toopazo.github.io

Primera versión: 06-07-2021 | Última actualización: 30-04-2022 | Categoría: CyT de Drones

Introducción a las tecnologías open-source para vehículos aéreos autónomos

Las compañías de Drones dominantes como DJI, Parrot, SenseFly, Freefly, etc, ofrecen al mercado soluciones cerradas de hardware y software. Sus vehículos son de muy buena calidad y su fama es merecida. Ofrecen además soluciones integradas para un conjunto de actividades relacionadas con la captura de imágenes (producción audiovisual, inspección de infraestructura, aerofotogrametría, vigilancia, etc.)
Pero si nuestra intención es aprender, experimentar, modificar o innovar con esta tecnología, entonces existen alternativas mejores que nos permiten hacer todo esto y más. En este articulo describiremos las alternativas open-source disponibles para diseñar, construir y operar vehículos aéreos autónomos. Los componentes críticos analizados son: (1) hardware, (2) software, (3) comunicación, (4) carga útil y finalmente (5) estructura y actuadores.

Figura 1: NASA's Advance Air Mobility vision

Existen un conjunto de proyectos open-source tanto de hardware como de software apoyados oficialmente por la Fundación Dronecode. Dronecode es, a su vez, parte de la Fundación Linux, la cual se dedica a fomentar el desarrollo y mantenimiento de proyectos de software libre.

En sus propias palabras «The Linux Foundation provides a neutral, trusted hub for developers to code, manage, and scale open technology projects».

Por su parte la fundación Dronecode asegura que

> The Dronecode Foundation fosters communities and innovation through open-standards using open-source. Dronecode is a vendor-neutral foundation for open source drone projects.

Existen por supuesto valiosas iniciativas open-source que no están relacionadas con Dronecode o Linux y que este articulo no menciona. Esto es debido a que la iniciativa Dronecode es la más establecida dentro de la comunidad científico-académica.

Pero más allá de los proyectos específicos que podamos mencionar, la revolución del mundo Drone está en plena macha y ya ha cambiado la forma en que muchas industrias trabajan. Y promete cambiar en el futuro cercano la forma en que carga y pasajeros se mueven dentro de las ciudades.

1. Hardware de vuelo

Uno de los componentes críticos para mantener el vuelo de cualquier aparato es el computador de vuelo. Es la unidad encargada de:

- recibir la información de los sensores (giróscopo, acelerómetro, velocidad relativa, altura, etc)

- recibir los comandos del piloto, piloto automático u otro algoritmo
- estimar las variables de estado del vehículo (velocidad, orientación, etc)
- calcular las variables de control necesarias
- enviar las comandos de control a cada uno de los actuadores (alerones, elevadores, hélices, etc)

La siguiente imagen muestra la relación del computador de vuelo con el resto de componentes de un vehículo aéreo, así como el tamaño y apariencia exterior algunos modelos usados comúnmente por la aviación comercial, helicópteros y UAV militares.

