



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DISEÑO, CONFIGURACIÓN E
IMPLEMENTACIÓN DE UN DISPOSITIVO
BASADO EN EL SISTEMA DE
POSICIONAMIENTO OPTITRACK PARA EL
CONTROL INTEGRAL DE UN QUADROTOR**

AUTOR: CONRADO GARCÍA SALINAS

TUTOR: PEDRO JOSÉ GARCÍA GIL

Curso Académico: 2017-18



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño, configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño, configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

“Quiero aprovechar este espacio para dar las gracias por su inestimable apoyo y guía a lo largo de todo el proceso de creación de este TFG a Vicente Balaguer Garín, doctorando del Ai2, sin cuya participación este trabajo nunca hubiera visto la luz. Asimismo, considero justo el agradecimiento a Pedro García Gil, mi tutor, por sus consejos y su tutela y a mis compañeros de laboratorio. “



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño, configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



Este trabajo se ha orientado a la creación de un dispositivo que pueda controlar de forma integral un quadrotor en vuelo libre. Para ello se ha utilizado el sistema de posicionamiento Optitrack, se ha montado un quadrotor a partir de los elementos existentes en el laboratorio y el software desarrollado por el Ai2, se ha programado un servidor para comunicar el sistema Optitrack y el quadrotor y se ha diseñado físicamente el dispositivo, estableciéndose las experiencias necesarias para constatar el buen funcionamiento de todos los elementos. Los resultados de los experimentos confirmaron la bondad de los diseños. Se trata por tanto de un proyecto transversal que engloba diferentes áreas de conocimiento como pueden ser la informática, la automática, el diseño, la electrónica y las telecomunicaciones.

Palabras Clave: Dron, control, quadrotor, Optitrack, servidor.

This paper is dedicated to the creation of a control device, which is able to fully manage a flying quadrotor. In order to achieve that goal, Optitrack position system has been used. Also, a complete quadrotor was set and even a client-server program was written. Finally, a physical design of the control device was made and the necessary experiments were performed in order to check the right working of every element. The results of the experiments confirmed the rightness of the design. Therefore, in this paper we can find knowledge from a wide variety of subjects such as IT, automatics, design, electronics and even telecommunications.

Keywords: Drone, quadrotor, Optitrack, Server

Aquest treball s'ha orientat a la creació d'un dispositiu que (puga o pugui) controlar un "quadrotor" en vol lliure. Per això s'ha utilitzat el sistema Optitrack, s'ha (montat o muntat) un quadrotor a partir dels elements existents al laboratori i el "software desenvolupat per el Ai2, s'ha programat un servidor per comunicar el sistema Optitrack amb el quadrotor i s'ha dissenyat físicament el dispositiu, establint-se les experiències necessàries per constatar el bon funcionament de tots els elements. Els resultats dels experiments van confirmar la bondat dels dissenys. Es tracta per tant d'un projecte transversal que engloba diferents àrees de coneixement com poden ser la informàtica, l'automàtica, el disseny, l'electrònica i les telecomunicacions

Paraules Clau: Dron, control, quadrotor, Optitrack, servidor.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño, configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



DOCUMENTOS CONTENIDOS EN EL TFG

- Memoria
- Presupuesto
- Planos del dispositivo
- Anexos: códigos y planos

INDICE - MEMORIA

Capítulo 1 INTRODUCCIÓN TEÓRICA, OBJETIVOS Y ESTADO DEL ARTE	1
1.1 Motivación y objetivos	1
1.2 Estructura del documento.....	2
Capítulo 2 CINEMÁTICA Y DINÁMICA DE QUADROTORES	3
2.1 Cinemática.....	3
2.1.1 Sistemas de referencia y variables puestas en juego.....	3
2.1.2 Matrices de Rotación.....	4
2.2 Dinámica.....	6
2.2.1 Aproximación de Newton-Euler	6
2.2.2 Formulación de Lagrange-Euler.....	9
Capítulo 3 ELEMENTOS QUE COMPONEN EL SISTEMA DESARROLLADO	15
3.1 Hardware.....	15
3.1.1 Raspberry pi 2 model B	15
3.1.2 Puente de conexión Raspberry a Arduino (Cooking hacks)	15
3.1.3 Placa de circuitería con IMU de 10 DOF.....	16
3.1.4 Motores DC.....	16
3.1.5 Hélices Fibra de carbono	17
3.1.6 Marco DJI.....	17
3.1.7 Tornillería y cableado	17
3.1.8 Rotula	17
3.1.9 Sistema Optitrack	18
3.2 Software	19
3.2.1 Code Blocks	19
3.2.2 HMI.....	19



3.2.3 Putty	19
3.2.4 WinScp.....	19
Capítulo 4 TAREAS REALIZADAS	20
4.1 Implementación del quadrotor	22
4.1.1 Puesta a punto de la Raspberry.....	22
4.1.2 Modificación del Software de Ai2	23
4.1.3 Calibración de los motores y obtención de parámetros de los PID	24
4.1.4 Marcadores y Optitrack.....	25
4.1.5 Alimentación	27
4.1.6 Creación de un prototipo de dispositivo de control para hacer pruebas	27
4.2 Telecomunicaciones	28
4.2.1 Conexión Dispositivo-Optitrack.....	28
4.2.2 Conexión Opitrack-HMI-dron	28
4.3 Diseño físico del Dispositivo de Control	31
Capítulo 5 VALIDACIÓN EXPERIMENTAL	33
5.1 Vuelo en rótula	33
5.2 Vuelo libre sin rótula	35
5.4 Vuelo en rótula y control con dispositivo de control	37
Capítulo 6 CONCLUSIONES Y TRABAJOS FUTUROS	39
Capítulo 7 Bibliografía y Webgrafía.....	41



INDICE- FIGURAS MEMORIA

Fig. Cap 2-1 (Stojche Deskovski, 2016).....	3
Fig. Cap 2-2 (Raffo, 2007).....	14
Fig. Cap 3-1 Raspberry pi 2 model b (Amazon S.A, 2018)	15
Fig. Cap 3-2Raspberry to Arduino Shield (Arduino S.A, 2018)	15
Fig. Cap 3-3 IMU utilizada en el proyecto	16
Fig. Cap 3-4 Motor T-motor MT2216-12	16
Fig. Cap 3-5 Hélices de fibra de carbono.....	17
Fig. Cap 3-6 Marco DJI (DJI , 2018)	17
Fig. Cap 3-7 Rótula.....	17
Fig. Cap 3-8 Pantalla principal de Motive (Optitrack S.A, 2018)	18
Fig. Cap 4-1 Configuración genérica del archivo Interfaces de la Raspberry	23
Fig. Cap 4-2 Pestaña de Test del HMI.....	24
Fig. Cap 4-3 Pestaña Guided de HMI.....	25
Fig. Cap 4-4 Prototipo de dispositivo de control	27
Fig. Cap 4-5 Terminal con el servidor en ejecución.....	30
Fig. Cap 4-6 Impresión 3D del dispositivo de control.....	32
Fig. Cap 5-1 Pitch en rótula	33
Fig. Cap 5-2 Roll en rótula	34
Fig. Cap 5-3 Yaw en rótula.....	34
Fig. Cap 5-4 Roll en vuelo libre	35
Fig. Cap 5-5 Pitch en vuelo libre	35
Fig. Cap 5-6 Yaw en vuelo libre	36



INDICE-FIGURAS PRESUPUESTO

Fig. Cap 2-1 Cuadro de precios de materiales.....	1
Fig. Cap 2-2 Cuadro de precios de Mano de Obra	2
Fig. Cap 3-1 Cuadro de precios Unitarios	2
Fig. Cap 4-1 UO.1.....	3
Fig. Cap 4-2 UO.2.....	4
Fig. Cap 4-3 UO.3.....	5
Fig. Cap 4-4 UO.4.....	5
Fig. Cap 5-1 Presupuesto Parcial	6
Fig. Cap 6-1 Presupuesto Base de licitación.....	6



INDICE- ECUACIONES MEMORIA

EC (2-1)	4
EC (2-2)	4
EC (2-3)	5
EC (2-4)	5
EC (2-5)	6
EC (2-6)	6
EC (2-7)	7
EC (2-8)	7
(EC 2-9)	7
EC (2-10)	8
EC (2-11)	8
EC (2-12)	9
EC (2-13)	9
EC (2-14)	9
EC (2-15)	10
EC (2-16)	11
EC (2-17)	11
EC (2-18)	11
EC (2-19)	12
EC (2-20)	13
EC (2-21)	14



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

MEMORIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



Capítulo 1 INTRODUCCIÓN TEÓRICA, OBJETIVOS Y ESTADO DEL ARTE

A continuación, se pasa a hacer una breve introducción a los asuntos tratados en este TFG así como de la estructura de la presente memoria.

1.1 Motivación y objetivos

A lo largo de la última década hemos asistido a un desarrollo notable de los “drones”. Dentro de ellos, los impulsados por 4 hélices y conocidos como “quadrotors” son los más comunes tanto en usos recreativos como en industriales. Normalmente estos vehículos aéreos se controlan con la ayuda de un mando o “controladora” que se comunica con el dron mediante ondas de radio y es el elemento que lo provee de las referencias a seguir en cuanto a su orientación en el espacio. Al variar esta, varía también su posición y es entonces posible, para un ser humano con la suficiente pericia, controlar el quadrotor a su voluntad.

Sin embargo, esta controladora tiene como principales limitaciones el uso de las ondas de radio, las cuales, si hay varios drones volando en un espacio cerrado (Como en el caso de las cada vez más famosas carreras de drones) pueden solaparse y crear problemas. Además, resulta intuitivo su uso solamente hasta cierto punto ya que, con las nuevas tecnologías de realidad virtual es posible crear un dispositivo más parecido a un mando como los de los cazas o helicópteros, más interactivo y que nos permita una mejor experiencia de pilotaje.

Es por esto que el principal objetivo de este TFG es la creación de dicho dispositivo de tal manera que sea posible su implementación efectiva y que sienta las bases para posteriores trabajos que puedan mejorarlo. Así pues, las principales metas de este trabajo quedan resumidas en:

- Estudio de las características cinemáticas, dinámicas y relativas al control de un quadrotor como paso previo a la realización del dispositivo
- Creación de la infraestructura necesaria para que el dispositivo funcione como puedan ser la consecución de un prototipo de quadrotor capaz de volar establemente o las telecomunicaciones necesarias para la correcta transmisión de datos entre los elementos que intervienen en el control.
- Diseño del dispositivo y obtención de un prototipo capaz de controlar el dron.
- Realización de los ensayos necesarios para certificar el correcto funcionamiento de todos los diseños y el control efectivo del dron.

En definitiva, se ha enfocado el trabajo de una manera eminentemente práctica y no sólo como aplicación teórica, por lo que ha sido abundante el trabajo en laboratorio y los contratiempos derivados de la aplicación real de la teoría.



El trabajo se ha basado en la aplicación del sistema de posicionamiento Optitrack. Dicho sistema se compone de una serie de cámaras, las cuales, a partir de la radiación infrarroja emitida por ciertos cuerpos, llamados “marcadores” es capaz de construir una nube de puntos para describir la posición y orientación tanto de sólidos rígidos como de “esqueletos”. Este sistema es ampliamente utilizado en la industria del cine y de los videojuegos para captar movimientos naturales.

1.2 Estructura del documento

Esta memoria se estructura de la siguiente manera: tras abordar la introducción, se pasa a la descripción teórica de las particularidades cinemáticas y dinámicas del dron tipo quadrotor. Tras ello, una enumeración de los diferentes elementos usados en la realización del prototipo de quadrotor empleado en las experiencias. Posteriormente se detallan las tareas realizadas en cuanto a la comunicación entre el quadrotor y el Optitrack, así como otros detalles relativos a la comunicación entre elementos. Después se aborda el diseño del dispositivo adjuntándose los planos pertinentes y finalmente se tratan las experiencias realizadas y los resultados obtenidos.

Capítulo 2 CINEMÁTICA Y DINÁMICA DE QUADROTORES

En este capítulo se aborda la descripción de manera teórica del comportamiento cinemático y dinámico de los quadrotors. Hay que decir que existe abundante literatura al respecto y también numerosos recursos en la web. Han sido de particular ayuda los documentos facilitados por Pedro García Gil, tutor de este trabajo, así como el TFG “Desarrollo Integral de un Quadrotor: Diseño de un algoritmo de control para la posición X-Y basado en señales GPS” (Frasquet, 2014).

2.1 Cinemática

Primero se aborda la cinemática. Describiremos los principales sistemas de referencia, las variables involucradas y llegaremos a un modelo matemático que exprese la orientación en el espacio de dron, lo que se conoce como “actitud” o en inglés “attitude”.

2.1.1 Sistemas de referencia y variables puestas en juego

Primeramente, se definen los sistemas de referencia utilizados en el estudio. Todos ellos se sirven de los clásicos ejes cartesianos XYZ a partir de los cuales se definen los ángulos y variables necesarias :

El primero de ellos es el que tiene como origen el centro de masas del quadrotor y que es solidario a el, y matemáticamente es: $B = \{\vec{x}_L \vec{y}_L \vec{z}_L\}$. La siguiente figura ilustra, si bien con algunos cambios en la nomenclatura, lo expuesto anteriormente

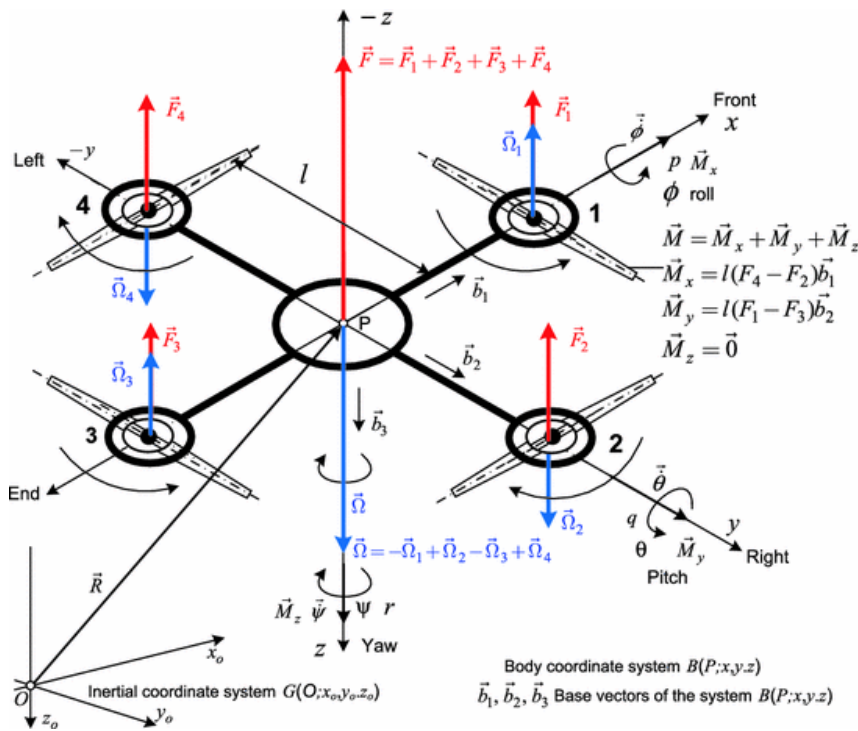


Fig. Cap 2-1 (Stojche Deskovski, 2016)



Nótese que en dicha figura también aparecen las fuerzas ejercidas por las hélices en cada rotor, que se pueden reducir a una sola fuerza situada en el c.d.m. También aparecen en esta figura tres ángulos; uno por cada eje y que describen la rotación del cuerpo rígido alrededor de dicho eje. Son los conocidos como “roll” (ϕ) “pitch” (θ) y “yaw” (ψ).

