y ya este comunica con el dron.

la *imágen en directo* la podíamos ver a través del teléfono a gran calidad (a?adiendo la GoPro) pero situando la colocación de ésta antes del despegue.

Otro punto muy importante era la *bateria*, esta tenía 5.200mAh de capacidad, que siempre nos fue suficiente para las pruebas que se realizaron, pero permite un tiempo de vuelo de 20 minutos. Decir que con este dron hubo menos tiempo de vuelo que con el anterior.

Y por último hay que hablar sobre la potencia que tiene el *motor* que permite alcanzar los 88.5km/h. Un nconveniente de tan alta potencia, son las corrientes internas que puede generar. En exteriores esto no supone nada ya que cuando se levanta aproximádamente un metro sobre el suelo, es capaz de estabilizarse él solo. Sin

embargo, en interiores puede desestabilizarlo bastante, y como ocurrió en una prueba, ir dirección a la pared y sufrir algún da?o como el romperse una hélice.

Dejando de un lado los drones, otro material con el que se trabajó fue la **Intel Compute Stick** es un dispositivo del tama?o de la palma de la mano que nos permite llevar un ordenador encima en todo momento (dispone incluso una peque?a pesta?a para poder a?adirlo al llavero) y que podemos conectar a cualquier pantalla con conexion HDMI. Además tiene un puerto USB 3.0, al cual podemos conectar un hub para poder a?adir un teclado, ratón u otro dispositivo de almacenamiento de datos. Éste necesitaba un cable de alimentación para poder iniciarse. Su uso muy favorable, pues gracias a él podríamos llevar el software realizado encima en todo momento, con todas las necesidades para poder conectarlo a un dron y de esta forma el algoritmo pudiera ejecutarse sin problemas, ya que gracias a su antena WiFi podrá conectarse al dron para poder comunicarse. Por otro lado, se nota que no está preparada para grandes tareas de ejecución (tiene 2Gb de memoria RAM), y por tanto, a la hora de ejecutar programas pesados, la velocidad de éstos se ve notablemente disminuida, factor negativo teniendo en cuenta lo importante que es la velocidad de procesamiento de la información y transmisión de las instrucciones.

También ha sido muy importante para el proyecto el poder probar los distintos avances en el simulador **Gazebo**, el cual nos posibilita hacer pruebas en cualquier momento y sin preocuparnos de da?os materiales. Gazebo permite la creación de entornos de manera precisa y poder probar los robots en distintos tipos de ambientes. Se trata de un programa OpenSource, lo que ha permitido su expansión con facilidad, y muchos a?adidos como plugins o repositorios con robots comerciales para poder acceder a su uso. Esta plataforma ha sido en la que se ha trabajado principalmente durante todo el proyecto, pues existe en ésta un simulador de ArDrone, el cual se controlaba a través de las herramientas de JdeRobot. El escenario principal aquí utilizado consistía en el dron y un coche con una baliza, el cual tenía que encontrar, centrarse sobre ella y aterrizar. Cabe destacar que había una gran diferencia en los movimientos comparando con el dron real. Esto ha sido un problema en los periodos que se ha ido desarrollando el algoritmo en las dos partes a la vez, pues un mínimo retoque en Gazebo podía suponer un gran problema en el real, aunque para probar como podían ser los distintos movimientos ha sido de gran ayuda, así como la parte de visión, ya que era muy sencillo comprobar cualquier modificación, ver si algo funcionaba sobre un entorno perfecto y de esta forma haber mejorado esta parte continuamente.

Como software principal se ha trabajado con **JdeRobot**. De aquí se han trabajado con diversas herramientas, las cuales nos han permitido seguir unos pasos firmes desde un inicio hasta el final del desarrollo. En primer lugar se utilizo *color filter* para ver el funcionamiento de los filtros de color. Esta herramienta nos permite, a partir de una imagen o vídeo, trabajar en los distintos espacios de color (como puede ser RGB, HSV, HSI...) y poder ir variando los parametros máximo y mínimo (entre 0 y 255) de