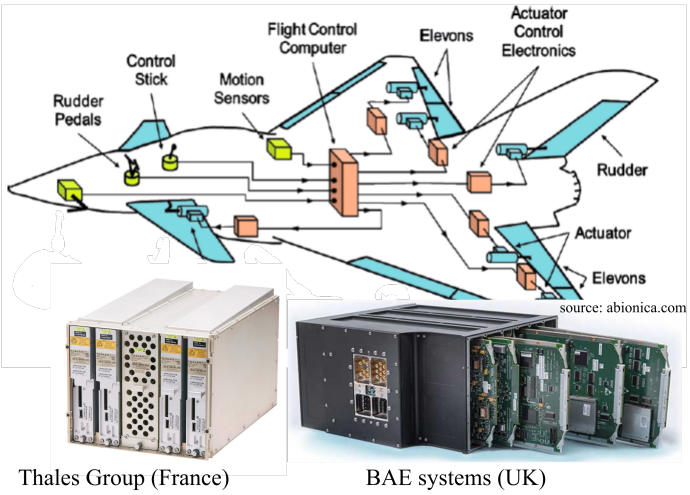


Figura 2: Ejemplos de computadores de vuelo comerciales

En el caso de los vehículos aéreos autónomos de menor tamaño (bajo los 20kg), el computador de vuelo debe ser más liviano, mucho más compacto y de menor costo. Afortunadamente la miniaturización y el avance en la electrónica han permitido cumplir con estas características sin sacrificar demasiado la calidad, seguridad ni robustez del aparato. El computador de vuelo Pixhawk es uno de los proyectos apoyados por Dronecode. Es una estándar que define el tipo de sensores y la interconexión del computador de vuelo con sensores externos, baterías, cámaras u otros aparatos. Es el punto en común desde el que diversos fabricantes comercializan sus productos.

Uno de esos fabricantes es HolyBro que comercializa el producto Pixhawk 4. Una segunda opción es HEX/Proficnc y el producto The Cube Orange Standard Set. Una tercera compañía es Auterion con el producto Skynode. La imagen más abajo muestra estos tres productos.

Figura 3: Ejemplos de computadores de vuelo Pixhawk

Existe un excelente resumen de plataformas de hardware con estándar Pixhawk en la página web de Px4 - flight controller. Respecto a Px4 hablaremos en la siguiente sección, pero por ahora nos basta saber que existen múltiples alternativas de computadores de vuelo.

2. Software de vuelo

El componente complementario al computador de vuelo es el software de vuelo. Se puede describir sucintamente como el algoritmo empleado para mantener en vuelo al vehículo. En otras palabras, es el algoritmo que indica como procesar y dirigir la información desde los sensores y comandos del piloto hacia cada uno de los actuadores del vehículo.

PX4 es un software de vuelo open-source que permite estimar el estado del vehículo y ejecutar algoritmos de control para estabilizar el vehículo usando bucles de control PID (o similares). También forma parte de los proyectos apoyados por la fundación Dronecode y es frecuentemente usado para volar aviones, multi-rotores y diseños experimentales.

2.1. Estimación

La primera tarea que todo software de vuelo debe realizar es el procesamiento de sensores inerciales (acelerómetros, giroscopios, magnetómetro, barómetro, etc), sensores de posición y velocidad como los GNSS (e.g. GPS, GLONASS, BDS y Galileo) y otros (altímetro, odómetro láser, radar, etc). Estos sensores son combinados con a un modelo dinámico del vehículo para obtener una estimación de las variables del

vehículo. Escrito matemáticamente, el vector de estado más común esta definido por

$$\mathbf{x} = \left[x, y, z, u, v, w, \theta, \phi, \psi, P, Q, R\right]^T \quad (1)$$

donde $\mathbf{x}$ es el vector de estado, $[x, y, z]$ la posición, $[u, v, w]$ la velocidad, $[\theta, \phi, \psi]$ la posición angular (i.e. ángulos de Euler) y finalmente $[P, Q, R]$ la velocidad angular.

2.2. Control

Luego de realizada la estimación de las variables de estado del vehículo, se hace necesario el manipularlas según necesitemos a través de actuadores. Los actuadores son todos aquellos dispositivos que ejercen una fuerza y/o torque sobre el vehículo: rotores, propulsores, alerones, elevadores, y un largo etcétera son ejemplos de actuadores.

Existen generalmente dos problemas típicos en la teoría de control. El primero es el problema de regulación, dónde queremos mantener al vehículo en un determinado estado (e.g. vuelo estático a 5 m sobre el suelo, o volando a 5 m/s en dirección norte) a pesar de las perturbaciones existentes (e.g. viento, inestabilidades del motor, ruido en los sensores). El segundo es el problema de seguimiento, en donde queremos mantener el estado del vehículo (e.g. posición y/o velocidad) lo más cercano posible a una señal o trayectoria de referencia (e.g. comandos del piloto, ruta predefinida).