También definimos otro sistema de referencia esta vez fijo en el espacio y respecto al cual se expresará el movimiento de traslación del quadrotor $I = \{\vec{x} \vec{y} \vec{z}\}$.

2.1.2 Matrices de Rotación

Con estas variables ya podemos definir la matriz de rotación general, la cual se puede componer como el producto de las matrices de rotación en cada eje:

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\text{sen}\psi \\ 0 & \text{sen}\psi & \cos\psi \end{bmatrix}; R_\theta = \begin{bmatrix} \cos\theta & 0 & \text{sen}\theta \\ 0 & 1 & 0 \\ -\text{sen}\theta & 0 & \cos\theta \end{bmatrix}; R_\psi = \begin{bmatrix} \cos\psi & -\text{sen}\psi & 0 \\ \text{sen}\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$$R_i = R_\phi \cdot R_\theta \cdot R_\psi = \begin{bmatrix} \cos\psi \cdot \cos\theta & \cos\psi \cdot \text{sen}\theta \cdot \text{sen}\psi - \text{sen}\psi \cdot \cos\theta & \cos\psi \cdot \text{sen}\theta \cdot \cos\phi + \text{sen}\psi \cdot \text{sen}\phi \\ \text{sen}\psi \cdot \cos\theta & \text{sen}\psi \cdot \text{sen}\theta \cdot \text{sen}\phi + \cos\psi \cdot \cos\phi & \text{sen}\psi \cdot \text{sen}\theta \cdot \cos\phi - \cos\psi \cdot \text{sen}\phi \\ -\text{sen}\theta & \cos\theta \cdot \text{sen}\phi & \cos\theta \cdot \cos\phi \end{bmatrix} \quad (2-1)$$

Si definimos también el vector ω como el vector de las velocidades angulares en el sistema del quadrotor:

$$\omega = [p, q, r]$$

Y de ahí la matriz de velocidades (Olfati-Saber, 2001):

$$S(\omega) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ q & p & 0 \end{bmatrix}; \quad (2-2)$$



Podemos entonces expresar la variación de la orientación del aparato como:

$$\begin{aligned} \dot{R}_i &= R_i \cdot S(\omega); \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \cdot \sec\theta & \cos\phi \cdot \sec\theta \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{aligned} \quad (2-3)$$

Solo resta definir el movimiento traslacional del quadrotor, para ello combinamos las velocidades lineales de cada eje en un nuevo vector de velocidades V definido en el sistema del quadrotor y otro vector v , definido en el sistema fijo:

$$v = [v_0 \ u_0 \ w_0];$$

$$V = [v_l \ u_l \ w_l];$$

Siendo su relación

$$v = R_i \cdot V \quad (2-4)$$



2.2 Dinámica

Respecto de este punto, hay que decir que existen dos aproximaciones muy usadas en la literatura: la de Newton-Euler y la de Lagrange-Euler.

La primera de ellas parte de la conocida ley de la dinámica $F=m \cdot a$, para a partir de ahí extraer un modelo dinámico en 3 dimensiones. La segunda utiliza los conceptos de fuerza y coordenada generalizada para alcanzar un modelo compuesto por dos subsistemas, uno de rotación y otro de traslación.

2.2.1 Aproximación de Newton-Euler

La expresión $F=m \cdot a$, expandida a las tres dimensiones y teniendo en cuenta las aceleraciones angulares y la matriz de inercia (todo ello en el sistema de referencia del quadrotor) puede escribirse como:

$$\begin{bmatrix} m \cdot I_{3 \times 3} & 0 \\ 0 & J \end{bmatrix} \cdot \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mv \\ \omega \times J\omega \end{bmatrix} = \begin{bmatrix} F + Fd \\ \tau + \tau d \end{bmatrix}; \quad (2-5)$$

Donde se puede suponer diagonal la matriz de inercia (Raffo, 2007):

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2-6)$$

A continuación, consideraremos el vector de estado:

$$[\xi \ v \ \eta \ \omega]^T;$$

Donde $\xi \ v \in \mathfrak{R}^3$ representan la posición y velocidad lineal en el sistema fijo, y $\eta \ , \omega$ representan los ángulos de roll pitch, yaw y las velocidades angulares de cada eje expresado en el sistema del quadrotor.



Así las cosas, se pueden escribir las ecuaciones de movimiento del quadrotor como sigue:

$$\begin{aligned}\dot{\xi} &= v \\ m\dot{v} &= R_i \cdot F_b \\ \dot{R}_i &= R_i \cdot S(\omega) \\ J\dot{\omega} &= -\omega \times J\omega + \tau_b\end{aligned}\tag{2-7}$$

Donde F_b , τ_b son las fuerzas y pares externos aplicados al dron: su propio peso, las fuerzas aerodinámicas, las fuerzas de empuje y los pares desarrollados por los cuatro motores, pudiendo desarrollarse, además:

$$R_i \cdot F_b = -mg \cdot E_3 + R_{iE_3} \left(\sum_{i=1}^4 b \cdot \Omega_i^2 \right) + A_T;\tag{2-8}$$

$$\tau_b = - \sum_{i=1}^4 J_R (\omega \times E_3) \cdot \Omega_i + \tau_a + A_R;\tag{2-9}$$

Donde A_r y A_t son vectores que representan las fuerzas y pares aerodinámicos aplicados sobre el aparat y que se pueden modelar según $A_i = 0.5 \cdot \rho_{aire} C_i W^2$ donde ρ_{aire} es la densidad del aire, W es la velocidad del dron respecto del aire y C_i es el coeficiente de aerodinámico que depende de la geometría y la rugosidad del material sobre el que incide el aire.

Por otra parte, g es la constante gravitacional, J_R es el momento de inercia rotacional del rotor, b es el empuje de los motores y Ω_i es la velocidad angular de cada rotor.

En otro orden de cosas, el sumatorio de las fuerzas traslacionales está compuesto por el empuje de los rotores, la fuerza de la gravedad y la fuerza aerodinámica $U_1 = (\sum_{i=1}^4 b \cdot \Omega_i^2)$.

Según (Castillo, García, Lozano, & Albertos, 2007) el par de control aplicado a los tres ejes τ_a viene dado por:

$$\tau_a = \begin{bmatrix} (f_2 - f_4)l \\ (f_3 - f_1)l \\ \sum_{i=1}^4 \tau_{Mi} \end{bmatrix} = \begin{bmatrix} (\Omega_2^2 - \Omega_4^2)lb \\ (\Omega_3^2 - \Omega_1^2)lb \\ k_T(\Omega_3^2 + \Omega_1^2 - \Omega_2^2 - \Omega_4^2) \end{bmatrix} \cdot \begin{bmatrix} l \cdot U_2 \\ l \cdot U_3 \\ U_4 \end{bmatrix} \quad (2-10)$$

Y finalmente considerando como en (Mistler, Benallegue, & M'Sirdi, 2001) el vector de estados

$$\zeta = [x \ y \ z \ u_0 \ v_0 \ w_0 \ \phi \ \theta \ \psi \ p \ q \ r]^T$$

La ecuación del movimiento se puede escribir como:

$$\dot{\zeta} = \begin{cases} \dot{x} = u_0 \\ \dot{y} = v_0 \\ \dot{z} = w_0 \\ u_0 = \frac{1}{m} (\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) \cdot U_1 + \frac{A_x}{m} \\ v_0 = \frac{1}{m} (\sin\psi \sin\theta \cos\phi + \cos\psi \sin\phi) \cdot U_1 + \frac{A_y}{m} \\ w_0 = -g + \frac{1}{m} (\cos\theta \cos\phi) \cdot U_1 + \frac{A_z}{m} \\ \dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \\ \dot{\theta} = q \cos\phi - r \sin\phi \\ \dot{\psi} = q \sin\phi \sec\theta + r \cos\phi \sec\theta \\ \dot{p} = \frac{(I_{yy} - I_{zz})}{I_{xx}} qr - \frac{J_R \cdot \Omega}{I_{xx}} q + \frac{l}{I_{xx}} U_2 + \frac{A_p}{I_{xx}} \\ \dot{q} = \frac{(I_{zz} - I_{xx})}{I_{yy}} pr - \frac{J_R \cdot \Omega}{I_{yy}} q + \frac{l}{I_{yy}} U_3 + \frac{A_q}{I_{yy}} \\ \dot{r} = \frac{(I_{xx} - I_{yy})}{I_{zz}} qp - \frac{1}{I_{zz}} U_4 + \frac{A_r}{I_{zz}} \end{cases} \quad (2-11)$$

Según (Mistler, Benallegue, & M'Sirdi, 2001) este modelo puede asumirse suficientemente preciso para la representación de los movimientos del dron, sin embargo, según (Raffo, 2007) no es adecuado para el diseño del control, pues depende de fuerzas y momentos que son imprevisibles y por tanto, desconocidos. En consecuencia, lo que se hace es despreciar estos términos e introducirlos como perturbaciones externas



2.2.2 Formulación de Lagrange-Euler

La formulación de Lagrange-Euler está basada en el concepto de energía mecánica, la cual es suma de la energía asociada al movimiento o cinética E_c y de la posición o potencial E_p . Así, primeramente, se define la función lagrangiana.

$$L = E_c - E_p; \quad (2-12)$$

Y las fuerzas o pares generalizados dados por fuerzas o pares no conservativos se definen como:

$$\Gamma_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}; \quad (2-13)$$

Donde q_i representa a la coordenada generalizada y \dot{q}_i su primera derivada temporal

A continuación, consideramos las coordenadas generalizadas de un cuerpo rígido en el espacio tridimensional (Castillo, García, Lozano, & Albertos, 2007)

$$q = [x \ y \ z \ \phi \ \theta \ \psi] \in \mathfrak{R}^6$$

Que a su vez se puede separar en dos vectores pertenecientes a \mathfrak{R}^3

$$\xi = [x \ y \ z];$$

$$\eta = [\phi \ \theta \ \psi];$$

Que expresan la posición del centro de masas en el sistema de referencia fijo y los ángulos descritos en 2.2.1.

Entonces se desarrolla el término de Energía cinética de rotación:

$$E_{cRot} = \frac{1}{2} \int I v_B^2 dm \quad (2-14)$$

Donde $I v_B^2$ es la derivada de r_B^1 , el cual es un vector de posición que representa a un punto fijo del sistema de coordenadas B pero expresado en función del sistema de coordenadas I y que según (Fu, Gonzalez, & Lee, 1987) es de la forma:

$$r_{Bx}^1 = (\cos\psi\cos\theta)x_L + (\cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi)y_L + (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi)z_L$$

$$r_{By}^1 = (\sin\psi\cos\theta)x_L + (\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi)y_L + (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)z_L$$

$$r_{Bz}^1 = (-\sin\theta)x_L + (\cos\theta\sin\phi)y_L + (\cos\theta\cos\phi)z_L$$



Y siendo según (Raffo, 2007) $v_B^I = \dot{R}_1 \cdot r_B^1$ donde \dot{R}_1 es la matriz de rotación de B a I; queda la ecuación

$$\begin{aligned}
 E_{cRot} = & \frac{1}{2} \int (y_L^2 + z_L^2) dm (\dot{\psi}^2 \sin^2 \theta - 2\dot{\psi}\dot{\phi} \sin \theta + \dot{\phi}^2) \\
 & + \frac{1}{2} \int (x_L^2 + z_L^2) dm (\dot{\psi}^2 \sin^2 \phi \cos^2 \theta + 2\dot{\psi}\dot{\theta} \cos \phi \sin \phi \cos \theta + \dot{\theta}^2 \sin^2 \phi) + \\
 & \frac{1}{2} \int (x_L^2 + y_L^2) dm (\dot{\psi}^2 \cos^2 \phi \cos^2 \theta - 2\dot{\psi}\dot{\theta} \cos \phi \sin \phi \cos \theta + \dot{\theta}^2 \sin^2 \phi) + \\
 & \int (x_L y_L) dm (\dot{\psi}^2 \cos \theta \sin \theta \sin \phi + \dot{\psi}(\dot{\theta} \sin \theta \cos \phi - \dot{\phi}^2 \cos \theta \sin \phi) - \dot{\theta}\dot{\phi} \cos \phi) + \\
 & \int (x_L z_L) dm (\dot{\psi}^2 \cos \theta \sin \theta \cos \phi + \dot{\psi}(-\dot{\theta} \sin \theta \sin \phi - \dot{\phi} \cos \theta \cos \phi) + \dot{\theta}\dot{\phi} \sin \phi) + \\
 & \int (y_L z_L) dm (\dot{\psi}^2 \cos^2 \theta \sin \phi \cos \phi + \dot{\psi}(\dot{\theta} \cos \theta - 2\dot{\theta} \cos \theta \cos^2 \phi) + \dot{\theta}^2 \sin \phi \cos \phi)
 \end{aligned}$$

donde se ven los términos de la matriz de inercia

$$I_{xx} = \int (y_L^2 + z_L^2) dm$$

$$I_{yy} = \int (x_L^2 + z_L^2) dm$$

$$I_{zz} = \int (x_L^2 + y_L^2) dm$$

$$I_{xy} = \int (x_L y_L) dm = 0$$

$$I_{xz} = \int (x_L z_L) dm = 0$$

$$I_{yz} = \int (y_L z_L) dm = 0$$

La energía cinética puede ser reescrita pues, de la siguiente manera:

$$E_{cRot} = \frac{1}{2} I_{xx} p^2 + \frac{1}{2} I_{yy} q^2 + \frac{1}{2} I_{zz} r^2 = \frac{1}{2} \omega^T J \omega; \quad (2-15)$$

Se podría simplificar más aún esta expresión si se define W_η como el jacobiano que relaciona ω con $\dot{\eta}$ y se define además la siguiente matriz



$$J = J(\eta) = W_{\eta}^T J W_{\eta}; \quad (2-16)$$

Quedando finalmente la expresión

$$E_{cRot} = \frac{1}{2} \dot{\eta}^T J \dot{\eta}; \quad (17)$$

Finalmente se considera el término de la Energía potencial, la cual puede ser definida en términos de las coordenadas generalizadas como:

$$E_p = mgz; \quad (18)$$

Ahora, usando la función Lagrangiana previamente definida, se obtiene que las ecuaciones del movimiento completo son de la forma:

$$\begin{bmatrix} F_{\xi} \\ \tau_{\eta} \end{bmatrix} = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right); \quad (2-17)$$

Donde $\tau_{\eta} \in \mathfrak{R}^3$ representa los momentos de balanceo, cabeceo y guiñada y

$$F_{\xi} = R_1 \hat{F} = R_{1E_3} U_1 + A_T$$

Entonces se pueden obtener separadamente las ecuaciones de traslación y de rotación, puesto que el lagrangiano no contiene términos de la energía cinética que combinen $\dot{\xi}$ con $\dot{\eta}$.

