POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es

Miguel Fernández Cortizas

05 TRABAJO FIN DE GRADO

INDUSTRIALES

TRABAJO FIN DE GRADO

KIT DE DESARROLLO Y VALIDACIÓN DE ALGORITMOS DE CONTROL DE ACTITUD PARA CUADRICÓPTEROS.

TRABAJO FIN DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE GRADUADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

SEPTIEMBRE **2019**

Miguel Fernández Cortizas

DIRECTOR DEL TRABAJO FIN DE GRADO:
Pascual Campoy Cervera

INDUSTRIALES
ETSII | UPM

POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

POLITÉCNICA

KIT DE DESARROLLO Y VALIDACIÓN DE ALGORITMOS DE CONTROL DE ACTITUD PARA CUADRICÓPTEROS.

Miguel Fernández Cortizas

Tutor académico:
D. Pascual Campoy Cervera

Madrid - España
2020

En memoria de mi padrino Juan,
seguiré trabajando hasta alcanzar las metas
que me hubiese gustado celebrar contigo.

Agradecimientos

Resumen ejecutivo

Introducción

Los drones de carreras autónomos aun están lejos de alcanzar el rendimiento de los pilotos humanos. añadir todos los campos que tienen relevancia: estimacion, control, generación de trayectorias, percepción y todo a muy altas velocidades con sistemas a bordo

En 2019 la *Drone Racing League* y la empresa *Lockheed Martin* organizaron **Alphapilot** una carrera de drones autónomos con un gran premio de 1 millon de dólares para el ganador.

Alcance

Los objetivos de este trabajo consisten en diseñar un controlador capaz de permitir que un cuadricóptero sea capaz de volar a muy altas velocidades, superiores a 5 m/s, a través de un circuito de carreras para drones, así como generar las trayectorias que el dron debe seguir para recorrer el circuito de forma óptima.

Solución realizada

- **Controlador pequeños ángulos:**
- **Controlador grandes ángulos:**
- **Generador de trayectorias óptimas en acceleracion:**
- **Generador de trayectorias óptimas en *snap*:**

Experimentación y resultados.

Para la simulación se ha empleado el entorno de simulación Flightgoogles citar el cual fue el empleado para las pruebas clasificatorias virtuales del Alphapilot 2019.

Conclusiones y trabajo futuro

Palabras clave

UAV, cuadricóptero, control clásico, aprendizaje automático, inteligencia artificial, aprendizaje por refuerzo, redes neuronales.

Códigos UNESCO

120304 INTELIGENCIA ARTIFICIAL

120326 SIMULACIÓN

330104 AERONAVES

330412 DISPOSITIVOS DE CONTROL

330703 DISEÑO DE CIRCUITOS

Índice general

Introducción

cuidado con usar dron y uav, pueden ser los militares... Los drones son vehículos no tripulados cuya popularidad en el mundo de la industria es cada vez mayor siendo utilizados para realizar tareas en campos muy diversos, como la inspección industrial, la industria cinematográfica o para su uso en operaciones de búsqueda y rescate. Dentro de este grupo los más empleados por la industria son los cuadricópteros, debido a la simplicidad de estos, lo que reduce su peso y su coste.

La inmensa mayoría del uso de estas aeronaves es mediante teleoperación, dentro del paradigma de los RPAS (*Remotely Piloted Aircraft System*) aunque el auge de estas aeronaves han llevado a la comunidad científica hacia el desarrollo de sistemas que permita a los cuadricópteros ser capaces de realizar tareas de forma autónoma.

Una prueba de estos avances se pueden observar en diversas competiciones internacionales como IMAV, MBZIRC o más recientemente el AIRR. Competiciones en las que el objetivo es conseguir que UAV sean capaces de realizar pruebas complejas de forma autónoma.

1.1. Motivación

Las velocidades a las que suelen volar estos uav normalmente no suelen exceder los 3 m/s estando la inmensa mayoría de ellas por debajo del 1.5 m/s. El aumentar la velocidad de vuelo, exige tener algoritmos de percepción y de estimación de estado más precisos y rápidos, así como algoritmos de planificación y de control más rápidos y ligeros.

Además en la competición toda la computación se realiza a bordo de la aeronave en un SBC, por lo que la capacidad de cálculo es limitada.

Generar un sistema autonónomo desde cero presenta una gran complejidad lo que escaparía del alcance de este trabajo, es por eso que en este trabajo se han simplificado el desarrollo de los módulos de estimación de estado y de percepción del circuito, empleando datos provistos por el simulador que permiten que el trabajo se haya centrado en el módulo de control y de generación de trayectorias.

1.2. Solución propuesta

En este trabajo se propone un sistema autónomo capaz de recorrer un circuito de carreras simulado a altas velocidades de forma autónoma y con incertidumbre sobre el recorrido en sí.

Para desarrollar los algoritmos se empleará el simulador fotorrealista FlightGoogles, empleado para las pruebas clasificatorias del AIRR 2019 como entorno de pruebas.

1.3. Objetivos

Estado del arte

Modelado de un cuadricóptero

Un cuadricóptero es un robot aéreo con 6 grados de libertad (3 rotacionales y 3 traslacionales) y 4 motores, al tener menos motores que el número de grados de libertad, se dice que es un sistema subactuado.