cada uno para ver que colores cumplen las características y cuales no, viendose de esta forma, en otra imagen, sólo los colores que pasan este filtro dejando el resto como un fondo negro. Una vez utilizada esta herramienta y realizamos peque?os algoritmos para ver que funcionaba igual un filtro propio que este filtro, pasamos a trabajar con otra llamada *follow turtleboot*. Ésta nos permitía manejar el dron, teniendo un cuadro que nos permite el movimiento en los ejes X e Y, una barra para realizar un cambio en la altura y un peque?o circulo para que pudiera rotar sobre si mismo. Por otro lado tenía un boton para aterrizaje y despegue y unos botones que permiten iniciar un algoritmo creado, de forma que podemos probar si lo programado funciona. Esta herramienta puede sacar tres cuadros diferentes, uno para la cámara, el cual nos da una imagen en directo de lo que está transmitiendo el dron y nos da la opción de elegir la cámara que queremos ver. El segundo, nos muestra las distintas propiedades del dron en ese momento como la orientación, altura o inclinación. Por último, el último cuadro es para el filtro de color. Éste funciona cuando el algoritmo programado está en marcha. En un primer momento como indica su nombre, nos permite trabajar con el filtro de color que habíamos practicado antes sobre la otra herramienta, pero al final utilizamos éste para realizar marcas sobre la imagen de las cosas que nos interesaban de esta, como puede ser encuadrar la baliza, la cruceta o pintar sobre esta. Por último, a todo esto le hemos a?adido un diagrama de estados, gracias a una peque?a aplicación desarrollara a partir de *visualHFSM*. nos permitía crear un dibujo de los diferentes estados por los que debería pasar el dron, según lo que estuviera viendo en cada momento o la acción concreta que estaba realizando. De esta forma se marca en el diagrama en el estado en el que se encuentra, pudiendo saber así cual sería el siguiente estado esperado y viendo que funciona de forma correcta.

En la siguiente figura podemos observar un ejemplo de este conjunto, pudiendo ver el diagrama de estados y la imagen con los datos interesantes marcados:

Ya por último, decir lo utilizado para programar el algoritmo. Este se ha desarrollado utilizando Python, gracias al cual hemos podido utilizar bibliotecas como PIL o OpenCV principalmente para el manejo de imágenes. Nos permitían obtener la imagen transmitida por el dron con facilidad y aplicarle los filtros de color deseados en cada momento, así como realizar realizar distintas operaciones morfológicas (erosión , dilatación, apertura y cierre) sobre los fotogramas en cada momento, obteniendo imágenes más claras sobre las cuales podemos trabajar y con sus funciones, como puede ser drawcontours, obtener un código más compacto y eficiente. Esto nos permite un mejor rendimiento del algoritmo, lo que nos llevará a unos mejores resultados.

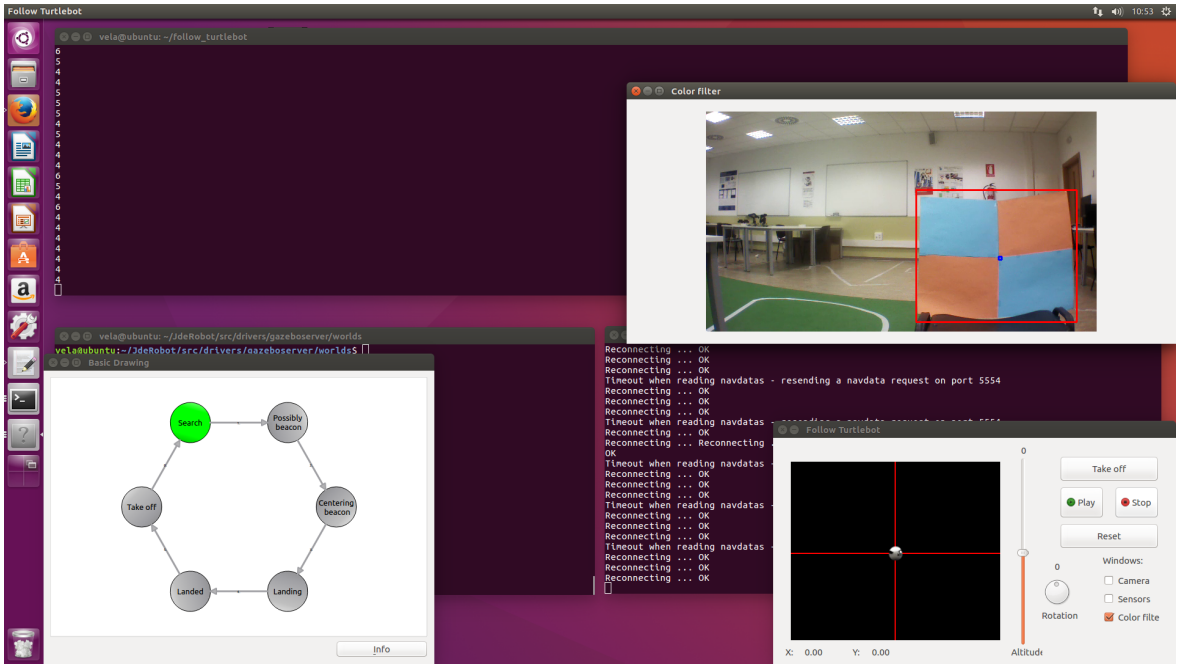

Figura 1.1: Esta imagen muestra el conjunto de herramientas utilizadas.