Queda pues, para el movimiento de traslación

$$\begin{aligned} \frac{\partial L(\xi, \dot{\xi})}{\partial \xi} &= -mgE_3, \\ \frac{\partial L(\xi, \dot{\xi})}{\partial \dot{\xi}} &= -m\dot{\xi}, \end{aligned} \quad (2-18)$$



$$\frac{d}{dt} \left(\frac{\partial L(\xi, \dot{\xi})}{\partial \dot{\xi}} \right) = -m\ddot{\xi},$$

$$\frac{d}{dt} \left(\frac{\partial L(\xi, \dot{\xi})}{\partial \dot{\xi}} \right) - \frac{\partial L(\xi, \dot{\xi})}{\partial \xi} = F_{\xi}$$

$$m\ddot{\xi} + mgE_3 = F_{\xi}$$

De donde desarrollando los términos del vector de estados:

$$\begin{cases} \ddot{x} = \frac{1}{m} (\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) U_1 + \frac{A_x}{m} \\ \ddot{y} = \frac{1}{m} (\cos\psi \sin\theta \cos\phi - \cos\psi \sin\phi) U_1 + \frac{A_y}{m} \\ \ddot{z} = -g + \frac{1}{m} (\cos\theta \cos\phi) U_1 + \frac{A_z}{m} \end{cases}$$

Y las ecuaciones de rotación

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\eta}} \right) - \frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\eta}} &= \tau_{\eta} \\ \frac{d}{dt} \left(\frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\phi}} \right) - \frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\phi}} &= \tau_{\phi} \\ \frac{d}{dt} \left(\frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\theta}} \right) - \frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\theta}} &= \tau_{\theta} \\ \frac{d}{dt} \left(\frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\psi}} \right) - \frac{\partial L(\eta, \dot{\eta})}{\partial \dot{\psi}} &= \tau_{\psi} \end{aligned} \tag{2-19}$$

Donde se pueden resolver las ecuaciones y se llega según (Castillo, García, Lozano, & Albertos, 2007) a la expresión del modelo matemático:

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau_{\eta} \quad (2-20)$$

Donde $M(\eta)$ es $J(\eta)$, es decir

$$M(\eta) = \begin{bmatrix} I_{xx} & 0 & -I_{xx}\sin\theta \\ 0 & I_{yy}\cos^2\phi + I_{zz}\sin^2\phi & (I_{yy} - I_{zz})\cos\phi\sin\phi\cos\theta \\ -I_{xx}\sin\theta & (I_{yy} - I_{zz})\cos\phi\sin\phi\cos\theta & I_{xx}\sin^2\theta + I_{yy}\sin^2\phi\cos^2\theta + I_{zz}\cos^2\phi\cos^2\theta \end{bmatrix}$$

Y

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

Con

$$c_{11} = 0$$

$$c_{12} = (I_{xx} - I_{yy})(\dot{\theta}\cos\phi\sin\phi + \dot{\psi}\sin^2\phi\cos\theta) + (I_{zz} - I_{yy})\dot{\psi}\cos^2\phi\cos\theta - I_{xx}\dot{\psi}\cos\theta$$

$$c_{13} = (I_{zz} - I_{yy})\dot{\psi}\cos^2\theta\sin\phi\cos\phi$$

$$c_{21} = (I_{zz} - I_{yy})(\dot{\theta}\cos\phi\sin\phi + \dot{\psi}\sin^2\phi\cos\theta) + (I_{zz} - I_{yy})\dot{\psi}\cos^2\phi\cos\theta - I_{xx}\dot{\psi}\cos\theta$$

$$c_{22} = (I_{zz} - I_{yy})\dot{\phi}\sin\phi\cos\phi$$

$$c_{23} = -I_{xx}\dot{\psi}\sin\theta\cos\theta + I_{yy}\dot{\psi}\sin^2\phi\cos\theta\sin\phi + (I_{zz})\dot{\psi}\cos^2\phi\sin\theta\cos\theta$$

$$c_{31} = (I_{zz} - I_{yy})\dot{\psi}\cos^2\theta\cos\phi\sin\phi - I_{xx}\dot{\theta}\cos\theta$$

$$c_{32} = (I_{zz} - I_{yy})(\dot{\theta}\cos\phi\sin\phi\sin\theta + \dot{\phi}\sin^2\phi\cos\theta) + (I_{yy} - I_{zz})\dot{\theta}\cos^2\phi\cos\theta\sin\theta$$

$$c_{33} = (I_{yy} - I_{zz})\dot{\phi}\cos^2\theta\cos\phi\sin\phi - I_{yy}\dot{\theta}\sin^2\phi\cos\theta\sin\theta - I_{zz}\dot{\theta}\cos^2\phi\cos\theta\sin\theta + I_{xx}\dot{\theta}\sin\theta\cos\theta$$

Finalmente se obtiene la ecuación matricial

$$\ddot{\eta} = M(\eta)^{-1}(\tau_{\eta} - C(\eta, \dot{\eta})\dot{\eta}) \quad (2-21)$$

En la figura siguiente (fig2) se puede ver como se divide el sistema en dos subsistemas interconectados

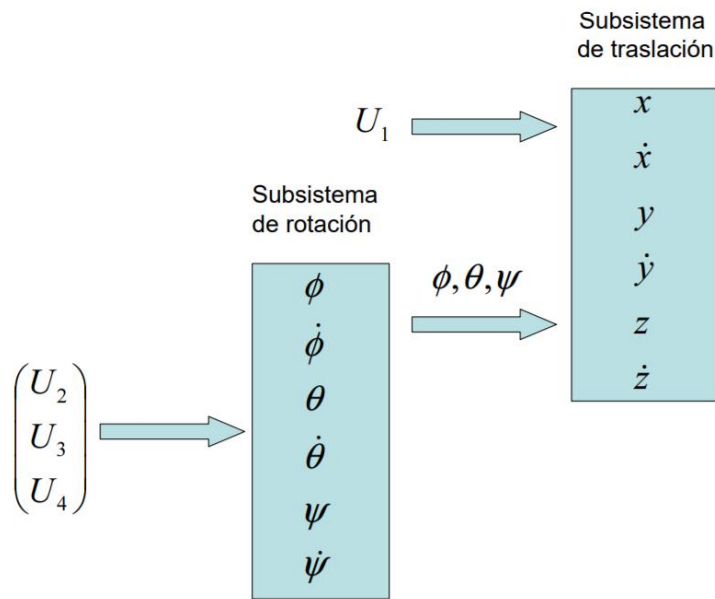


Fig. Cap 2-2 (Raffo, 2007)

Capítulo 3 ELEMENTOS QUE COMPONEN EL SISTEMA DESARROLLADO

Durante la realización del presente trabajo, se han utilizado una gran variedad de elementos los cuales se pasan a enumerar y describir a continuación:

Diferenciaremos entre los medios físicos o Hardware y el Software. En la parte de Hardware el elemento más relevante han sido la Raspberry pi, eje central del control del quadrotor. En cuanto al software, ha sido imprescindible el uso del autopiloto desarrollado por el equipo del laboratorio AI2 del departamento de automática.

3.1 Hardware

Los principales elementos Hardware utilizados han sido la Raspberry pi 2, el puente de conexión, la IMU, los motores, el marco y las hélices, a los que hay que asumir otros.

3.1.1 Raspberry pi 2 model B

Se trata de un miniordenador con un precio reducido pero unas características técnicas que lo hacen idóneo para desarrollar proyectos de robótica.

Cuenta con un procesador de cuatro núcleos a 900 MHz ARM y 1 GB de RAM.

Además, cuenta con los siguientes puertos y conexiones:

- 4 puertos USB
- 40 pines GPIO (General Purpose Input Output)
- 1 puerto Full HDMI
- 1 puerto Ethernet
- 1 Jack para audio de 3.5mm
- Ranura para tarjeta microSD
- Interfaz de cámara CSI
- Interfaz de pantalla DSI



Fig. Cap 3-1 Raspberry pi 2 model b (Amazon S.A, 2018)

Es importante comentar que el SO operativo utilizado en este trabajo ha sido un SOTR (Sistema Operativo en Tiempo Real)

3.1.2 Puente de conexión Raspberry a Arduino (Cooking hacks)

Se trata de una placa que permite usar diferentes periféricos de la misma forma que se haría con un Arduino pero aprovechando las capacidades de la Raspberry. Lo que hace es un “remapeo” de los pines de las raspberry para adecuarlos a la nomenclatura de Arduino.

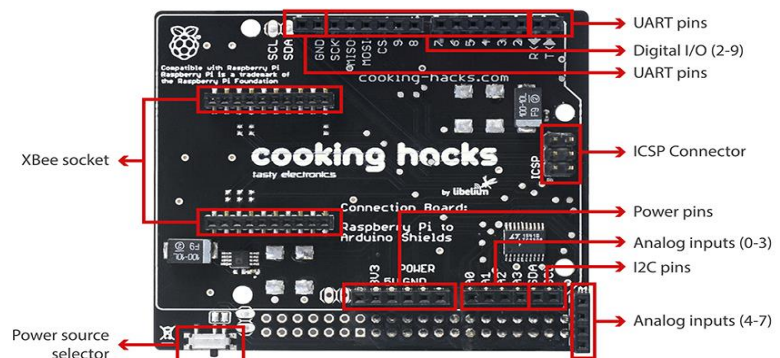


Fig. Cap 3-2Raspberry to Arduino Shield (Arduino S.A, 2018)

Es importante hacer notar que en este proyecto no se ha utilizado ningún Arduino, sin embargo, la inclusión de este componente se justifica mediante la necesidad de conectar los pines de los motores y la salida de la IMU de una manera cómoda.

3.1.3 Placa de circuitería con IMU de 10 DOF

Para que el dron sepa su propia orientación en el espacio necesita una IMU (Inertial Measurement Unit). Esta consta de 3 giróscopos situados uno en cada eje, de tal manera que se puede obtener la velocidad angular en cada dirección a partir de la medida de los giróscopos. La conexión al resto de elementos se realiza mediante unos terminales que están colocados en una placa de circuitería diseñada a tal efecto. Este diseño es anterior a este proyecto.

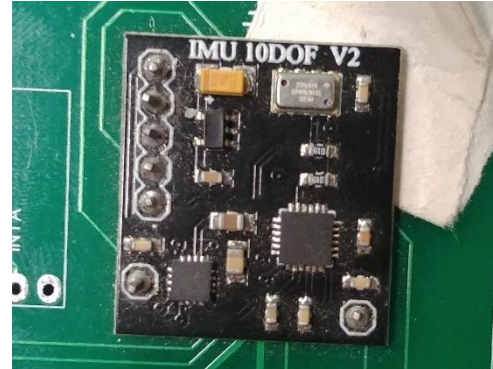


Fig. Cap 3-3 IMU utilizada en el proyecto

3.1.4 Motores DC

Se han utilizado 4 motores DC (Direct Current) marca T-motor, modelo MT2216-12.

Estos motores tienen las siguientes especificaciones técnicas extraídas de la página web oficial:

KV.....	800
Configuración.....	12N14P
Diámetro de Estator.....	22mm
Longitud Estator.....	16mm
Diámetro eje.....	4mm
Diámetro de motor (Dia.*Len).....	Φ27.5×34mm
Peso (g).....	75g
Corriente...(10)@10v(A).....	0.3A
No.de celdas(Lipo).....	3-4S
Max corriente continua(A)180S.....	18A
Max potencia (W)180S.....	260W
Max. Eficacia de corriente.....	(5-15A)>80%
Resistencia interna.....	175mΩ



Fig. Cap 3-4 Motor T-motor MT2216-12

3.1.5 Hélices Fibra de carbono

Se han incorporado al modelo 4 hélices de fibra de carbono de la marca T-motor y 22 Pulgadas de diámetro. Sin embargo, por motivos de seguridad, para la realización de los experimentos se implementaron hélices de 16 pulgadas y plástico.



3.1.6 Marco DJI

La estructura sobre la que se acomodan el resto de los elementos es un marco de dos colores de la firma DJI, concretamente el modelo F550. Sobre este marco se han atornillado en forma de torre el resto de los componentes.

Fig. Cap 3-5 Hélices de fibra de carbono

3.1.7 Tornillería y cableado

Para la realización de las conexiones pertinentes se han usado diversos cables macho-hembra de diferentes longitudes, así como varios tornillos de 2,5 MM y 3 MM, tuercas de la misma medida y topes.



Fig. Cap 3-6 Marco DJI (DJI , 2018)

3.1.8 Rotula

Para la calibración de los parámetros de control del dron y buscando siempre que no se desestabilizase, chocase y rompiese, se ha dispuesto de una rótula que permite el movimiento en orientación del quadrotor, pero no en traslación. Dicha rótula se acopla al marco del dron con una placa asegurada mediante tornillos.



Fig. Cap 3-7 Rótula

3.1.9 Sistema Optitrack

El sistema Optitrack consta de una serie de cámaras las cuales enfocan hacia un área determinada. Son capaces de detectar la presencia en tiempo real de ciertos marcadores en forma esférica, y trasladar esta información de un modo visual e intuitivo al usuario a través del software Motive. Dicho programa nos permite además definir tantos objetos rígidos como queramos, esqueletos y un sinfín de opciones. Además, posee salida multicast de los datos obtenidos, los cuales cuentan con una precisión del orden de milímetros.

En el caso concreto del laboratorio del Ai2 donde está instalado, se cuenta con 6 cámaras que delimitan un área rectangular. Además, se cuenta con una varilla especial de calibración y diversos marcadores, los cuales han formado parte del primer prototipo del dispositivo de control que constituye el objetivo principal de este TFG.

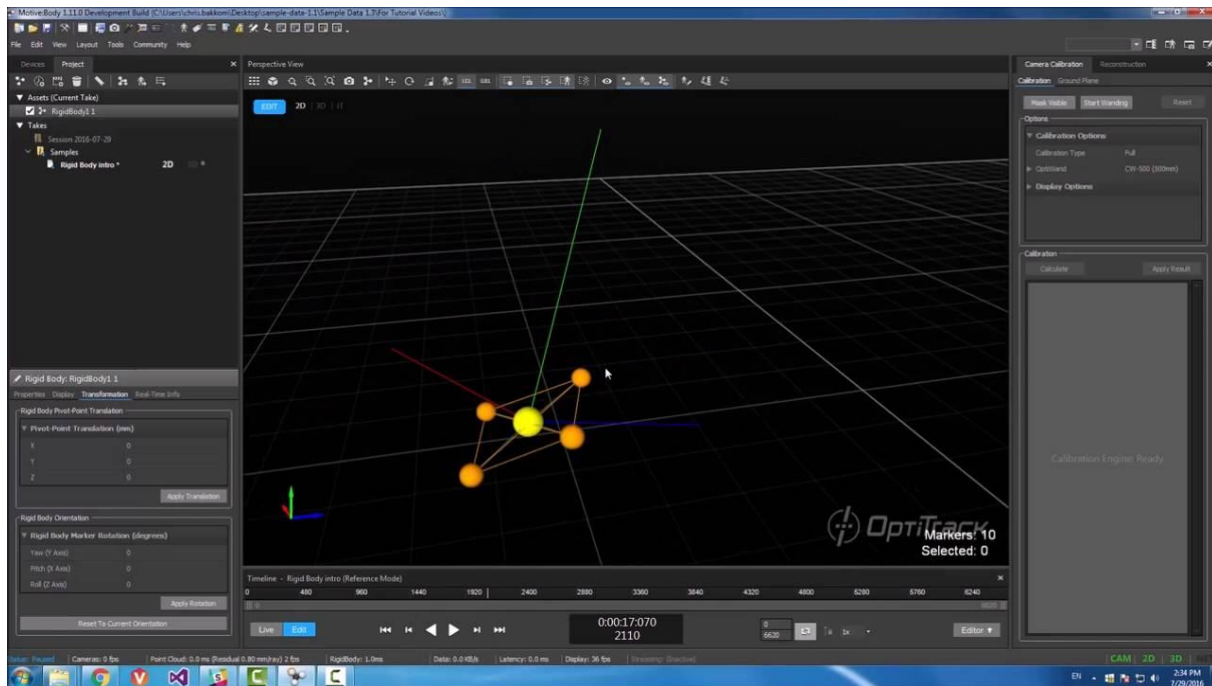


Fig. Cap 3-8 Pantalla principal de Motive (Optitrack S.A, 2018)



3.2 Software

En este apartado hablaremos de las diferentes herramientas informáticas usadas a lo largo de todo el presente trabajo. Ya se ha comentado la importancia del proyecto desarrollado en el laboratorio, sin embargo, también hay que mencionar otros elementos como puedan ser los siguientes:

3.2.1 Code Blocks

Code Blocks es un IDE de programación muy versátil y ha sido el único entorno de desarrollo usado en el proyecto.