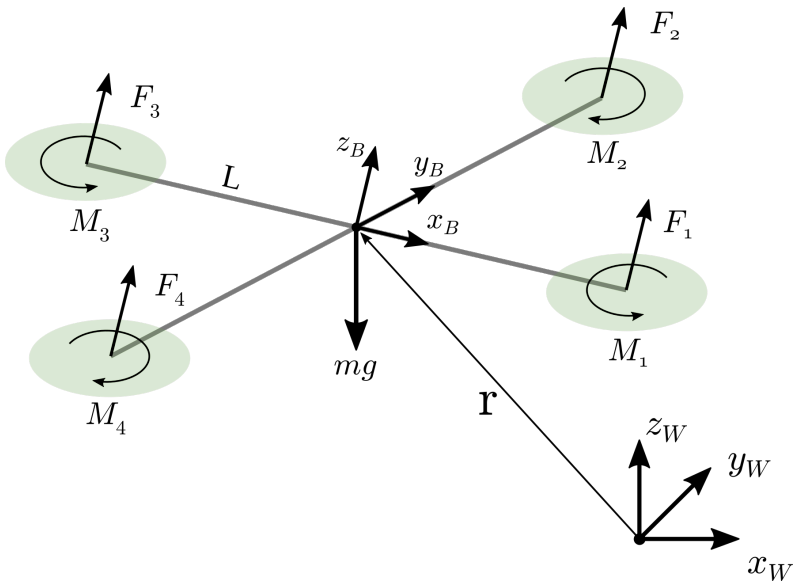

Figura 3.1: Esquema de fuerzas y momentos que actúan sobre un cuadricóptero y sus sistemas de referencia asociados.

Como puede observar en la figura 3.1, se ha empleado el subíndice W para hacer referencia al sistema de referencia del mundo, así como el subíndice B para referirse al sistema asociado al cuerpo del cuadricóptero. El *frame* B posee su origen O_B en el centro de masas de la aeronave, con el eje x_B coincidente con la dirección de avance preferente de la aeronave.

Para modelar las rotaciones del *frame* B con respecto a W se emplearán los ángulos de Euler Z-X-Y, es decir, la matriz de rotación R para transformar coordenadas desde B a W consiste en la composición de las siguientes rotaciones:

$$R = R_{\mathbf{z},\psi} R_{\mathbf{x},\phi} R_{\mathbf{y},\theta} \tag{3.1}$$

siendo $R_{\mathbf{i},\alpha}$ una rotación de un ángulo α respecto al eje i. Al desarrollar la expresión 3.1 se obtiene la matriz R resultante

$$R = \begin{bmatrix} c_\psi c_\theta - s_\phi s_\psi s_\theta & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ c_\theta s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi & s_\psi s_\theta - c_\theta s_\phi c_\psi \\ -c_\phi s_\theta & s_\phi & c_\phi c_\theta \end{bmatrix} \tag{3.2}$$

donde c_θ y s_θ denotan $cos(\theta)$ y $sen(\theta)$ respectivamente.

En el sistema de referencia B las componentes del vector velocidad angular Ω están definidas por p, q y r de la forma:

$$\Omega = p\mathbf{x_B} + q\mathbf{y_B} + r\mathbf{z_B} \tag{3.3}$$

Estas componentes están relacionadas con las derivadas de los ángulos de Euler de acuerdo a

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & -c_\phi s_\theta \\ 0 & 1 & s_\phi \\ s_\theta & 0 & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.4}$$

3.1. Análisis dinámico

3.1.1. Ecuaciones de movimiento traslacional

Como se puede observar en la figura 3.1, las fuerzas que actúan sobre el cuadricóptero son: la gravedad, en la dirección $-\mathbf{z_W}$ y la fuerza de cada uno de los motores, en la dirección $\mathbf{z_B}$. Para hallar las ecuaciones que rigen la dinámica del centro de masas del sistema C se aplican las ecuaciones de Newton sobre él. Siendo $\mathbf{r}$ el vector de posición del centro de masas C con respecto al origen de W obtenemos:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \tag{3.5}$$

Si denominamos $F = \sum_{i=1}^{4} F_i$, al expandir la ecuación anterior con la definición de R en 3.2 obtenemos las ecuaciones que describen el movimiento traslacional del centro de masas del cuadricóptero:

$$m\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} s_\theta c_\psi + s_\phi c_\theta s_\psi \\ s_\theta s_\psi - s_\phi c_\theta c_\psi \\ c_\phi c_\theta \end{bmatrix} F \tag{3.6}$$

3.1.2. Ecuaciones de movimiento rotacional

Como se puede observar en la expresión anterior, el movimiento del cuadricóptero depende de la rotación R, por lo que es necesario modelar el movimiento rotacional del mismo. Se define el momento angular H como:

$$H = \mathbf{I}\Omega \tag{3.7}$$

donde $\mathbf{I} \in \mathbb{R}^{3\times3}$ representa el tensor de inercia del cuadricóptero en el sistema B, y $\Omega = [p, q, r]$ representa el vector de velocidad angular de B respecto W.

Si denotamos $M_c = [\tau_x, \tau_y, \tau_z]^t$ como el momento total del cuadricóptero en el sistema B

$$\begin{aligned} M_c &= \frac{d}{dt}H \\ &= \mathbf{I}\dot{\Omega} + \Omega \times \mathbf{I}\Omega \end{aligned} \tag{3.8}$$

Se considera que el cuadricóptero presenta una distribución de masa simétrica, por lo que el tensor de inercia $\mathbf{I}$ es un tensor diagonal de la forma:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.9}$$

siendo I_{xx}, I_{yy}, I_{zz} los momentos principales del cuadrícoptero con respecto a los ejes x_B, y_B, z_B respectivamente.

Desarrollando la expresión 3.8 y reorganizando sus términos:

$$\mathbf{I}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.10}$$

A partir del diagrama de la figura 3.1 se puede calcular los valores de τ_x, τ_y y τ_z a partir de las fuerzas y momentos ejercidos por los motores.

$$\begin{aligned} \tau_x &= L(F_2 - F_4) \\ \tau_y &= L(F_3 - F_1) \\ \tau_z &= M_1 - M_2 + M_3 - M_4 \end{aligned} \tag{3.11}$$

Finalmente se unen las expresiones 3.10 y 3.11 para obtener la ecuación de movimiento rotacional del cuadricóptero.

$$\mathbf{I}\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.12}$$

3.2. Differential flatness

Control

En este capítulo se desarrollará la teoría general para el control de un cuadricóptero Citar artículos a cholón, la cual se adaptará posteriormente al caso particular de la arquitectura empleada.