Por lo general el sistema dinámico del vehículo es expresado matemáticamente como

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2)$$

donde $\mathbf{u}$ es el vector de variables de control (e.g. velocidad del motor, ángulo de deflexión del alerón).

Tanto Px4 como la mayoría de los software populares, utilizan las herramientas clásicas de control lineal y hacen uso de bucles PID para resolver el problema de seguimiento. En este caso se linealiza la dinámica del vehículo en torno a un vuelo estático para el caso de un multi-rotores, y un vuelo a velocidad y altura constante para el caso de un avión. Así se obtiene un sistema

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3)$$

donde tanto $\mathbf{A}$ como $\mathbf{B}$ son por lo general matrices con coeficientes constantes pero dependientes del punto de linealización. La derivación matemática de como diseñar un controlador queda para otro ocasión, pero para el caso específico de un multi-rotores el resultado es el de la imagen de más abajo.

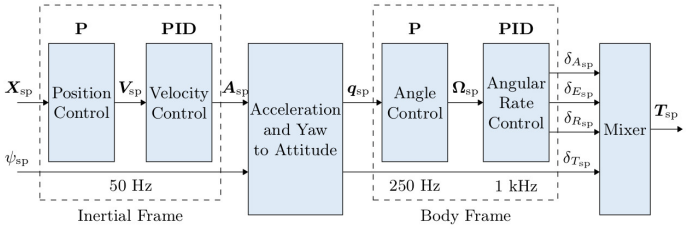


Figura 4: Px4 - Controller Diagrams

Las entradas al diagrama $\mathbf{x}_{sp}$ y ψ_{sp} corresponden a los comandos de desplazamiento en cuatro ejes provenientes del piloto (o piloto automático), la salida $\mathbf{T}_{sp}$ corresponde a las señales de control para cada uno de los motores del vehículo.

La velocidad a la que se ejecuta este bucle es de vital importancia, tanto en lo teórico como en lo práctico. En lo teórico existe una frecuencia por debajo de la cual el sistema de control no podrá estabilizar el vehículo. En la práctica los multi-rotores de peso mayor a 1kg son estables a partir de una frecuencia de control de 50 Hz. Esto es la cantidad de veces por segundo que se envían comandos a los motores. Los controladores de motores más modernos permiten llegar a los 400 Hz y hasta los 1000 Hz, permitiendo vuelos más suaves y precisos.

3. Comunicación

Para todo vehículo aéreo, pero en especial para el caso de vehículos autónomos es de vital importancia el contar con un sistema de comunicaciones que permita conocer el estado del aparato y a la vez enviar ordenes cuando así lo queramos o necesitemos.

En el caso de los vehículos open-source se utilizan radios de distinta potencia y tasa de transmisión. La forma más frecuente de hacerlo es usando radios en la banda de frecuencia amateur (915 Mhz en EEUU y 430 MHz en Europa y Chile). El alcance efectivo suele ser de unos 10 km en linea de vista directa y una tasa de transmisión de unos 100 kbit/s. Sobre esta radio se suele implementar un protocolo de comunicaciones llamado MAVLink, este protocolo permite enviar comandos y recibir telemetría desde una estación terreno

hacia uno o más aparatos en vuelo. Por ultimo está el software de la estación terrena misma. QGroundControl es uno de esos software de estación terrena, esta construido usando el protocolo de comunicaciones MAVLink. Tanto MAVLink como QGroundControl son proyectos apoyados por Dronecode.

En QGroundControl se puede planificar la ruta por la que un vehículo volará, las zonas prohibidas, alturas, velocidad, etc. Una vez en el aire, este programa permite conocer la posición, velocidad, carga de baterías y otros valores de vuelo importantes.