3.2.2 HMI

Este programa, desarrollado en java por Vicente Balaguer Garín, es una interfaz humano-máquina (Human machine Interface) que permite acceder a los parámetros de los controladores PID implementados en el proyecto principal (los PID son sólo código) y modificarlos. Además permite la conexión con Optitrack, permitiendo una interacción entre este sistema y el quadrotor.

3.2.3 Putty

Putty es una popular herramienta informática que permite conectarse a la raspberry sin necesidad de acceder físicamente a ella, lo cual es de enorme utilidad a la hora de ejecutar los programas necesarios.

3.2.4 WinScp

Otra popular herramienta que en este caso proporciona la posibilidad de enviar y recibir archivos de la Raspberry en tiempo real. Utilizada para “subir” las diferentes actualizaciones del proyecto que hace funcionar al dron.



Capítulo 4 TAREAS REALIZADAS

En este capítulo se abordarán las principales acciones encaminadas a dar cumplimiento a los objetivos propuestos. Primero se profundizará en el montaje físico del quadrotor y su puesta a punto, posteriormente se describirán las soluciones aportadas al problema de telecomunicaciones entre el dispositivo de control, el sistema Optitrack y el dron y finalmente se mostrarán las etapas de diseño final de dicho dispositivo.

Es necesario dar visibilidad al hecho de que en un trabajo de índole práctica, aparecen con frecuencia problemas imprevistos que lastran la labor de laboratorio y retrasan y dificultan la consecución de objetivos, llegando incluso a alterarlos. Este trabajo no ha sido una excepción, y de todos estos problemas y contratiempos se da cuenta a lo largo de las siguientes páginas.

El trabajo en laboratorio se ha orientado en base a la consecución de objetivos parciales o hitos, algunos dependientes de tareas previas y otros totalmente independientes.

En la siguiente tabla se resumen estos objetivos organizados, acompañados de las experiencias de prueba que se propusieron para darlos por concluidos.

Tarea	Objetivo	Experimento	Hitos
Descargar versión debian para raspberry 3	Conseguir una raspberry funcional	Conexión con putty a la raspberry y ejecución de programas	Se decide cambiar a raspberry 2
Modificar archivo interfaces para lograr Ip estática			Conexión con putty e inicio de sesión
Conseguir una Raspberry 2 con adaptador Wi-fi			
Descargar versión debian para raspberry 2			
Montaje del quadrotor	Conseguir vuelo estable en rótula	Vuelo en rótula y resistencia ante perturbaciones	Los motores responden
Calibración de motores			HMI transmite información correctamente
Conexión con HMI			El quadrotor se estabiliza en la rótula
Ajuste parámetros PID			
Programación del programa cliente-servidor	Conseguir un programa que gestione el tráfico de datos entre Optitrack y dron	Ejecución del programa y análisis de datos obtenidos	Conexión correcta
Ajuste de sistema de referencia de Optitrack y Autopiloto	Conseguir un sistema de referencia unificado y vuelo libre sin rótula	Vuelo libre en rótula con cambios en la referencia de los ángulos roll pitch y yaw	Vuelo libre controlado desde HMI
Diseño de la función que gestiona el movimiento del Dispositivo de control	Controlar el dron a partir de los movimientos del Dispositivo	Vuelo en rótula y cambio en la referencia mediante el movimiento del dispositivo	Control posible
Mejorar el diseño del dispositivo	Obtención de un Dispositivo atractivo		Impresión 3d del diseño

Tabla 4-1 Cronología e hitos



4.1 Implementación del quadrotor

Tal como se ha dicho, se abordarán primero las actuaciones encaminadas a la obtención de un quadrotor capaz de volar

4.1.1 Puesta a punto de la Raspberry

La primera tarea a realizar, fue obtener una Raspberry capaz de operar con el Software disponible en el laboratorio. Primeramente, se valoró la opción de implementar una Raspberry Pi 3, esto se justifica por el hecho de que es una versión más reciente y por tanto lleva asociadas una serie de mejoras:

- Wifi integrado, por lo que se ahorra el uso de una antena wifi
- Mayor velocidad de procesamiento
- Mejor integración con el software más actual

Sin embargo, a la hora de la puesta en práctica, se encontraron diversos problemas que imposibilitaron esta elección, ya que tras cargar el Sistema Operativo anteriormente descrito en ella se constató que el wifi integrado no funcionaba correctamente y no tomaba la IP estática de manera correcta, perdiendo la conexión en algunos momentos e imposibilitando seguir con el trabajo. Además presentaba problemas al operar con el programa que gestiona el control del dron, llegándose a colgar en varias ocasiones.

Tras numerosos intentos y una larga búsqueda de soluciones, se concluyó que lo mejor era volver a la Raspberry pi 2 modelo b, la cual ya había sido probada en anteriores experimentos con resultados satisfactorios.

Se tomó pues, este modelo de Raspberry con el Sistema Operativo ya comentado y se accedió a ella a través de la herramienta Putty. Para esto fue necesario asignarle previamente una IP estática, lo cual se hizo modificando el archivo "interfaces":

```
GNU nano 2.2.6           Fichero: /etc/network/interfaces

auto lo

iface lo inet loopback
#iface eth0 inet dhcp
iface eth0 inet static

address 192.168.1.12
gateway 192.168.1.1
netmask 255.255.255.0

allow-hotplug wlan0
auto wlan0
iface wlan0 inet static
address 192.168.1.12
gateway 192.168.1.1
netmask 255.255.255.0
wpa-conf /etc/wpa.conf
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
#iface default inet dhcp
```

Fig. Cap 4-1 Configuración genérica del archivo Interfaces de la Raspberry

Para hacer efectiva la conexión se dispuso de una antena wifi en uno de los puertos USB, ya que el modelo 2 no lleva Wifi integrado.

Finalmente, la IP elegida para la Raspberry fue 192.168. 3.101. pudiendo conectarse exitosamente.

4.1.2 Modificación del Software de Ai2

Para el gobierno del dron se dispuso de un proyecto en Code Blocks desarrollado por el personal del laboratorio donde se realizó este TFG. Dicho proyecto, escrito en lenguaje C++, es realmente complejo y se necesitaron varios días para entender su funcionamiento y ser capaz de acometer las modificaciones oportunas. Dichas modificaciones fueron en algunos casos bastante simples, como es el caso de la actualización de ciertos parámetros como por ejemplo la IP.

Sin embargo, no siempre fue así pues también fue necesario adaptar una de las funciones preexistentes para gestionar dos cuerpos rígidos en lugar de uno solo, lo cual es algo más complejo. No obstante, la tarea se facilitó puesto que la función de la cual se partía fue diseñada con polimorfismo.

Esta función recibe los datos que se mandan desde el ordenador y los almacena. Posteriormente se desarrolló otra función que cogía estos datos almacenados y los convertía en la referencia a seguir por el dron. Dicha función la podemos ver en la siguiente figura

El control del dron se implementa situando la posición en cascada con la orientación y utilizando controladores PID, los cuales pueden modificarse para funcionar como PD o PID. Además, cuentan con filtro UDE. En este proyecto no se ha considerado oportuno modificar la estructura de control, simplemente se han efectuado actuaciones en los sensores de posición y el tratamiento de los datos por lo que sobre este apartado no se abundará en exceso.

4.1.3 Calibración de los motores y obtención de parámetros de los PID

Tras el conexionado de todos los elementos se procedió a probar el movimiento de los motores. Para ello se situó el quadrotor en la rótula, se alimentó mediante una fuente de tensión de 12 V y se estableció contacto con la Raspberry, ejecutándose el programa QuadRotor.exe, que es el nombre del software desarrollado por el Ai2. Paralelamente se ejecutó la interfaz hombre-máquina, (HMI) y se procedió a ir subiendo progresivamente la variable “thrust” para ver a partir de qué valor se comenzaba a mover el motor con fuerza suficiente para estabilizar el dron, siendo este 0.3. Una vez obtenido este valor, se reajustó dentro del programa QuadRotor la variable relativa a la PWM para que ese valor se definiera como el cero, es decir, el punto de partida.

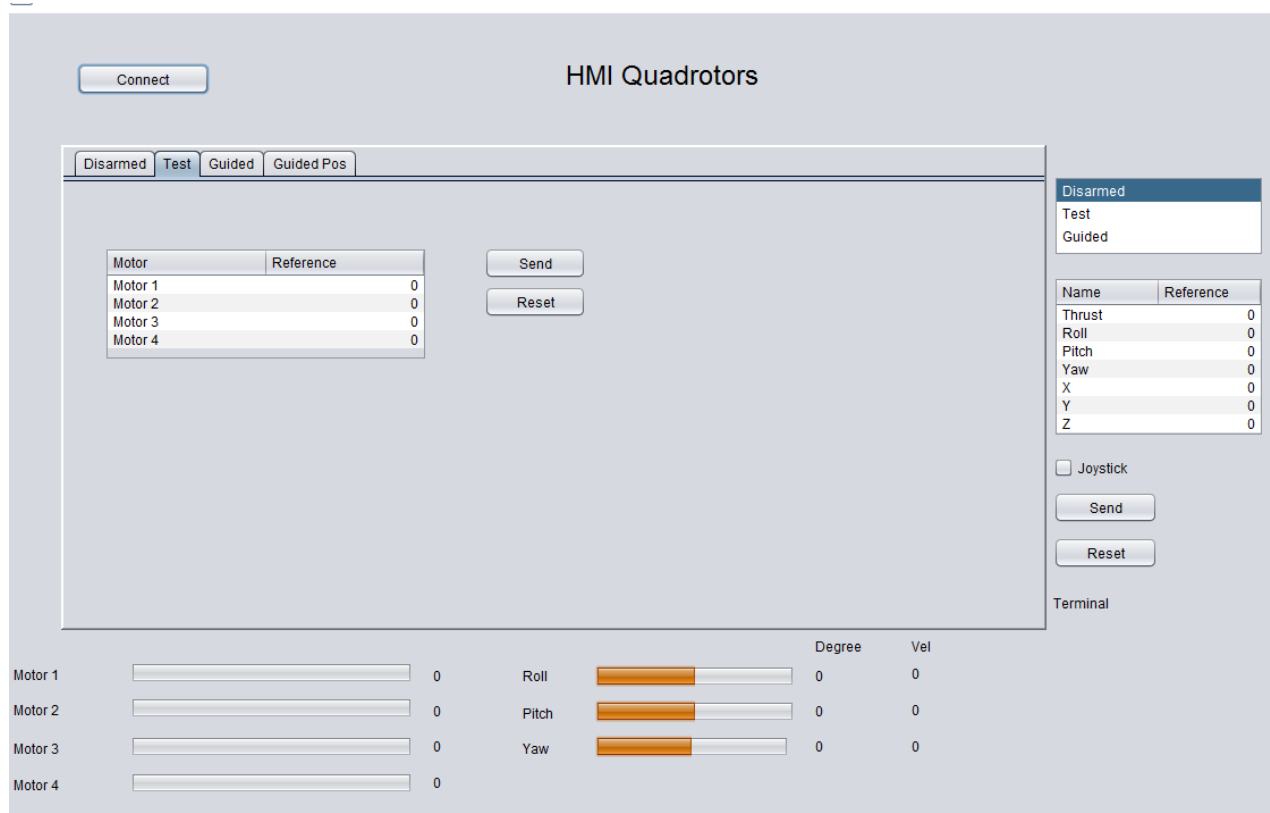


Fig. Cap 4-2 Pestaña de Test del HMI

Una vez conseguida la potencia necesaria, con un thrust de 0.3, se procedió a estimar mediante ensayo y error los valores de los parámetros de los PID Kp y Kd que efectúan el control del dron.

En la siguiente figura se refleja esto:

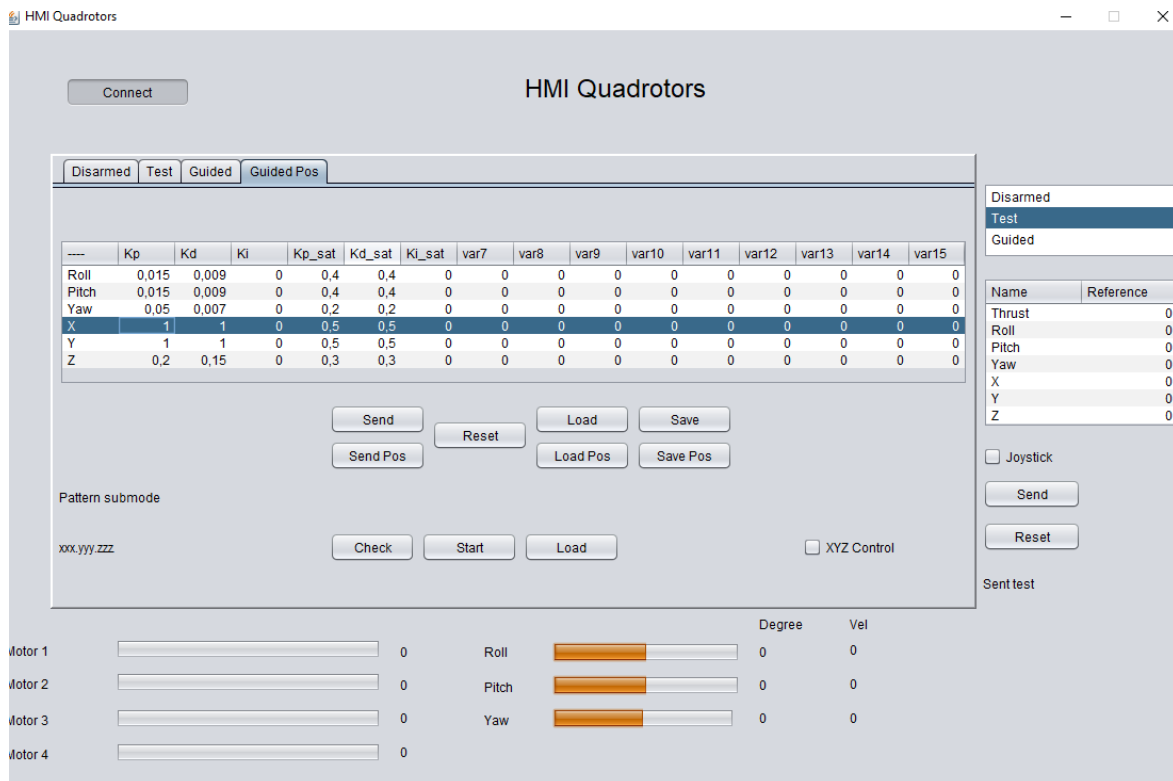


Fig. Cap 4-3 Pestaña Guided de HMI

Obteniéndose finalmente los valores reflejados en la figura anterior. Hay que hacer notar que se ha introducido una saturación de la acción de control, lo cual ayuda a que las reacciones del quadrotor no sean demasiado violentas y por tanto evita accidentes en el laboratorio.