El problema del control se puede expresar formalmente: Dado el estado del sistema $x(t)$ continuar . ... como encontrar la función $u(t)$ tal que, el estado $x(t)$ sigue la trayectoria desesada $x^{des}(t)$ a lo largo del tiempo.
Si se define el error $e(t)$ del control como:

$$e(t) = x^{des}(t) - x(t) \tag{4.1}$$

el objetivo del controlador sería conseguir que el error $e(t)$ converja de forma exponencial a 0.

El estado $x(t)$ de un cuadricóptero consta de 12 variables, las 6 correspondientes a su pose y sus derivadas correspondientes.

$$x(t) = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^t \tag{4.2}$$

Si llamamos q al vector de configuración formado por las 6 primeras variables:

$$q(t) = [x \;\; y \;\; z \;\; \phi \;\; \theta \;\; \psi \;]^t \tag{4.3}$$

$$x(t) = [q \;\; \dot{q}]^t \tag{4.4}$$

Para el control de la aeronave se han definido dos señales de entrada $\mathbf{u_1}$ y $\mathbf{u_2}$ siendo

$$\mathbf{u_1} = F = F_1 + F_2 + F_3 + F_4 \tag{4.5}$$

$$\mathbf{u_2} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} \tag{4.6}$$

como se puede observar la señal de control u_1 actúa sobre el empuje total de la aeronave en el eje z_B mientras la señal u_2 controla los momentos totales aplicados sobre el cuadricóptero. Los pares M_i que ejerce cada motor se relacionan con F_i de la forma $M_i = \gamma F_i$, siendo γ una constante aerodinámica dependiente de las hélices empleadas. De esta forma se puede expresar u_2 de la siguiente forma:

$$\mathbf{u_2} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{4.7}$$

La arquitectura de control general consiste en dos controladores anidados: un controlador de posición y un controlador de orientación, como se puede observar en la figura .incluir diagrama

4.1. Controlador para ángulos pequeños

Cuando las trayectorias que el cuadricóptero son poco agresivas se puede linealizar el modelo del cuadricóptero entorno a su punto de equilibrio. En la situación de equilibrio el cuadricóptero se encuentra en *hover*, es decir manteniendo la posición en el aire . En *hover* el estado x_0 de la aeronave es de la forma

$$\begin{aligned} x_0 &= [q_0 \;\; 0]^t \\ q_0 &= [x_0 \;\; y_0 \;\; z_0 \;\; 0 \;\; 0 \;\; \psi_0 \;]^t \end{aligned} \tag{4.8}$$

Como los ángulos ϕ y θ son pequeños se puede aproximar $cos(\phi) \approx 1, cos(\theta) \approx 1$ y $sen(\phi) \approx \phi, sen(\theta) \approx \theta$. En este estado de equilibrio la fuerza total que ejercen los motores debe ser igual al peso del cuadricóptero por lo que:

$$F_{i,0} = \frac{mg}{4} \quad , \quad \mathbf{u_{1,0}} = mg \tag{4.9}$$

Al desglosar y linealizar las ecuaciones 3.6 entorno a este estado obtenemos las ecuaciones linealizadas del movimiento traslacional:

$$\begin{aligned} \ddot{x} &= g(\Delta\theta \, cos\psi_0 + \Delta\phi \, sen\psi_0) \\ \ddot{y} &= g(\Delta\theta \, sen\psi_0 - \Delta\phi \, cos\psi_0) \\ \ddot{z} &= \frac{1}{m}\mathbf{u_1} - g \end{aligned} \tag{4.10}$$

Por otro lado, para obtener las ecuaciones que permiten controlar la rotación del cuadricóptero debemos desarrollar las ecuaciones 3.12. Debido a la simetría del cuadricóptero se considera que $I_{xx} \approx I_{yy}$, con lo que se obtienen las ecuaciones del movimiento rotacional:

$$\begin{aligned} I_{xx}\dot{p} &= u_2 - qr(I_{zz} - I_{yy}) \\ I_{yy}\dot{q} &= u_3 - pr(I_{xx} - I_{zz}) \\ I_{zz}\dot{r} &= u_4 \end{aligned} \tag{4.11}$$

Se puede asumir que la componente r es pequeña por lo que los productos de esta con otros términos, también son pequeños en comparación con el resto de términos. Además, en el estado de *hover* $\dot{\phi} \approx p$, $\dot{\theta} \approx q$ y $\dot{\psi} \approx r$. Es por esto que la expresión anterior se puede simplificar por:

$$\begin{aligned} \dot{p} &= \frac{u_2}{I_{xx}} \\ \dot{q} &= \frac{u_3}{I_{yy}} \\ \dot{r} &= \frac{u_4}{I_{zz}} \end{aligned} \tag{4.12}$$

Gracias a esto es posible expresar el valor de las acciones de control $\mathbf{u_2}$ como:

$$u_{2,des} = k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p) \tag{4.13}$$
$$u_{3,des} = k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q) \tag{4.14}$$
$$u_{4,des} = k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r) \tag{4.15}$$

En el estado de hover,las velocidades de roll y pitch son cero por lo que $p_{des} = 0$, $q_{des} = 0$.

4.1.1. Controlador de posición

En cuanto al tipo de controlador se ha decidido emplear un controlador PD. En un controlador PD el objetivo de seguir la trayectoria r_T deseada es conseguir que la expresión 4.16 tienda a 0 en un tiempo exponencial:

$$(\ddot{r_T} - \ddot{r_c}) + K_d e_v + K_p e_p = 0 \tag{4.16}$$

siendo $e_v = (\dot{r}_T - \dot{r}_c)$ y $e_p = (r_T - r_c)$. Las constantes $K_d, K_p \in \mathbb{R}^+$ del controlador definen la dinámica y la estabilidad del sistema.