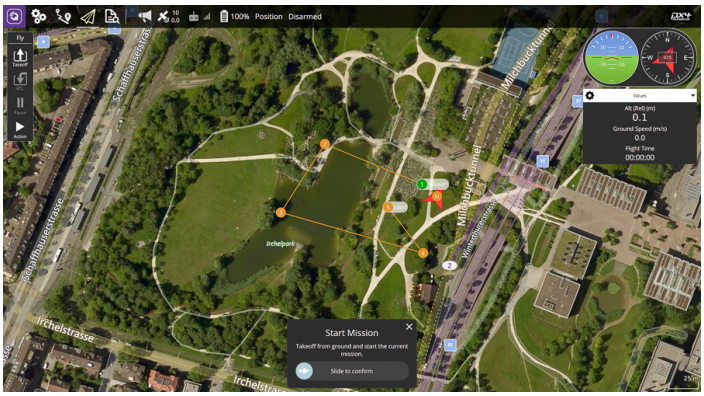


Figura 5: Software de estación terrena QGroundControl

4. Carga útil

Evidentemente un vehículo aéreo que no pueda transportar pasajeros, carga, o algún tipo de sensores y cámaras es de muy baja utilidad. La integración y uso de cámaras y su procesamiento es el aspecto más débil de los proyectos open-source en los que he trabajado; a su vez es uno de los mejor desarrollados por compañías como DJI o Yuneec. Los últimos modelos de compañías comerciales permiten transmitir vídeos en vivo, identificar a otros aparatos volando en el área, evitar colisiones con objetos cercanos, etc. En general todos los últimos avances en procesamiento de video están siendo incorporados.

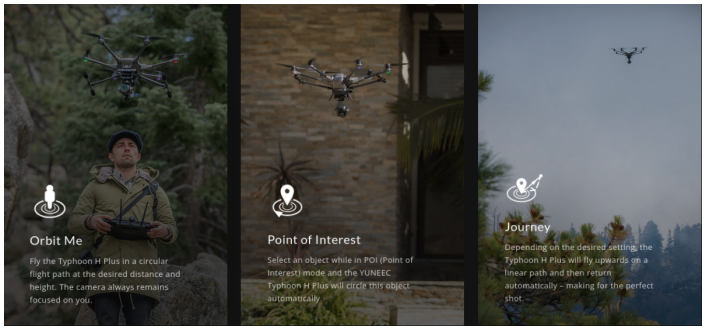


Figura 6: Ejemplos de enfoque de cámara avanzado en un Drone comercial (Yuneec - Typhoon H, Mayo del 2022)

Afortunadamente todo esto también puede ser logrado con tecnología open-soruce, aunque de forma separada. Todo lo que respecta al uso de gimbals y cámaras es posible de adquirir por separado. Existe gente mucho más capacitada que yo para saber que productos son mejores para cada situación por lo que solo me limitaré decir que existen muchas alternativas. En cuento al procesamiento de video, la forma más directa y sencilla en mi opinión es la de usar un computador secundario (companion computer) de alto rendimiento que se encargué del procesamiento de video, mientras el computador de vuelo solo se concentra en la estabilidad y navegación del vehículo. La comunicación entre computadores se puede establecer fácilmente usando MAVLink. Más aún, existen una serie de herramientas de software dedicadas a integrar más fácilmente el computador de vuelo con el computador secundario: MAVSDK o DroneKit.

5. Estructura y actuadores

Las secciones anteriores han discutido diferentes aspectos de como volar un vehículo usando alternativas open-source, pero nada ha sido dicho respecto del aparato en si mismo. ¿Es todo lo anterior aplicable a un avión, un helicóptero o un multi-rotor?. Afortunadamente el nivel actual de la tecnología permite -hasta cierto punto- poder ocupar exactamente el mismo computador de vuelo, el mismo software y los mismos sistemas de comunicaciones para controlar vehículos muy distintos entre sí.