4.1.4 Marcadores y Optitrack

Posteriormente se estimó necesario incluir en la estructura del dron algunos marcadores reconocibles por Optitrack. Con ellos, el Software Motive puede detectar el dron e interactuar con él, bien recibiendo medidas de su posición bien mandándole datos para que siga una determinada referencia.



Tras probar varias configuraciones, se optó por instalar sendas varillas de plástico en la línea de los brazos del quadrotor con un marcador al final de tres de ellas, además de algunos otros marcadores en puntos aleatorios de la estructura. Con esto se obtiene una nube de puntos asimétrica, lo cual es una condición necesaria para que Optitrack pueda asignar correctamente el sistema de coordenadas al cuerpo rígido que se construye asociando los distintos marcadores entre sí.

4.1.5 Alimentación

Inicialmente se planteó alimentar el quadrotor con una fuente de tensión existente en el laboratorio, la cual va unida por cable a los drivers de los motores y a un reductor de tensión, el cual hace que los 12 V de entrada pasen a los 5V necesarios para alimentar la Raspberry Pi.

Sin embargo, durante la realización de los experimentos, se constató que los motores demandaban demasiada energía, lo cual, si fijamos el voltaje en 12 V, producía un incremento de la Intensidad, llegando a los 19 A. En este punto, el convertidor se desconectaba y la Raspberry quedaba sin alimentación. La solución consistió en alimentar separadamente la Raspberry Pi con su cable "oficial".

De esta manera del quadrotor salen 2 cables, lo cual, aunque algo engorroso, permite realizar los ensayos de laboratorio sin tener que depender de una batería.

4.1.6 Creación de un prototipo de dispositivo de control para hacer pruebas

Con el objetivo de realizar pruebas en el laboratorio, se pegaron varios marcadores a un listón de madera, constituyéndose un rudimentario prototipo del dispositivo que constituye el fin de este trabajo. Se instalaron un total de 7 marcadores de manera asimétrica.

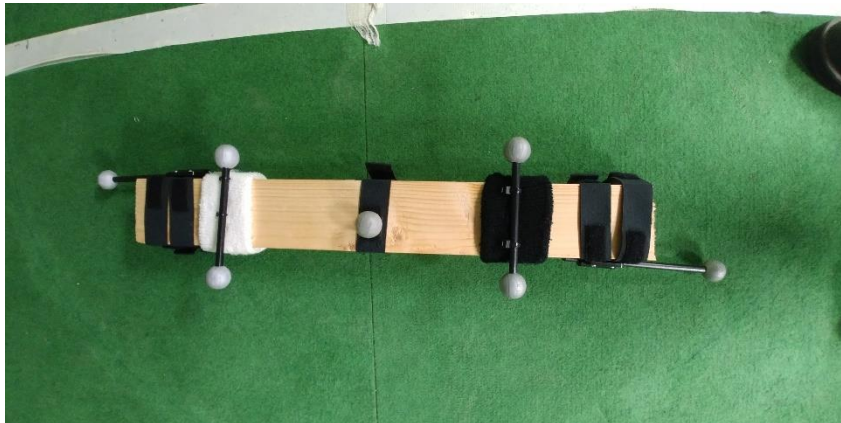


Fig. Cap 4-4 Prototipo de dispositivo de control



4.2 Telecomunicaciones

En este apartado figuran todas aquellas tareas encaminadas a lograr la correcta interacción entre dron, Optitrack y HMI.

Inicialmente se planteó la posibilidad de disponer de una red wifi, la cual fuera accesible por todos los elementos, y que permitiera en un futuro tener varios drones funcionando a la vez. Esto se pensó cuando se estaba probando la Raspberry 3. Se configuró un router wifi para tener una red con el nombre de “quadrotor”, sin embargo, con el cambio de Raspberry se consideró que la conexión directa entre dron y ordenador era más rápida y se dejó de lado esta opción. No obstante, ha quedado operativa en el caso de que se necesite en el futuro.

4.2.1 Conexión Dispositivo-Optitrack

Primeramente, se construyó un prototipo muy básico del dispositivo que constituye el principal objetivo de este TFG. Dicho prototipo constaba de una barra de madera rectangular a la cual se adosaron los diferentes marcadores. Una vez conseguido esto, en la interfaz de Motive, se configuraron los diferentes marcadores para que fueran reconocidos como un “sólido rígido”. De esta manera, Optitrack le asigna un centro geométrico a partir del cual define unos ejes que serán el sistema de referencia en función del cual vendrán dados los parámetros de posición y orientación.

Tenemos pues, las coordenadas XYZ, de las cuales el origen se corresponde con el centro del círculo del campo de fútbol para robots que hay en laboratorio. Situado el prototipo en el origen y posición de reposo horizontal, se constató una inclinación de cero grados en cualquier dirección.

4.2.2 Conexión Optitrack-HMI-dron

Optitrack presenta la opción de enviar los datos que recibe en tiempo real a través de una conexión multicast. Dicha opción permite enviar los datos desde Optitrack a cualesquiera dispositivos estén dados de alta en el “grupo multicast”. Este grupo presenta una IP que en nuestro caso es: 205.49.55.107. Además, la conexión se realiza a través del puerto 1051.

Así las cosas, se necesita que esta información sea recibida por el HMI. Este programa cuenta con una clase denominada NatNet. La cual puede ser modificada para gestionar dos sólidos rígidos. Esta es la opción que se valora en un primer momento, teniendo éxito en su consecución. Sin embargo, se plantea también la posibilidad de crear un programa cliente-servidor, el cual gestione el tráfico de datos independientemente del HMI. Este programa además debe traducir lo que manda el Optitrack a un formato “entendible por el dron”. A continuación, se pasa a explicar esto con más detalle.

El caudal de datos que aporta optitrack tiene la forma de un vector tipo “char” de 512 bits donde no todos los datos son útiles. Optitrack lo que hace es “congelar” la imagen que detectan las cámaras cada pocos milisegundos. Esta “fotografía” se denomina “frame”. En esta imagen puede haber uno o varios sólidos rígidos, a los cuales se les asigna un número de identificación. Posteriormente se



adjuntan los datos de las coordenadas XYZ y los cuaterniones de cada sólido rígido. La estructura es similar a lo siguiente.

INFO	Md	Longitud	Frame	Data sets	Model name	Marker data	Num RB	ID1	X	Y	Z	Q1
POSICION (BITS)	0	2	6	10	267	271	275	279	283	287	291	295
INFO	Q2	Q3	Q4	ID2	X	Y	Z	Q1	Q2	Q3	Q4	
POSICION (BITS)	299	303	307	411	415	419	423	427	431	435	439	

Tabla 4-2 Buffer de entrada al servidor

Y el dron lee un mensaje con el siguiente formato

INFO	ID 1	X	Y	Z	roll	pitch	yaw	ID 2	X	Y	Z	roll	pitch	yaw
POSICION (BITS)	0	4	8	12	16	20	24	28	30	34	38	42	46	50

Tabla 4-3 Buffer de entrada al dron

Hay que mencionar, que previamente a este TFG, no se había trabajado nunca con programas cliente-servidor. Por lo tanto, había un desconocimiento general del funcionamiento y las particularidades de esta rama de la informática

Se realizó una investigación al respecto donde fue necesario aprender la estructura de este tipo de programas, así como diversos puntos relacionados con la programación.

Se diseñó pues un programa en lenguaje C que hiciera posibles dos procesos de comunicación:

- Por un lado, funcionando como servidor, recibiendo datos de Optitrack, que sería el cliente
- Por otro, funcionando como cliente, enviando datos al dron, que sería en este caso el servidor

Finalmente se consiguió el código que figura en los Anexos, el cual una vez ejecutado arroja:

```
Seleccionar C:\Users\Optitrack\Desktop\ultimaesperanza1.exe
ID : 0
pos: [0.02,-0.01,0.06]
ori: [-0.00,-0.02,-0.04,1.00]
End Packet
-----
sending to drone with direction 192.168.3.100
mensaje enviado:
waiting for data
Byte count : 220
Frame # : 883078
Marker Set Count : 0
Rigid Body Count : 1
ID : 0
pos: [0.02,-0.01,0.06]
ori: [-0.00,-0.02,-0.04,1.00]
End Packet
-----
sending to drone with direction 192.168.3.100
mensaje enviado:
waiting for data
Byte count : 220
Frame # : 883079
Marker Set Count : 0
Rigid Body Count : 1
ID : 0
pos: [0.02,-0.01,0.06]
ori: [-0.00,-0.02,-0.04,1.00]
End Packet
-----
```

Fig. Cap 4-5 Terminal con el servidor en ejecución

Es necesario comentar que se ha tomado un ejemplo ya escrito por Genesis con fecha de 07/11/02 y se ha modificado convenientemente usando algunos ejemplos que provee la casa Optitrack y conocimientos propios. Esto queda patente en la cabecera del programa, donde aparece toda la información del código original. Posteriormente se incluyen las librerías necesarias. Hay que destacar “Winsock.h” que es la que nos permite trabajar con Windows de una forma similar a como se hace en sistemas UNIX (Linux).

Una versión completa de este código puede encontrarse en los anexos. Cabe destacar que esta ha sido con diferencia la parte más complicada de todo el trabajo realizado debido por una parte al desconocimiento de los programas cliente-servidor que ha sido fuente de errores y por otra, a la propia naturaleza de la programación y el “debugging” en sí.

Como principales contratiempos una vez compilado y probado el programa hay que mencionar el hecho de que, debido a la poca capacidad de la conexión Wifi entre dron y ordenador, se perdía demasiada información en el proceso de comunicación, incumpléndose el tiempo real y ocasionando la saturación de la Raspberry Pi. Para solucionarlo, se instalaron varias antenas en la torre del ordenador las cuales mejoraron la intensidad de la señal.

El otro gran inconveniente vino ocasionado por la frecuencia de muestreo de datos. Optitrack envía datos con un “caudal” de 1 MB/s, a 100 frames por segundo. En aras de una mayor fidelidad, el cliente-servidor en un primer momento enviaba un caudal igual al dron, lo cual resultó ser demasiado y ocasionaba la saturación del programa Quadrotor.exe. La solución consistió en introducir todo el proceso de lectura, transformación y envío de referencias dentro de un bucle “if”, el cual solo se



ejecuta cuando el contador que tiene asignado llega a un valor de 20. De esta manera se reduce la frecuencia con la que se envían datos y la Raspberry puede gestionarlo.

Como consecuencia de esta medida, hubo que introducir un contador similar dentro del programa QuadControl.exe. Concretamente en la parte de control, pues al estar este fragmento de código ejecutándose con mayor frecuencia, se realizaba control varias veces sobre los mismos datos, y puesto que la derivada de varios datos consecutivos e iguales es cero (no hay incremento ni decremento) esto afectaba al control desestabilizando el dron.

Ahora, dentro del dron, se necesita un código que, tras recibir el mensaje, “entienda” que estas son las nuevas referencias y las pase como tal a la parte del programa que realiza el control.

Esto se ha conseguido mediante la definición de dos nuevos objetos de tipo “Rigid Object” y “Rigid Object Euler” estas clases ya existían en el proyecto del Ai2.

4.3 Diseño físico del Dispositivo de Control

Una vez que ya se han puesto a prueba las demás partes y viendo que el quad cumple con su función, se estima oportuno el diseño de una versión mejor del prototipo. Principalmente se busca una mayor integración en el cuerpo del piloto, un diseño que economice la cantidad de material y un bajo coste. Se decide, dado el uso que se le ha dado durante la carrera, la utilización del software Inventor, de la casa Autodesk.

El diseño debe cumplir algunas especificaciones técnicas:

- Debe contar con un número adecuado de marcadores reconocibles por Optitrack
- Estos marcadores deben de estar en una posición y tener una forma tal que posibiliten correctamente su detección por las cámaras, siendo la asimetría un factor clave.
- Debe ser un sólido rígido, con los marcadores estáticos.
- Debe permitir una gama de movimientos tal que permita el control efectivo del dron en todas las direcciones
- Debe tener un peso reducido
- Debe ser barato de fabricar

En un primer momento se plantea la opción de hacer un “guante” con marcadores en los dedos, sin embargo, para que se diera la condición de sólido rígido, se deberían tener los dedos estirados en todo momento, lo cual no resulta cómodo. Además, no hay mucha distancia entre marcadores, lo cual hace que las medidas de Optitrack no sean fiables.

Posteriormente se decide crear una “muñequera” modificada para que cubra también el antebrazo. Sobre esta estructura, aligerada con huecos, se disponen los marcadores. Los planos pueden encontrarse en los anexos

Se ha optado por fabricar una copia a escala 1:3 en resina blanca mediante el proceso de fabricación aditiva o Impresión 3D para lo cual se ha contado con la colaboración de la empresa “Printable 3D” la cual, sirviéndose de un modelado 3D en Inventor ha realizado la impresión. Dicha empresa ha proporcionado también información relativa al costo de este proceso, la cual puede encontrarse en el Presupuesto.



Fig. Cap 4-6 Impresión 3D del dispositivo de control

Capítulo 5 VALIDACIÓN EXPERIMENTAL

Para dar validez a los diseños realizados se han desarrollado ciertos experimentos:

El primero de ellos se hizo al conseguir que el dron estuviera estable en la rótula. El segundo, una vez conseguido el vuelo libre controlado por HMI y finalmente se planteó un tercero al desarrollar el código que gestiona el Dispositivo de control.

5.1 Vuelo en rótula

En este experimento se sitúa el dron atornillado a la rótula, la cual, aunque introduce algo de resistencia, permite la estabilidad del quadrotor. En este experimento no se usa el sistema Optitrack, obteniendo los datos de la IMU de 10 DOF con la que cuenta el dron.