Desarrollando 4.16:

$$\ddot{r_c} = \ddot{r_T} + K_d e_v + K_p e_p \tag{4.17}$$

Al combinar 4.10 con 4.17:

$$\begin{bmatrix} g(\Delta\theta\, cos\psi_0 + \Delta\phi\, sen\psi_0) \\ g(\Delta\theta\, sen\psi_0 - \Delta\phi\, cos\psi_0) \\ \frac{1}{m}\mathbf{u_1} - g \end{bmatrix} = \ddot{r_T} + K_d e_v + K_p e_p \tag{4.18}$$

por lo que :

$$\mathbf{u_1} = m\,(g + \ddot{z_{des}} + K_{d,z}(\dot{z_{des}} - \dot{z}) + K_{p,z}(z_{des} - z)) \tag{4.19}$$
$$\phi^{des} = \frac{1}{g}\,(\ddot{x}_{des} sin\,\psi_T - \ddot{y}_{des} cos\,\psi_T) \tag{4.20}$$
$$\theta^{des} = \frac{1}{g}\,(\ddot{x}_{des} cos\,\psi_T + \ddot{y}_{des} sin\,\psi_T) \tag{4.21}$$

Como el angulo de *yaw* $\psi_T(t)$ se incluye en la trayectoria definida, entonces:

$$\psi_{des} = \psi_T(t) \tag{4.22}$$
$$r_{des} = \dot{\psi}_T(t) \tag{4.23}$$

completar un poco más

4.2. Controlador para ángulos grandes

A continuación se presenta un controlador que no parte de la premisa de que el estado de la aeronave es cercano al de equilibrio y que permite seguir trayectorias $\sigma_T = [\mathbf{r}_T(t)^t, \psi_T(t)]^t$.

El objetivo de este controlador es orientar el vector z_B de la aeronave con la dirección de la fuerza necesaria $\mathbf{F}_{des}$ para poder seguir la trayectoria. imagen

Para ello primero definiremos los errores en posición y velocidad:

$$\mathbf{e}_p = \mathbf{r} - \mathbf{r}_T \tag{4.24}$$

$$\mathbf{e}_v = \dot{\mathbf{r}} - \dot{\mathbf{r}}_T \tag{4.25}$$

La fuerza deseada $\mathbf{F}_{des}$ en cada instante de tiempo viene dada por la expresión

$$\mathbf{F}_{des} = -K_p\mathbf{e}_p - K_v\mathbf{e}_v - mg\mathbf{z}_W + m\ddot{\mathbf{r}}_T \tag{4.26}$$

donde K_p y K_v son matrices definidas positivas. Como se puede observar, la fuerza debe tener en cuenta el peso de la aeronave, la aceleración impuesta por la trayectoria en ese instante y el error de seguimiento de la trayectoria en ese instante para poder corregirlo.

Al proyectar $\mathbf{F}_{des}$ sobre el eje x_B del cuadricóptero se puede obtener u_1, por lo que:

$$u_1, des = \mathbf{F}_{des} \cdot \mathbf{z}_B \tag{4.27}$$

Las otras tres señales de control del sistema dependen de la orientación de la aeronave. El objetivo es alinear el eje $\mathbf{z}_B$ de la aeronave con la dirección de la fuerza deseada $\mathbf{z}_{B,des}$, es decir, que la matriz de rotacion R_{des} viene dada por:

$$R_{des}\mathbf{e_3} = \mathbf{z}_{B,des}\ , \quad \mathbf{z}_{B,des} = \frac{\mathbf{F}_{des}}{||\mathbf{F}_{des}||} \tag{4.28}$$

siendo $\mathbf{e_3}$ el vector director del eje z_B de la aeronave respecto al sistema B, es decir, $\mathbf{e_3} = [0, 0, 1]^t$. Debido a que el ángulo de yaw ψ_T viene dado por la trayectoria se pueden calcular los vectores de rotación deseados de la forma:

$$\mathbf{x}_{C,des} = [cos\,\psi_T, sin\psi_T, 0]^t \tag{4.29}$$

$$\mathbf{y}_{B,des} = \frac{\mathbf{z}_{B,des} \times \mathbf{x}_{C,des}}{||\mathbf{z}_{B,des} \times \mathbf{x}_{C,des}||}\ ,\ \mathbf{x}_{B,des} = \mathbf{y}_{B,des} \times \mathbf{z}_{B,des} \tag{4.30}$$

por lo que la matriz $R_{des} \in SO(3)$ se expresa como:

$$R_{des} = \begin{bmatrix} | & | & | \\ \mathbf{x}_{B,des} & \mathbf{y}_{B,des} & \mathbf{z}_{B,des} \\ | & | & | \end{bmatrix} \tag{4.31}$$

A continuación definiremos el error en orientación como:

$$\mathbf{e}_R = \frac{1}{2}\left(R_{des}^t R - R^t R_{des}\right)^\vee \tag{4.32}$$

donde $\vee$ denota el vee map que transforma elementos de $SO(3)$ a $\mathbb{R}$. El error de velocidad angular es simplemente la diferencia entre la velocidad angular de la aeronave y la velocidad angular deseada, ambas medidas con respecto las coordenadas del sistema de coordenadas del cuadricóptero B.

$$\mathbf{e}_\omega = {}^B[\omega] - {}^B[\omega_T] \tag{4.33}$$

Finalmente, las tres señales de salida restantes, correspondientes con los pares ejercidos sobre la aeronave, se calculan como:

$$\begin{bmatrix} u_{2,des} \\ u_{3,des} \\ u_{4,des} \end{bmatrix} = -K_R \mathbf{e}_R - K_\omega \mathbf{e}_\omega \tag{4.34}$$

Este controlador presenta dos ventajas considerables con respecto al controlador anterior:

1. El error en orientación está expresado mediante matrices de rotación y no mediante ángulos de Euler, los cuales presentan una singularidad en torno a $\theta = 90^o$ conocida como *gimbal lock*.