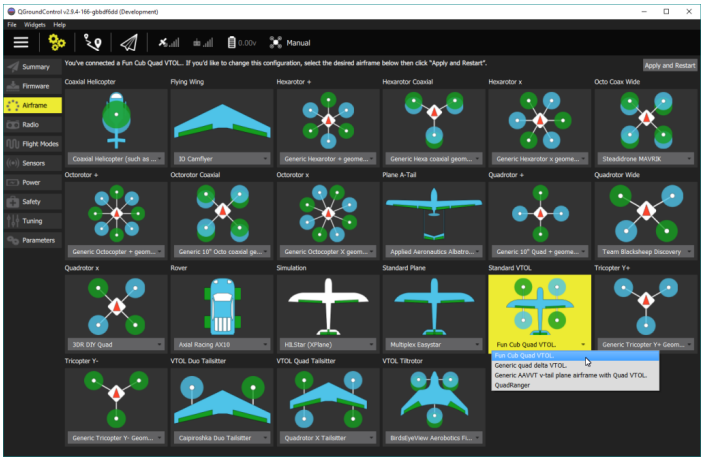
Desde un punto de vista ingenieril, La estructura de un vehículo aéreo soporta y ubica en posición a los actuadores (motores, hélices, superficies de con-

trol, etc). La estructura también da soporte físico a la carga principal del aparato (cámaras, sensores, pasajeros, etc). Además, la estructura también determina -en buena parte- las propiedades inerciales y aerodinámicas del vehículo. En términos simples: cuan fácil o difícil es mover y rotar al aparato (inercia), y que fuerzas y torques se generan al desplazarlo en un fluido (aerodinámica).

Las anteriores razones justifican decir que la forma y tamaño de un vehículo aéreo es una consecuencia del objetivo para el cual se diseñó. Si la función del aparato es realizar grandes desplazamientos de carga, entonces un diseño más cercano a un avión es óptimo. Si, en cambio, es necesario realizar operaciones en estático y a baja altura, entonces un diseño más cercano a un helicóptero o multi-rotor es óptimo.

Volviendo a lo practico, y tal como lo muestra la imagen siguiente, el ya mencionado software Px4 permite volar una gran cantidad de diseños predefinidos. Una lista con modelos comerciales (no necesariamente open-source) se puede ver en los siguientes enlaces

- `https://docs.px4.io/master/en/frames_multicopter/`
- `https://docs.px4.io/master/en/frames_plane/`
- `https://docs.px4.io/master/en/frames_vtol/`



En caso de tener un modelo sui-generis -como el vehículo coaxial Tdrone en la imagen siguiente- es posible incorporarlo, pero esto ya requiere de un mayor grado de conocimiento técnico y de un proceso de pruebas de vuelo que puede llegar a ser extenso.

Hasta el momento no he encontrado algún equivalente a Dronecode para modelos CAD de estructuras y actuadores. En buena parte porque el compartir y luego replicar un motor, una hélice o un engranaje as mucho más difícil que compartir software. Los siguientes enlaces (que no he usado personalmente) pueden servir como premio de consuelo.

- `https://clara.io/library?query=drone&gameCheck=true`
- `https://www.3dcadbrowser.com/3dmodels.aspx?download=drone`
- `https://grabcad.com/library/tag/drone`

Conclusión

En resumen, existe un solido sustrato de tecnologías open-source que permiten volar un gran número de diseños, desde los tradicionales a los más innovadores. Desde un punto de vista del usuario, la parte mejor desarrollada hasta el momento son el computador de vuelo y el software de vuelo, siendo todo lo relacionado con estructura y actuadores la parte menos madura. Diferentes compañías privadas también están avanzando para dotar a estos aparatos de con los últimos avances en visión computacional y automatización. Por ultimo, desde un punto de vista científico, el dotar a los Drones con mayores niveles de autonomía e inteligencia es un área en de investigación activa que se intersecta con la robótica, la biomecánica y la inteligencia artificial.

Tomás I. Opazo