Como se puede apreciar en principio existe un error de posición, debido a la ausencia de acción integral

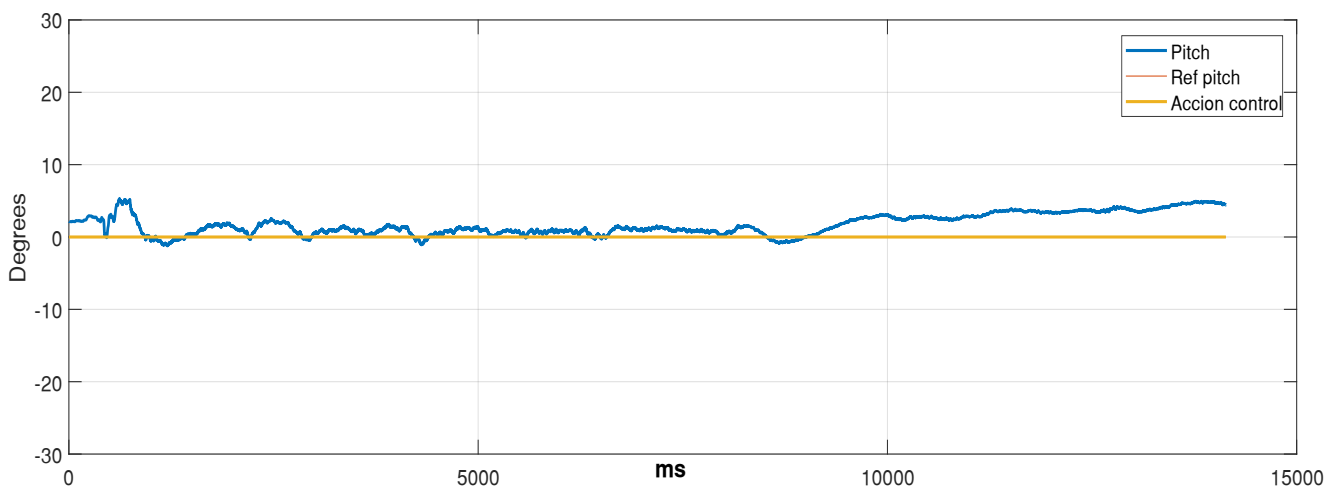


Fig. Cap 5-1 Pitch en rótula

y una importante sobreoscilación. No obstante, el dron se sostiene en la rotula y aún resiste perturbaciones no demasiado fuertes. En el tramo final del ensayo vemos una mayor deriva del pitch, no obstante, esto puede ser debido a la acción del cable de alimentación, que al enroscarse en la rótula puede haber ejercido fuerza hacia abajo.

En el caso del Roll, se ve que la sobreoscilación es menor y que también hay un error de posición y una

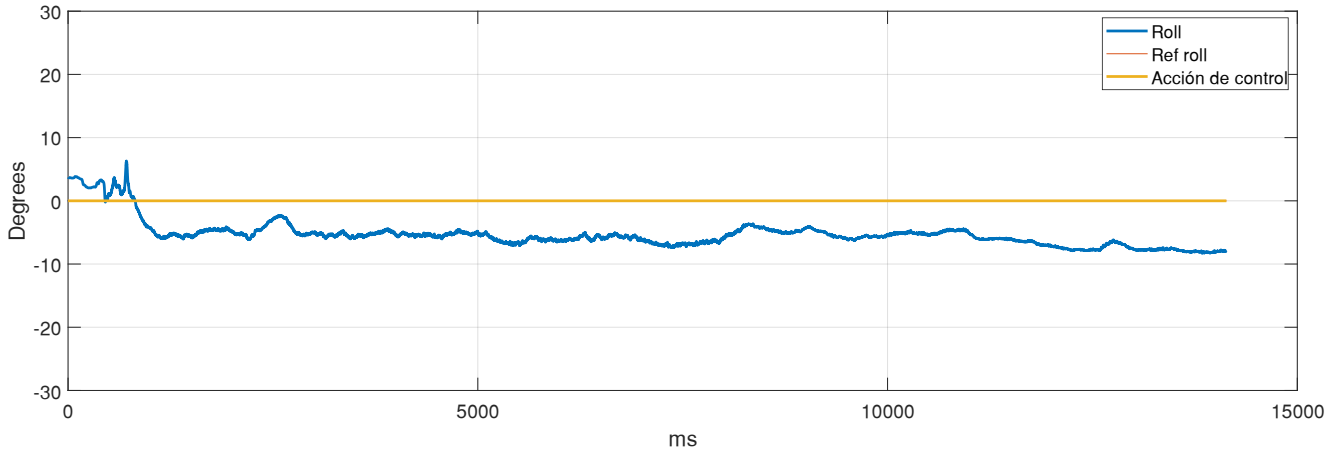


Fig. Cap 5-2 Roll en rótula

ligera tendencia negativa, probablemente causada, al igual que en el caso del Pitch, por la acción del cable.

En el caso del Yaw, se parte de yaw negativo, y se ve como a partir de cierto punto (cuando se modifica la posición del cable de alimentación) el dron rápidamente busca la posición de yaw cero, siguiendo la referencia. Se ha escogido esta grafica porque ilustra bien la capacidad del dron para estabilizarse.

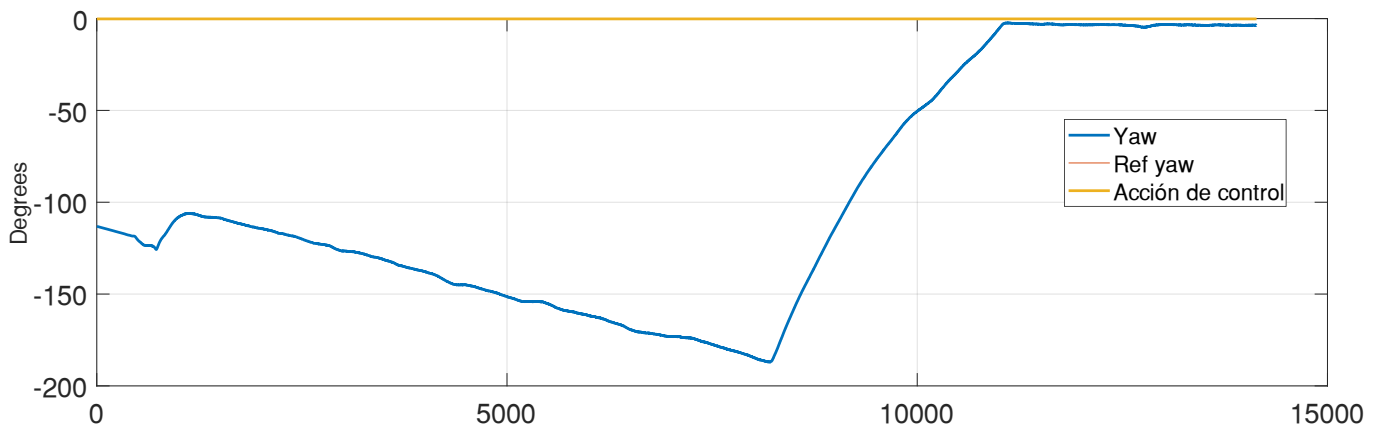


Fig. Cap 5-3 Yaw en rótula

5.2 Vuelo libre sin rótula

Este experimento consiste en el vuelo completamente libre, o con la mínima intervención, del dron.

Partiendo de una posición de reposo se fija un Thrust de 0.4 y una referencia de altura de 1 metro. A partir de aquí, se deja el dron volar para ver si mantiene su posición en el espacio obteniendo las medidas de posición del sistema Optitrack.

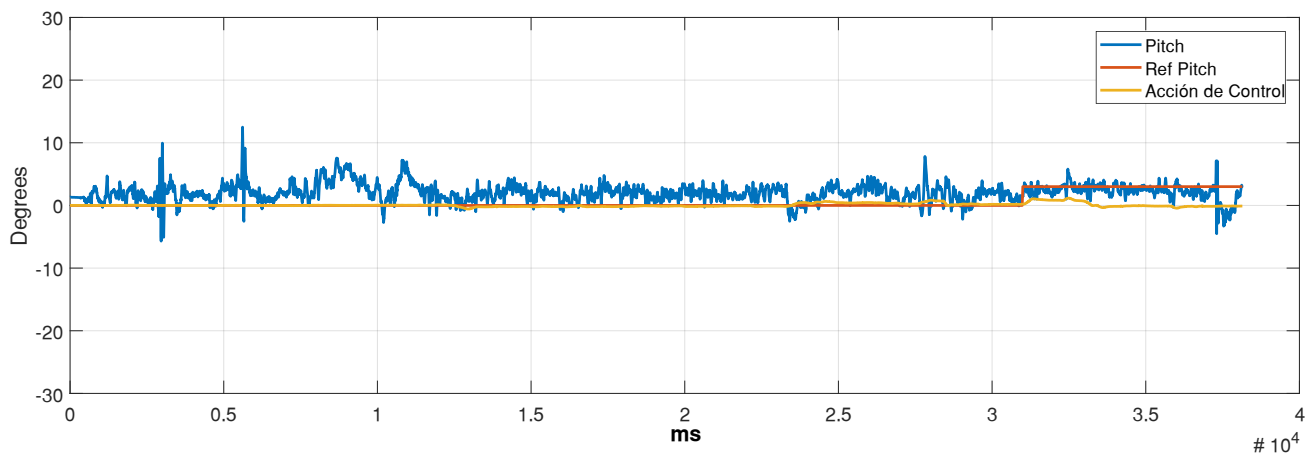


Fig. Cap 5-5 Pitch en vuelo libre

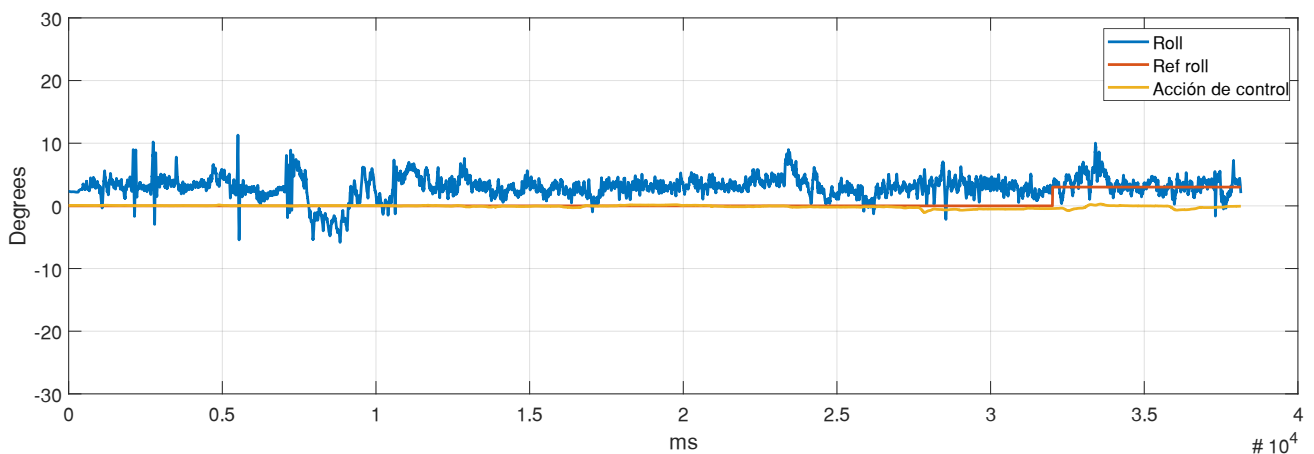


Fig. Cap 5-4 Roll en vuelo libre

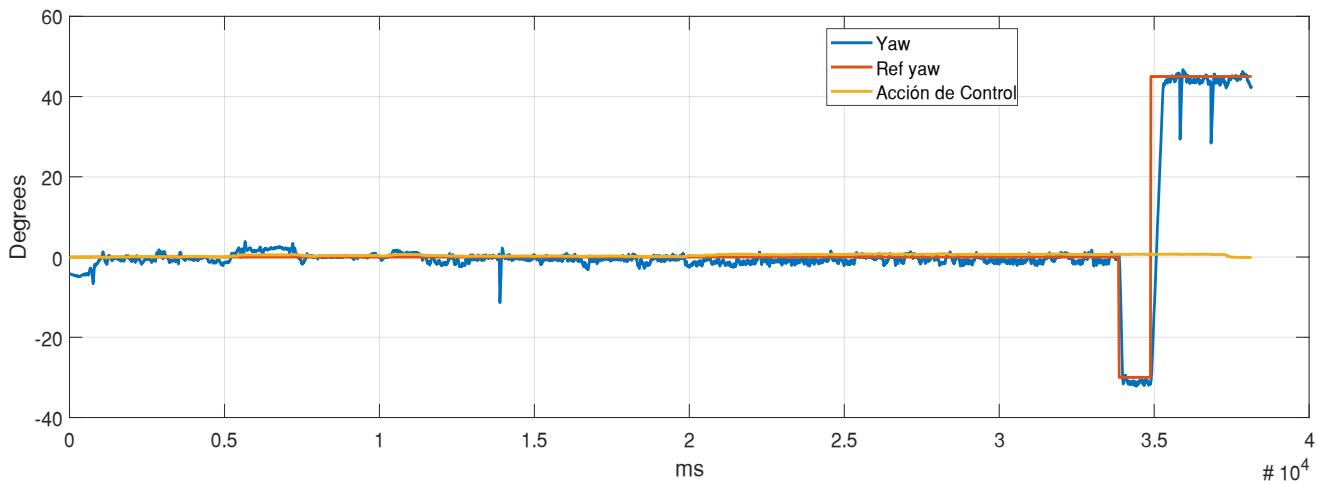


Fig. Cap 5-6 Yaw en vuelo libre

Como se desprende de la observación de las gráficas, hay presente una sobreoscilación relativamente importante que sin embargo no limita el control, pues el hecho es que el dron vuela establemente tal

Hay que mencionar también un error de posición debido a la ausencia de acción integral en los PID utilizados. Durante este experimento se saturó la salida de la acción de control. Durante el experimento se introdujeron perturbaciones tanto en roll como en pitch, con el objetivo de comprobar la estabilidad del dron, no observándose que el sistema se hiciera inestable.

5.4 Vuelo en rótula y control con dispositivo de control

En este experimento se comprobó que el prototipo diseñado no servía, pues Optitrack no reconocía sus marcadores. Como solución se utilizó otro quadrotor de otro proyecto el cual también cuenta con marcadores de Optitrack, y se utilizó como Dispositivo de Control.

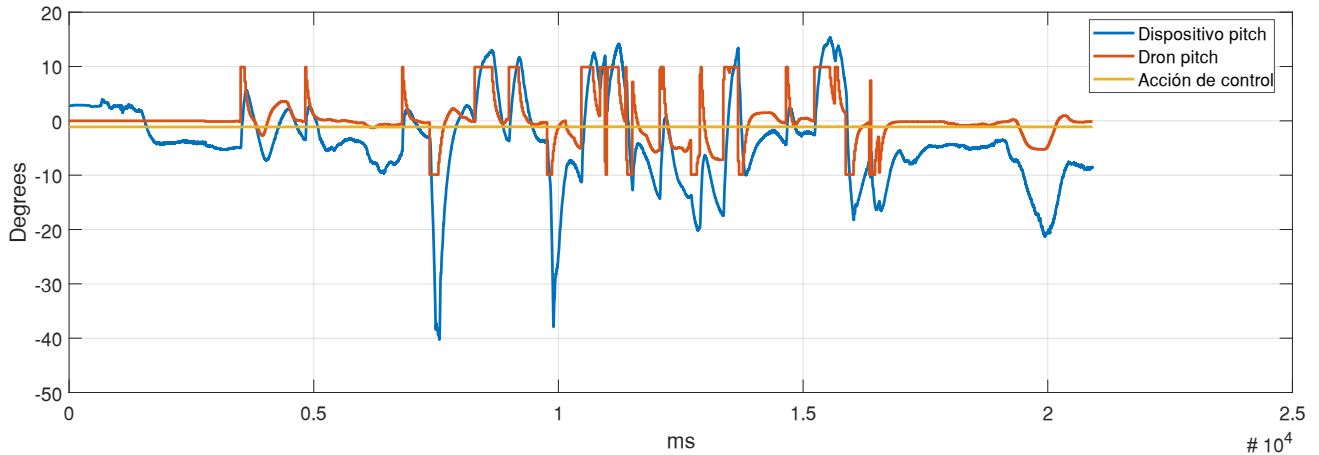


Fig. Cap 5-6 Vuelo en rótula con Dispositivo de control-pitch

Como se puede desprender de la observación de la gráfica, vemos que efectivamente hay un seguimiento de la referencia, si bien con cierto retraso o “lag” debido sin duda al tiempo que tardan los datos de posición de las cámaras hasta el servidor y de ahí al dron de nuevo para la realización del control.