2. La fuerza deseada se proyecta sobre el vector z_B de la aeronave en el instante actual, en lugar de considerar que éste siempre apunta verticalmente, por lo que mejora notablemente el control en trajectorias agresivas, donde la aeronave presenta una gran inclinación.

hablar de la estabilidad y robustez del sistema

4.3. Adaptación del controlador al entorno simulado.

En los controladores anteriores las señales de salida están directamente relacionadas con la velocidad de giro de cada motor y de la fuerza de empuje que genera cada uno, como se puede observar en las expresiones 4.5 y 4.7. Sin embargo, en la competición, no se tiene el control de cada motor de forma independientemente, si no que hay un controlador interno que controla los motores. Es por esto que, es necesario adaptar los controladores del estado del arte mostrados anteriormente al nuevo conjunto de señales de entrada disponibles.

diagrama nuevo

El controlador interno del cuadricóptero tiene como consignas a seguir las velocidades angulares del cuadrícoptero deseadas, así como el empuje total que deben proporcionar los motores. Estas son las señales que envían los pilotos de carreras de drones profesionales a través de sus mandos, debido al gran control que tienen de la aeronave. Al poder controlar el empuje total de la aeronave, la señal u_1 se mantiene igual que en los controladores anteriores, sin embargo es necesario realizar algunas pequeñas modificaciones en los controladores angulares. Las otras 3 señales de control que se van a emplear son las referencias del controlador interno p_{ref},q_{ref} y r_{ref}.

Modificaciones para ángulos pequeños

Como se puede observar en la expresión 4.13, el controlador angular es un control PD que depende de la orientación actual de la aeronave y de la orientación deseada. Dado que el objetivo es minimizar ese error de orientación, podemos expresar el controlador angular como:

$$u_2 = p_{ref} = p_{des} + k_{p,\phi}(\phi^{des} - \phi) \tag{4.35}$$

$$u_3 = q_{ref} = q_{des} + k_{p,\theta}(\theta^{des} - \theta) \tag{4.36}$$

$$u_4 = r_{ref} = r_{des} + k_{p,\psi}(\psi^{des} - \psi) \tag{4.37}$$

Modificaciones para ángulos grandes

Para el controlador para grandes ángulos se ha aplicado una estrategia similiar a la anterior. La expresión 4.34 se ha modificado para que sea de la forma:

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = {}^{B}[\omega] = {}^{B}[\omega_T] - K_R \mathbf{e}_R \tag{4.38}$$

Con estas pequeñas modificaciones se ha conseguido adaptar los controladores del estado del arte al interfaz provisto por el entorno de simulación que se empleará posteriormente en los experimentos.

Generación de trayectorias

En el apartado anterior se ha diseñado un controlador cuyo objetivo es seguir una trayectoria $x^{des}(t)$ minimizando el error de seguimiento. En este apartado se tratará sobre la forma en la que se generan estas trayectorias. Este desarrollo se realiza entorno a una trayectoria unidimensional en un único eje de desplazamiento , el eje x. Este desarrollo es ampliable al espacio tridimensional realizando una trayectoria por cada eje.

5.1. Trayectorias óptimas

El objetivo general del control óptimo es encontrar la función $x^*(t)$ que minimiza la expresión

$$x^*(t) = \underset{x(t)}{argmin} \int_0^T \mathcal{L}\left(x^{(n)}, x^{(n-1)}, ..., \dot{x}, x, t\right)\, dt \tag{5.1}$$

siendo $\mathcal{L}$ el índice que se debe optimizar.

Cuando el objetivo es generar trayectorias suaves la forma de la expresión a optimizar es :

$$x^*(t) = \underset{x(t)}{argmin} \int_0^T \left(x^{(n)}\right)^2\, dt \tag{5.2}$$

siendo $x^{(n)} = u$, la magnitud de control que se desea minimizar. El valor del parámetro $n \in \mathbb{N}$, expresa el grado de la derivada de la acción de control u, es decir, para generar trayectorias de mínima distancia, se empleará un valor de $n = 1$, por lo que $u = \dot{x}$, mientras que para generar trayectorias de mínima sobreaceleración (*jerk*) se empleara un valor de $n = 3$, por lo que $u = \dddot{x}$.

De forma general la expresión a optimizar será:

$$x^*(t) = \underset{x(t)}{argmin} \int_0^T \mathcal{L}\left(x^{(n)}, x^{(n-1)}, ..., \dot{x}, x, t\right)\, dt \tag{5.3}$$

De la ecuación de Euler-Lagrange se obtiene que la la función optima $x^*(t)$ debe satisfacer:

$$\frac{\partial \mathcal{L}}{\partial x} + \sum_{i=1}^{n} (-1)^i \frac{d^i}{dt^i}\left(\frac{\partial \mathcal{L}}{\partial x^i}\right) = 0 \tag{5.4}$$

Si particularizamos la expresión anterior a expresiones donde $\mathcal{L} = \left(x^{(n)}\right)^2$, todos las

derivadas parciales $\frac{\partial \mathcal{L}}{\partial x^i} = 0 \; ; \forall i \neq n$ por lo que:

$$\frac{d^n}{dt^n}\left(\frac{\partial \mathcal{L}}{\partial x^n}\right) = 0 \tag{5.5}$$

$$\frac{d^n}{dt^n}\left(2x^{(n)}\right) = 0 \tag{5.6}$$

$$x^{(2n)} = 0 \tag{5.7}$$

Al integrar la expresión anterior se obtiene el polinomio de grado $2n-1$, con $2n$ coeficientes:

$$x(t) = c_{2n-1}t^{2n-1} + c_{2n-2}t^{2n-2} + ... + c_2t^2 + c_1t + c_0 \tag{5.8}$$

Para resolver los coeficientes, son necesarias $2n$ condiciones de contorno que corresponden con los valores de la función $x(t)$ y de sus derivadas en los instantes $t = 0$ y $t = T$. Si se considera que la trayectoria transcurre entre un estado inicial y un estado final, ambos estados de equilibrio , se puede considerar que las derivadas son nulas por lo que las condiciones de contorno serían:

$$x(0) = a; \quad \dot{x}(0) = 0; \quad ... \quad x^{(2n-1)}(0) = 0 \tag{5.9}$$

$$x(T) = b; \quad \dot{x}(T) = 0; \quad ... \quad x^{(2n-1)}(T) = 0 \tag{5.10}$$

Con estas condiciones de contorno se calcula la trayectoria óptima para pasar de una posición inicial a a una posición final b en un tiempo T minimizando la acción de control $u = x^{(n)}$.

5.2. Trayectorias continuas a trozos

Para generar trayectorias que el cuadricóptero deba seguir para completar el circuito es conveniente contar con puntos de paso (*waypoints*) por los que se quiere que pase la aeronave, por ejemplo, el centro de las puertas.

dibujito guachi

Como se puede observar en el apartado anterior, las trayectorias que se generan entre dos puntos de paso a y b, son trayectorias polinómicas cuyo grado depende del orden de la derivada de la acción de control. Debido a esto, se ha decidido usar trayectorias de tipo *spline*, es decir, curvas diferenciables definidas en segmentos polinómicos, donde cada segmento sería una trayectoria entre dos *waypoints*. Cada trayectoria completa, $r_d(t)$ está compuesta por m segmentos de grado q, presentando la estructura

$$r_d(t) = \begin{cases} \sum_{i=0}^{q} a_{i1}t^i & , t_0 \leq x \leq t_1 \\ \sum_{i=0}^{q} a_{i2}t^i & , t_1 \leq x \leq t_2 \\ \vdots & \vdots \\ \sum_{i=0}^{q} a_{im}t^i & , t_{m-1} \leq x \leq t_m \end{cases} \tag{5.11}$$

siendo $a_{ij} \in \mathbb{R}$ los coeficientes de los polinomios. El grado q de estos polinomios dependerá del índice a minimizar durante el transcurso de la trayectoria, como se ha explicado en el apartado 5.1. Cada tramo tiene un tiempo inicial y un tiempo final, estos tiempos dependen de las limitaciones físicas de la aeronave, es decir, de su velocidad y

aceleración máximas, en el apartado NEW APARTADO se profundiza en la forma de generar estos tramos temporales.

De cara a resolver el valor de los distintos coeficientes de cada segmento polinomial se emplearán unas condiciones de contorno que garanticen la derivabilidad de la trayectoria completa en las q primeras derivadas.

Si se considera una trayectoria completa a través de n *waypoints* P distribuidos a lo largo del tiempo, siendo t_0 el tiempo inicial del primer segmento y t_n el tiempo final de la trayectoria, la trayectoria contará con $n-1$ segmentos polinómicos S de grado q. Considerando que en el punto inicial y el punto final de la trayectoria el cuadricóptero se encuentra en un estado de equilibrio, las condiciones de contorno empleadas para calcular los coeficientes de la *spline* son:

$$\begin{array}{ll} S_i(t=t_{i-1}) = W_{i-1} & \forall i \in 1,...,n \\ S_i(t=t_i) = W_i & \forall i \in 1,...,n \\ S_1^{(k)}(t=t_0) = 0 & \forall k \in 1,...,q \\ S_n^{(k)}(t=t_n) = 0 & \forall k \in 1,...,q \\ S_{i-1}^{(k)}(t=t_{i-1}) = S_i^{(k)}(t=t_i) & \forall k \in 1,...,q\,; \forall n \in 2,...,n \end{array} \tag{5.12}$$

siendo W_i la posición del i-ésimo *waypoint* , S_i el i-ésimo segmento polinómico y t_i el tiempo de finalización del segmento S_i.

Las dos primeras condiciones de contorno hacen referencia a las posiciones espaciales de los waypoints, es decir, que cada segmento empiece en un waypoint y termine en el siguiente. Las dos condiciones siguientes hacen referencia al valor de las derivadas en los estados iniciales y final, se ha considerado que los estado inicial y final, son estados de equilibrio por lo que el valor de sus derivadas es nulo. Finalmente la ultima condición hace referencia a la continuidad de las derivadas entre segmentos, es decir, que el valor de las derivadas del final de un segmento, sean iguales al las derivadas iniciales del segmento siguiente, de esta forma se consigue que la trayectoria completa generada cumpla la condición de derivabilidad a lo largo de todas sus q derivadas.

5.3. Estimación de tiempos

El objetivo final es generar trayectorias a través del circuito de forma que la aeronave sea capaz de recorrerlas teniendo en cuenta restricciones de posición, velocidad y aceleración. En los dos apartados anteriores se explica como generar trayectorias óptimas teniendo en cuentas restricciones de posición, fijando los *waypoints* por los que se desea que pase el cuadricóptero. Sin embargo, para poder generar trayectorias es también necesario elegir los segmentos temporales T_i entre los distintos puntos de paso, es decir, los tiempos que debe tardar la aeronave en pasar entre los distintos waypoints. Estos segmentos de tiempo deben tener en cuenta las limitaciones de velocidad y acceleración máximas de la aeronave, para poder generar trayectorias realizables. A continuación se tratará un posible método para establecer la duración de estos intervalos temporales.

Aproximación al tiempo medio

La forma más sencilla de establecer intervalos temporales es basarse en la distancia existente entre un par de waypoints y la velocidad media a la que se desea que vuele la

aeronave para estimar el tiempo necesario. Siendo $d_i = ||W_{i-1}, W_i||_2$ la distancia entre dos *waypoints* consecutivos y v_m la velocidad media de vuelo, se puede establecer la duración de un segmento S_i como

$$T_i = \frac{d_i}{v_m} \tag{5.13}$$

Para obtener los valores de tiempo absoluto t_i requeridos en el apartado anterior simplemente se deben sumar las duraciones de los segmentos anteriores.

$$t_i = \sum_{j=1}^{i} T_j \tag{5.14}$$

Para tener aumentar el control sobre la agresividad de la trayectoria se puede añadir un coeficiente α_i que permita modificar ligeramente el valor T_i siendo

$$T_i = \alpha_i \frac{d_i}{v_m} \tag{5.15}$$

Valores de α_i pequeños generan trayectorias más agresivas, así como valores más grandes producen trayectorias más seguras. Iterando el valor de este parámetro α_i se controla que las limitaciones en velocidad y aceleración máxima se cumplan para cada segmento S_i.