Hay que destacar que no se consigue error de posición nulo porque los controladores no tienen acción integral, solo proporcional y derivativa.

Además, durante el ensayo se introdujo un factor corrector sobre la posición del Dispositivo de control, para tener una menor “sensibilidad” y poder maniobrar más claramente.

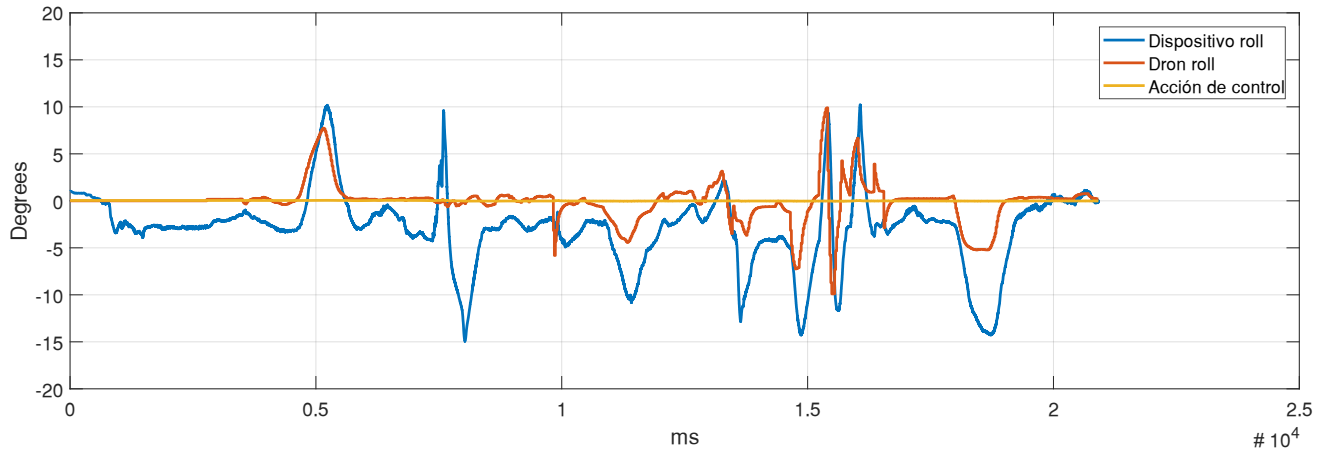


Fig. Cap 5-7 Vuelo en rôtula con Dispositivo de control-roll

El control de roll es en principio algo peor, se vuelve a observar el “lag” que también aparece en el pitch.

Lo más relevante de esta gráfica es que el dron se inclinaba mucho en roll negativo, esto es posiblemente debido a que se estaban probando también el pitch y el yaw al mismo tiempo, lo cual puede haber afectado al control.

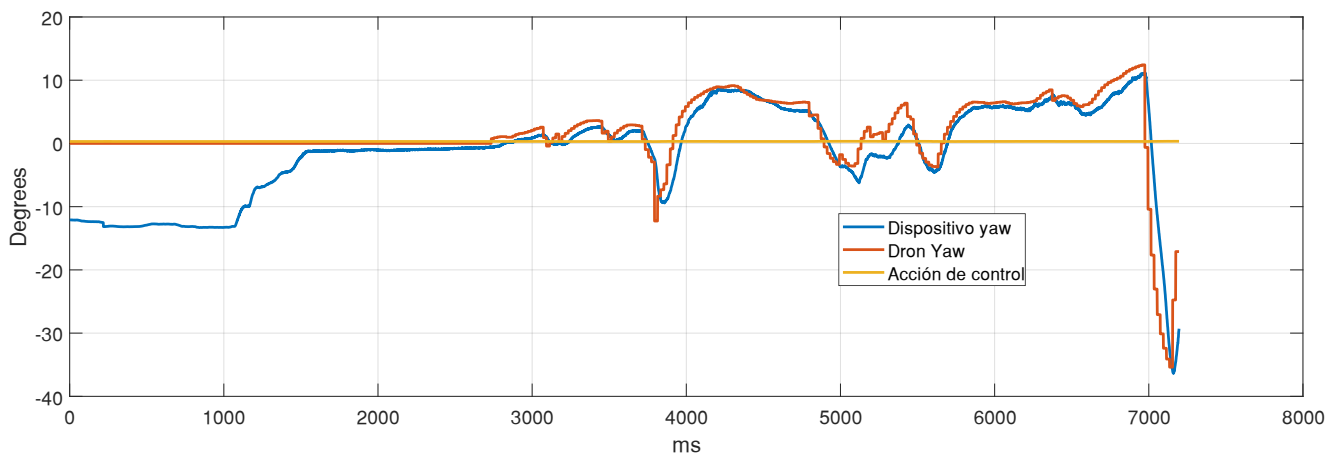


Fig. Cap 5-8 Vuelo en rôtula con Dispositivo de control-yaw

El control de yaw por el contrario se muestra mucho más preciso, con un nivel de “lag” menor. En esta dimensión se prescindió del factor de corrección.



Capítulo 6 CONCLUSIONES Y TRABAJOS FUTUROS

A la vista de los resultados obtenidos se pasa a discutir la consecución de objetivos

Respecto al primer objetivo “Estudio de las características cinemáticas, dinámicas y relativas al control de un quadrotor como paso previo a la realización del dispositivo” Hay que decir que se ha realizado un estudio en profundidad de lo arriba citado, obteniéndose una comprensión suficiente para realizar las posteriores tareas, por lo tanto, se da por conseguido.

Sobre el segundo objetivo, “Creación de la infraestructura necesaria para que el dispositivo funcione como puedan ser la consecución de un prototipo de quadrotor capaz de volar establemente o las telecomunicaciones necesarias para la correcta transmisión de datos entre los elementos que intervienen en el control”, es necesario decir que ha supuesto el grueso del trabajo, y que tal y como se desprende de los experimentos realizados y los videos grabados, ha sido un éxito. Se realizó el montaje Hardware del dron desde cero, además de remodelar todo el proceso de telecomunicación entre Optitrack y HMI, programando incluso un cliente-servidor para la gestión del tráfico de datos. Se consiguió asimismo la estabilidad en vuelo libre, por lo que este objetivo queda claramente superado.

Respecto al “Diseño del dispositivo y obtención de un prototipo capaz de controlar el dron” hay que decir que el prototipo inicialmente diseñado se reveló incapaz, pues no era adecuadamente reconocido por Optitrack. Hubo por tanto que cambiar este por un quadrotor con marcadores, que hacía las veces de Dispositivo. Con este método sí que se consiguió que el dron reaccionara a los cambios de actitud del Dispositivo y se realizó un control en rótula válido.

Se ha realizado también un diseño ayudado de Inventor para dotar a este prototipo de más atractivo, obteniéndose los archivos y los planos adjuntos. Se ha imprimido una reproducción a escala para mayor claridad.

Finalmente, con respecto a la “Realización de los ensayos necesarios para certificar el correcto funcionamiento de todos los diseños y el control efectivo del dron” es necesario señalar que con los experimentos de vuelo libre y vuelo en rótula con control del Dispositivo se dan por conseguidos el control efectivo del dron. Hay que decir también que no se ha realizado un ensayo en vuelo libre con control de dispositivo por precaución para preservar la integridad del dron.

Por lo tanto, se dan por conseguidos los objetivos que se plantearon al principio de esta memoria y se pasan a plantear ahora una serie de posibles vías para seguir trabajando y que pueden dar en resultados interesantes desde el punto de vista del control de quadrotors.

- Vuelo simultáneo de varios quadrotors con el mismo Optitrack. Efectivamente uno de los resultados de este TFG es la posibilidad de leer varios cuerpos rígidos y la gestión de estos datos en un servidor, que, convenientemente modificado, podría dar lugar a la gestión sin colisiones de varios drones simultáneamente.
- Incorporación de visión artificial para el control del dron junto con el Dispositivo.



- Sustitución del Optitrack por un sistema GPS, de tal manera que se pueda volar el dron fuera de la sala donde está instalado Optitrack. Esto de hecho ya es posible, sustituyendo en el dron las lecturas del Optitrack por las de la IMU.



Capítulo 7 Bibliografía y Webgrafía

- Amazon S.A. (Abril de 2018). *Amazon.es*. Obtenido de <https://www.amazon.es/Raspberry-Pi-Model-Placa-Quad-Core/dp/B00T2U7R7I>
- Amazon S.A. (2018). *Amazon.es*. Obtenido de <https://www.amazon.es/T-MOTOR-H%C3%A9lices-carbono-aeronaves-m%C3%BAltiples/dp/B010HEORBG>
- Arduino S.A. (2018). *Cooking-hacks.com*. Obtenido de <https://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge/>
- Balaguer Garín, V. (2015). *Sistema de Tiempo Real basado en Raspberry pi para el control de un quadrotor*. Trabajo de Fin de Grado, Universidad Politècnica de València, Departamento de automática e informática industrial (DISA)., Valencia.
- Castillo, García, Lozano, & Albertos, P. (2007). Modelado y Estabilización de un Helicóptero con Cuatro Rotores. *Revista Iberoamericana de automática e informática industrial*, 41-57.
- Castillo, Lozano, & Dzul. (2005). *Modelling and Control of Mini-Flying Machines*. Londres: Springer - Verlag.
- DJI S.A (2018). *icarushobby.com*. Obtenido de <http://www.icarushobby.com/product/dji-450-quadcopter-frame-kit/>
- Frasquet, A. C. (2014). *Desarrollo Integral de un Quadrotor: Diseño de un algoritmo de control para la posición X-Y basado en señales GPS*.
- Fu, K., Gonzalez, R., & Lee, C. (1987). *Robotics: control, sensing, vision, and intelligence*. New York: McGraw-Hill Book.
- Mistler, V., Benallegue, A., & M'Sirdi, N. K. (2001). *Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback*. IEEE Int. Workshop on Robot and Human Inter. Communic.
- Optitrack S.A. (2018). *youtube.com*. Obtenido de <https://www.youtube.com/watch?v=1e6Qqxqe-k0>
- Raffo, G. V. (2007). *Modelado y Control de un Helicóptero quadrotor*. Universidad de Sevilla, Departamento de telemática automática y robótica, Sevilla.
- Randstad S.A. (2018). *Informe de Tendencias Salariales*. Randstad.
- Stojche Deskovski, V. S. (2016). *Deskovski S., Sazdovski V., Gacovski Z. (2016) Guidance Laws and Navigation Systems for Quadrotor UAV: Theoretical and Practical Findings. In: Dimirovski G. (eds) Complex Systems. Studies in Systems, Decision and Control, vol 55. Springer, Cham.*