Esta es una forma simple y rápida de establecer los distintos segmentos temporales, sin embargo, es una aproximación que afecta a la generación de la trayectoria óptima, haciendo que el tiempo total que se tarda en recorrer el circuito sea mayor que el que se podría obtener usando otros métodos de optimización como los propuestos por añadir citaciones para la obtención de estos segmentos temporales. A pesar de esto, debido a que las posiciones de los *waypoints* cambian rápidamente a lo largo del circuito y se deben recalcular las trayectorias continuamente, el bajo coste computacional que presenta este método hace que merezca la pena su utilización.

Metodología

En los capítulos anteriores se ha presentado los distintos algoritmos de control y generación de trayectorias que se emplearñan para conseguir recorrer un circuito con un dron de carreras autónomo . En este apartado se presenta la metodología empleada para superar las pruebas clasificatorias del Alphapilot 2019

6.1. Sistemas de referencia

6.2. Generación de *waypoints*

Para recorrer el circuito de forma satisfactoria es necesario que la aeronave atraviese las distintas puertas o *gates* que componen el circuito en un orden concreto. Para conseguir esto es necesario conocer las posiciones de las puertas en el mundo y generar los puntos de paso necesarios para que la aeronave pase a través de ellas sin colisionar.

En la competición se proporciona el orden en el que se deben atravesar las puertas y una posición aproximada de las posiciones de cada una de ellas en el mundo.

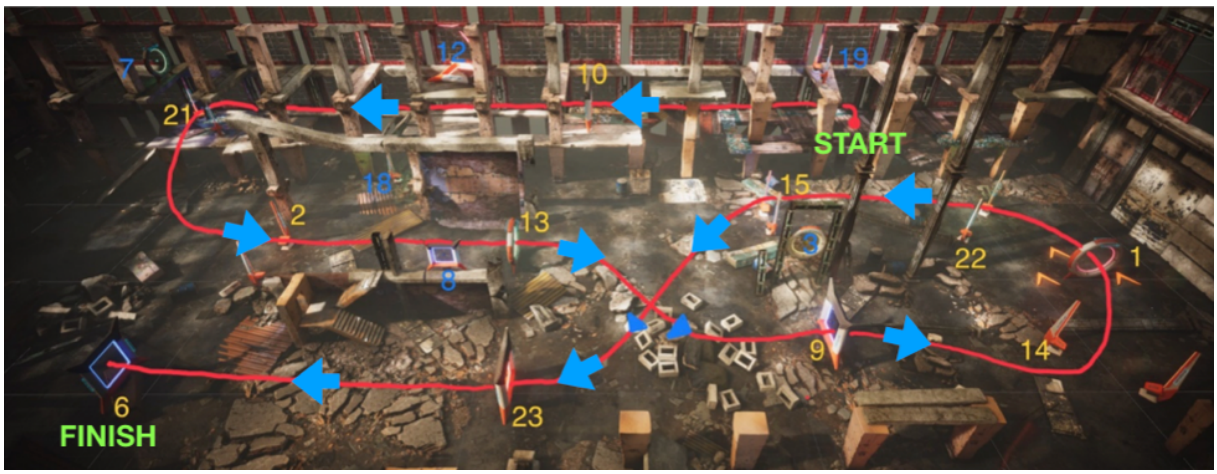

Figura 6.1: Vista aérea del circuito en el simulador FightGoggles, las puertas que se deben traspasar se simbolizan con su número en color amarillo.

Como se puede observar en la figura 6.1 la aeronave debe recorrer 11 puertas, cada una con un número de identificación, en el orden indicado.

6.3. Trayectorias a largo y corto plazo

Arquitectura del sistema

Para conseguir que los distintos módulos que integran el sistema trabajen de forma conjunta es necesario establecer una arquitectura que permita la comunicación eficiente entre ellos así. Para coordinar el trabajo de los distintos módulos que componen el sistema se ha empleado ROS (*Robot Operating System*) [1] un *framework* orientado a el desarrollo de software para robots ampliamente extendido en la comunidad robótica. En el paradigma de ROS el código se estructura en nodos independientes que se comunican entre ellos a través de tópicos (mensajes que un nodo difunde de forma general a los demás nodos y cualquiera lo puede leer) y servicios (peticiones de un nodo particular a otro).

Esto permite desarrollar cada componente del sistema de forma independiente y comunicarlos entre ellos mediante una interfaz común. Esto permite encapsular el código, lo que aumenta la reusabilidad y la robustez de cada módulo, independiente del resto de módulos que les rodeen.

A continuación se explicará más detalladamente los distintos nodos que componen la arquitectura:

- **Simulador:** En la arquitectura propuesta el entorno de simulación empleado, Flightgoogles, se comporta como un nodo adicional. Este nodo constituye el interfaz de comunicación entre la aeronave simulada y el entorno simulado, el cual publica datos sobre el estado de la aeronave como su posición y orientación , las imágenes de las cámaras simuladas y la posición de las esquinas de las puertas en las imágenes obtenidas. Asimismo, recibe los comandos de control enviados a la aeronave.

- **Estimación de estado.** Este modulo se encarga de completar la estimación de estado de la aeronave, empleando la información temporal sobre los cambios de pose de la aeronave. Además proporciona esta información a través de mensajes estándar al resto de los módulos de la arquitectura.

- **Percepción.** El simulador provee las imágenes, tomadas por las cámaras, del entorno simulado. En estas imágenes aparecen las puertas del circuitos con un indicador del número de puerta, así como las posiciones de las esquinas de las puertas en las imágenes tomadas. Conociendo los parámetros de la cámara y las dimensiones de las puertas se puede extrapolar la posición de las puertas respecto a la cámara. El modulo de percepción se encarga de enviar las posiciones aproximadas de las distintas puertas del circuito e ir actualizando estas estimaciones a medida que la aeronave avanza a través del circuito.

- **Generador de trayectorias.** Una vez conocida la posición de la aeronave en el circuito y las posiciones aproximadas de las puertas con respecto a la aeronave se genera la trayectoria que debe seguir la aeronave para pasar a través de las puertas

de la forma más rápida posible sin colisionar con el entorno. Para poder actualizar la trayectoria de forma rápida con un coste bajo computacional se divide este trabajo en dos módulos:

- **Trayectoria a largo plazo**: Obtiene las posiciones de las puertas provistas por el módulo de percepción y genera el recorrido tridimensional completo que debería realizar el cuadricóptero.
- **Trayectoria a corto plazo**: Recibe la trayectoria a largo plazo y la evalúa en un corto horizonte temporal respecto a la posición actual de la aeronave. Dentro de este horizonte se genera la trayectoria de control óptima a seguir. Este módulo también se encarga de evaluar la trayectoria actual a lo largo del tiempo y enviar las consignas de posición, velocidad y aceleración al controlador.

- **Controlador:** El módulo del controlador recibe el estado actual de la aeronave provisto por el modulo de estimación y la referencias provistas por el módulo trayectoria a corto plazo. Con esto genera las acciones de control que se envían al módulo del simulador.

Experimentos

8.1. Experimentos en simulación

8.2. Experimentos en real

Conclusiones y trabajo futuro

9.1. Conclusiones

9.2. Trabajo futuro

Presupuesto y Planificación

A.1. Presupuesto

El presupuesto del trabajo se puede separar en tres partes: recursos humanos, compra de material y amortización de los equipos utilizados.

En cuanto a los recursos humanos empleados, se ha tenido una dedicación por parte del alumno de unas 800 horas, esto es un número de horas mucho superior a las 360 horas (30h/ECTS) correspondientes a la carga temporal de los 12 ECTS del Trabajo fin de Grado (TFG). Esto se ha debido al gran alcance y a la complejidad del mismo. Un sueldo de investigador a media jornada en la universidad, sin estar graduado, es de unos 450 euros. Lo que se traduce en un salario de unos 5,625 euros la hora. Los salarios del tutor y el cotutor se han extraído del portal de transparencia de la UPM. La dedicación del tutor ha sido de unas 20 horas de implicación en el trabajo y la implicación del cotutor ha sido de unas 80 horas de implicación.

Recursos humanos	Horas	Coste Horario [EUR]	Total [EUR]
Alumno	800	5.625	4500
Cotutor	80	7.8	624
Tutor	20	33.72	674.4
Total			**5798.4**

Los costes de material del proyecto son debidos a la construcción del cuadricóptero y del autopiloto.

Material	Coste unitario [EUR]	Unidades	Total [EUR]
Cuadrcóptero			
Bobina PLA 1Kg	20	1.5	30
Perfiles aluminio	2	1	2
Pack 4 Motores MT2204 II	25	1	25
ESC BlHeli 4 in 1	50	1	50
Baterías LiPo	25	2	50
PCB autopiloto	20	1	20
Componentes PCB	50	1	50
Hélices HQ5040	2.5	4	10
Total			**239**

En cuanto a la amortización del equipo, se han empleado 2 ordenadores para el desarrollo del software y para el entrenamiento de los algoritmos. Se ha considerado una amortización lineal del 10 % de la vida útil (10 años).

Equipo	Precio	Coste Amortización(10 %)
Pc sobremesa	1980	198
Pc portátil	1300	130
Total		**328**

Añadiendo un coste de encuadernado de la memoria de unos 30 euros el presupuesto total del proyecto ha sido

Concepto	Total [EUR]
Recursos humanos	5798.4
Material	239
Amortización del equipo	328
Encuadernación	40
Total	**6405,4**

A.2. Planificación

La realización de este trabajo ha empleado un ritmo continuo de horas de trabajo desde su comienzo, siendo un poco menor en épocas de exámenes y un poco mayor al comienzo de los cuatrimestres y julio. La dedicación media diaria del trabajo ha sido de unas 4 horas semanales, durante un periodo de unos 10 meses (descontando agosto y septiembre), lo que da un total de unas 800 horas. La inmensa mayoría de estas horas se han dedicado en el Centro de Automática y Robótica (CAR) de la Escuela Técnica Superior de Ingenieros Industriales (ETSII) de la Universidad Politécnica de Madrid (UPM), concretamente en el grupo de investigación de Visión por Computador y Robots Aéreos (CVAR).

En cuanto a la distribución del trabajo en este tiempo, el trabajo comenzó a realizarse en septiembre de 2018, durante los primeros meses se realizó el curso sobre redes neuronales y aprendizaje profundo, en la plataforma online Coursera. La duración del curso se extendió hasta finales de diciembre. Paralelamente, a partir de octubre se comenzó con el diseño de la aeronave, y en noviembre con el del autopiloto. A principios de febrero se finalizo con el diseño y construcción del cuadricóptero y con el diseño y montaje de la PCB del autopiloto. A partir de este punto, el resto del tiempo se ha dedicado al software, tanto el del autopiloto, como el de la estación de tierra , al diseño de los algoritmos de control y a la experimentación real. Se ha realizado un diagrama GANTT (**??**) en el que se ha detallado más en profundidad la distribución temporal de las tareas. Asimismo, se ha esquematizado la organización del proyecto en un diagrama EDP

Impacto social y medioambiental

El impacto social que tiene este trabajo se ve reflejado en su posible empleo en la educación y la investigación. Actualmente las metodologías docentes están tendiendo hacia el aprendizaje práctico, hacia aprender haciendo. Esta plataforma podría emplearse en centros docentes debido a su montaje mecánico hecho casi en su totalidad con impresión 3D.

Desde el punto de vista de la investigación, tener la posibilidad de desarrollar y probar nuevos algoritmos para el control de cuadricópteros puede mejorar la efectividad del uso de estas aeronaves en múltiples aplicaciones. Cuanto mejor sea el controlador, más fácil será utilizar estas aeronaves para tareas de inspección, seguridad y búsqueda y rescate, entre otras.

El impacto medioambiental de la plataforma es reducido, ya que, el PLA es un plástico biodegradable y las baterías de Litio, una vez descargadas, son sencillas de desechar. EL proceso de fabricación de los componentes requiere de recursos materiales y energéticos, cuyo proceso de obtención puede provenir de fuentes no renovables. Sin embargo, los beneficios sociales que se pueden extraer de los resultados del proyecto hacen asumible este impacto medioambiental.

Índice de figuras

Índice de tablas

Bibliografía

[1] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org