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

ANEXO I: CÓDIGO DEL SERVIDOR



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



```
//===== file = mclient.c =====  
//= A multicast client to receive multicast datagrams  
=  
//=====  
//= Notes:  
=  
//= 1) This program receives on a multicast address and outputs the  
=  
//= received buffer to the screen.  
=  
//= 2) Conditionally compiles for Winsock and BSD sockets by  
setting the =  
//= initial #define to WIN or BSD as appropriate.  
=  
//= 3) The multicast group address is GROUP_ADDR.  
=  
//-----  
-----=  
//= Build: bcc32 mclient.c or cl mclient.c wsock32.lib for Winsock  
=  
//= gcc mclient.c -lsocket -lnsl for BSD  
=  
//-----  
//= History: JNS (07/11/02) - Genesis  
//= Modified for Optitrack by Conrado García Salinas  
//-----  
#define WIN // WIN for Winsock and BSD for BSD sockets  
  
//----- Include files -----  
  
#include <stdio.h> // Needed for printf()  
#include <stdlib.h> // Needed for memcpy()  
#ifdef WIN  
#include <winsock.h> // Needed for all Windows stuff  
#endif  
#ifdef BSD  
#include <sys/types.h> // Needed for system defined identifiers.  
#include <netinet/in.h> // Needed for internet address structure.  
#include <sys/socket.h> // Needed for socket(), bind(), etc...  
#include <arpa/inet.h> // Needed for inet_ntoa()  
#include <fcntl.h>  
#include <netdb.h>  
#endif  
  
//----- Defines -----  
  
#define PORT_NUM 1511 //PORT TO LISTEN TO OPTITRACK  
#define PORT_SERVER 1557 // PORT TO SEND TO DRONE  
#define GROUP_ADDR "239.255.42.99" // Address of the multicast group  
#define MAX_LEN 512  
#define SERVER "192.168.3.100" //Ip where we want to send our message  
#define NUM_RIG_OBJ 1 // Number of rigid objects detected by Optitrack
```



```
//===== Main program =====

void main(void)
{
#ifdef WIN
    WORD wVersionRequested = MAKEWORD(1,1); // Stuff for WSA functions
    WSADATA wsaData; // Stuff for WSA functions
#endif

//-----server stuff-----

    unsigned int multi_server_sock; // Multicast socket
    struct ip_mreq mreq; // Multicast group structure
    struct sockaddr_in client_addr; // Client Internet address
    unsigned int addr_len; // Internet address length
    unsigned char buffer[MAX_LEN]; // Datagram buffer
    int retcode; // Return code

//-----client stuff-----

    int s; //socket descriptor
    struct sockaddr_in server; //struct for the client
    unsigned char send_str[MAX_LEN]; //message to Drone
    unsigned int send_len; //length of this message

//-----optitrack stuff-----

    int md=0;
    int i,j,ID;
    char *ptr = buffer;
    float x[NUM_RIG_OBJ]= {0};
    float y[NUM_RIG_OBJ]= {0};
    float z[NUM_RIG_OBJ]= {0};
    float q1[NUM_RIG_OBJ]= {0};
    float q2[NUM_RIG_OBJ]= {0};
    float q3[NUM_RIG_OBJ]= {0};
    float q4[NUM_RIG_OBJ]= {0};
    float xa,ya,za,qx,qy,qz,qw;

//-----things to do to set up the server-----

#ifdef WIN
    // This stuff initializes winsock
    WSStartup(wVersionRequested, &wsaData);
#endif

    // Create a multicast socket and fill-in multicast address
    information

    multi_server_sock=socket(AF_INET, SOCK_DGRAM,0);
    mreq.imr_multiaddr.s_addr = inet_addr(GROUP_ADDR);
    mreq.imr_interface.s_addr = INADDR_ANY;
    printf("multicast socket created \n");
```



```
// Create client address information and bind the multicast socket

client_addr.sin_family = AF_INET;
client_addr.sin_addr.s_addr = INADDR_ANY;
client_addr.sin_port = htons(PORT_NUM);
retcode = bind(multi_server_sock, (struct sockaddr *)&client_addr,
               sizeof(struct sockaddr));

if (retcode < 0)
{
    printf("*** ERROR - bind() failed with retcode = %d \n", retcode);
    return;
}
printf("client address and bind ok \n");

// Have the multicast socket join the multicast group

retcode = setsockopt(multi_server_sock, IPPROTO_IP,
IP_ADD_MEMBERSHIP,
                   (char*)&mreq, sizeof(mreq)) ;
if (retcode < 0)
{
    printf("*** ERROR - setsockopt() failed with retcode = %d \n",
retcode);
    return;
}
printf("multicast group joined \n");
// Set addr_len

addr_len = sizeof(client_addr);

//-----Things to do to set up the client-----

// Create a datagram socket in the internet domain and use the default
protocol (UDP).

if ((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0)
{
    perror("socket()");
    exit(1);
}
// Set up the server name
server.sin_family = AF_INET; // Internet Domain
server.sin_port = htons(PORT_SERVER); // Server
Port
server.sin_addr.s_addr = inet_addr(SERVER); // Server's Address

//-----LOOP-----

memset(buffer, 0, sizeof(buffer)); //Cleaning the buffer for receiving
memset(send_str, 0, sizeof(send_str)); //Cleaning the message
int c=0;
while(1)
{
```



```
//-----RECEIVE FROM OPTITRACK-----

// Receive a datagram from the multicast server
//   printf("waiting for data \n");

    if( (retcode = recvfrom(multi_server_sock, buffer, sizeof(buffer),
0,
    (struct sockaddr *)&client_addr, &addr_len)) < 0){
        printf("*** ERROR - recvfrom() failed \n");
        printf("recvfrom() failed with error code : %d" ,
WSAGetLastError());
        exit(1);
    }

    // Output the received buffer to the screen as a string
    //   for(i=0;i<120;i++){
    //   printf("%i %d\n",i, buffer[i]);}
    // TRANSFORM THIS INFO IN SOMETHIN LEGIBLE FOR THE DRONE

    //rigid body pos/ori

ptr = buffer;
    memcpy(&md, ptr,2); ptr += 2;

    if (md==7){
        // FRAME OF MOCAP DATA packet

        int nBytes = 0;
        memcpy(&nBytes, ptr, 2); ptr += 2;

    //   printf("Byte count : %d\n", nBytes);

        // Next 4 Bytes is the frame number
        int frameNumber = 0; memcpy(&frameNumber, ptr, 4); ptr += 4;
    //   printf("Frame # : %d\n", frameNumber);

        // Next 4 Bytes is the number of data sets (markersets,
        rigidbodies, etc)
        int nMarkerSets = 0; memcpy(&nMarkerSets, ptr, 4); ptr += 4;
    //   printf("Marker Set Count : %d\n", nMarkerSets);

        // Loop through number of marker sets and get name and data
        for (int i=0; i < nMarkerSets; i++)
        {
            // Markerset name
            char szName[256];
            //strcpy_s(szName, ptr);
            int nDataBytes = (int) strlen(szName) + 1;
            ptr += nDataBytes;
    //   printf("Model Name: %s\n", szName);

            // marker data
            int nMarkers = 0; memcpy(&nMarkers, ptr, 4); ptr += 4;
    //   printf("Marker Count : %d\n", nMarkers);
```



```
    for(int j=0; j < nMarkers; j++)
    {
        float x = 0; memcpy(&x, ptr, 4); ptr += 4;
        float y = 0; memcpy(&y, ptr, 4); ptr += 4;
        float z = 0; memcpy(&z, ptr, 4); ptr += 4;
//        printf("\tMarker %d :
[x=%3.2f,y=%3.2f,z=%3.2f]\n", j, x, y, z);
    }
}

// Loop through unlabeled markers
int nOtherMarkers = 0; memcpy(&nOtherMarkers, ptr, 4); ptr +=
4;

// OtherMarker list is Deprecated
//printf("Unidentified Marker Count : %d\n", nOtherMarkers);
for(int j=0; j < nOtherMarkers; j++)
{
    float x = 0.0f; memcpy(&x, ptr, 4); ptr += 4;
    float y = 0.0f; memcpy(&y, ptr, 4); ptr += 4;
    float z = 0.0f; memcpy(&z, ptr, 4); ptr += 4;

    // Deprecated
    //printf("\tMarker %d : pos =
[%3.2f,%3.2f,%3.2f]\n", j, x, y, z);
}

// Loop through rigidbodies
int nRigidBodies = 0;
memcpy(&nRigidBodies, ptr, 4); ptr += 4;
//    printf("Rigid Body Count : %d\n", nRigidBodies);
for (int j=0; j < nRigidBodies; j++)
{
    // Rigid body position and orientation
    int ID = 0; memcpy(&ID, ptr, 4); ptr += 4;
    float xa = 0.0f; memcpy(&xa, ptr, 4); ptr += 4;
    float ya = 0.0f; memcpy(&ya, ptr, 4); ptr += 4;
    float za = 0.0f; memcpy(&za, ptr, 4); ptr += 4;
    float qx = 0; memcpy(&qx, ptr, 4); ptr += 4;
    float qy = 0; memcpy(&qy, ptr, 4); ptr += 4;
    float qz = 0; memcpy(&qz, ptr, 4); ptr += 4;
    float qw = 0; memcpy(&qw, ptr, 4); ptr += 4;

    x[j]=xa;
    y[j]=ya;
    z[j]=za;
    q1[j]=qx;
    q2[j]=qy;
    q3[j]=qz;
    q4[j]=qw;

//    printf("End Packet\n-----\n");
//printf("dentro packed \n");
```



```
//-----SEND THIS NEW INFO TO THE DRON (CLIENT PART)-----

//for (i=0;i<MAX_LEN;i++)
//{send_str[i]=buffer[i];}

//printf("sending to drone with direction "SERVER" \n");

//send_len=(strlen(send_str)+1);
if (c==20){
    send_len=0;
    ptr = send_str;
    for (int i=0;i<NUM_RIG_OBJ;i++){

        memcpy(ptr, &i, 4); ptr += 4;
        memcpy(ptr, &x[i], 4); ptr += 4;
        memcpy(ptr, &y[i], 4); ptr += 4;
        memcpy(ptr, &z[i], 4); ptr += 4;
        memcpy(ptr, &q1[i], 4); ptr += 4;
        memcpy(ptr, &q2[i], 4); ptr += 4;
        memcpy(ptr, &q3[i], 4); ptr += 4;
        memcpy(ptr, &q4[i], 4); ptr += 4;
        send_len+=8*4;
    }
printf("pos: [%3.2f,%3.2f,%3.2f]\n", x[0],y[0],z[0]);
printf("ori: [%3.2f,%3.2f,%3.2f,%3.2f]\n", q1[0], q2[0], q3[0],
q4[0]);

printf("length send %d \n",send_len);
    if (sendto(s, send_str, send_len, 0,
        (struct sockaddr *)&server, sizeof(server)) < 0)
    {
        perror("sendto()");
        exit(2);
    }
printf ("mensaje enviado: %s \n",send_str);

//    memset(buffer, 0, sizeof(buffer)); //Cleaning the buffer for
//next datagram
//
//    memset(send_str, 0, sizeof(send_str)); //Cleaning the message

c=0; }
c++;

}
}
// Close and clean-up
#ifdef WIN
    closesocket(multi_server_sock);
    WSACleanup();
#endif
#ifdef BSD
    close(multi_server_sock);
#endif
}
```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

ANEXO II: PLANOS DEL DISPOSITIVO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

A continuación, se adjuntan los planos generales del Dispositivo de Control. Al ser únicamente un plano el que se adjunta, no se estima oportuna la creación de un índice de planos. Dicho elemento ha sido desarrollado a partir del diseño original 3D y con la colaboración de la misma empresa que se ha hecho cargo de la impresión 3D a escala del dispositivo: Printable 3D.

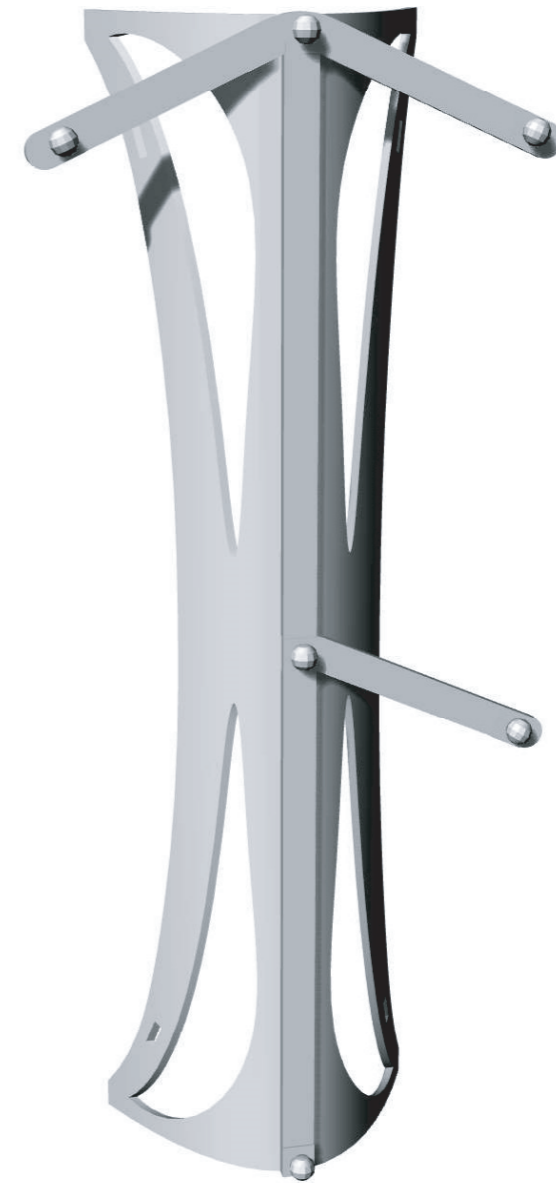
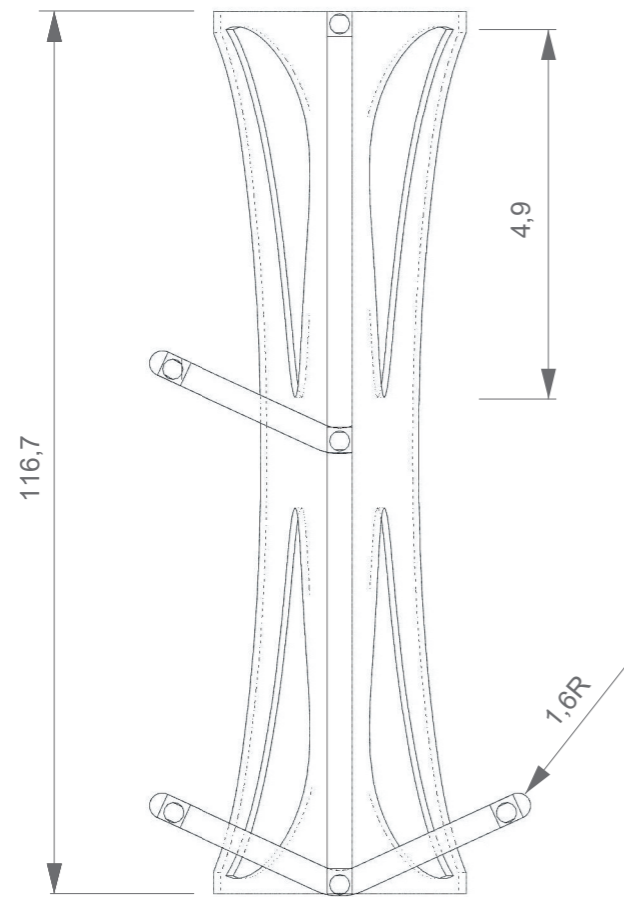
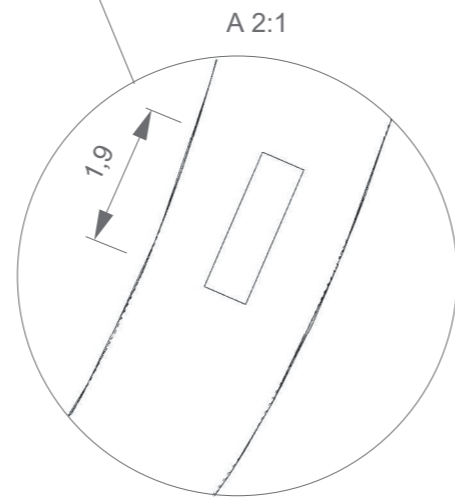
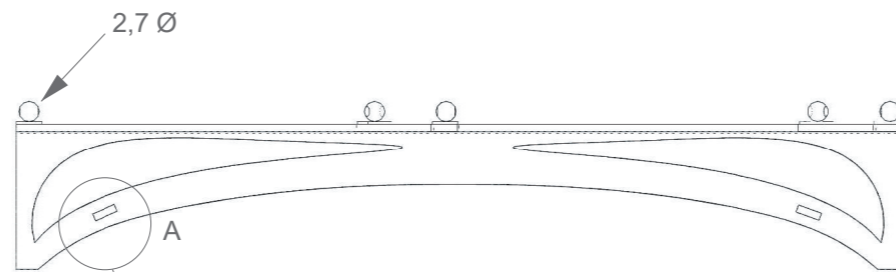
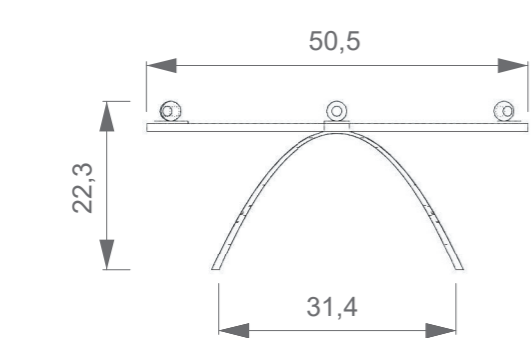


UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA





UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

PRESUPUESTO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Diseño configuración e implementación de un dispositivo basado en el sistema de posicionamiento Optitrack para el control integral de un quadrotor.



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA



Capítulo 1 Introducción

El siguiente presupuesto forma parte del Trabajo de Fin de Grado “Diseño e implementación de un dispositivo de control basado en Optitrack para un quadrotor”. Este documento consta de un solo capítulo contable: En el cual se desglosan las cuatro unidades de obra en las que se descompone el proyecto. Antes de eso se aportan dos cuadros: uno de elementos materiales y otro de mano de obra.

Capítulo 2 Cuadro de precios de materiales y mano de obra

En cuanto a los materiales utilizados, se exponen separadamente los elementos relativos al Quadrotor, al Optitrack y al ordenador usado durante todo el proceso. Además de esto, se ha utilizado una rúcula ya existente en el laboratorio, cuyo precio se ha estimado gracias a (Balaguer Garín, 2015) Los precios por hora del ordenador y la fuente de alimentación se han obtenido dividiendo su valor de mercado entre el total de horas utilizadas. Teniendo en cuenta que en este TFG se han invertido aproximadamente 5 meses con una dedicación de 4 horas diarias, lo que da un total aproximado de 400 horas, y que se supone un precio para el ordenador de gama media con dos monitores de 1000€ y una fuente de alimentación de 250€, se obtienen los resultados expuestos en la tabla siguiente

CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO
Q.RASPI	u	Raspberry PI 2 Model b	35 €
Q.COOK	u	Shield Raspberry pi to Arduino	40 €
Q.MARCO	u	Marco DJI F550	39 €
Q.IMU	u	IMU 10 DOF + placa de circuitería	30 €
Q.CONV	u	Conversor tensión 12 - 5 V	4,21 €
Q.HELIC	u	Helices de plástico 16 "	2,5 €
QMOT	u	Motor brushless T motor 2216-12	44,99 €
Q.CABLEADO Y TORNILLERIA	u	cableado y tornillos varios	40 €
Q.WIFI	u	Wifi Extendido	9,8 €
Q.DRIVER	u	Driver YGE 30i	25,95 €
Q.TARJ	u	Tarjeta Sd 8 GB	7,16 €
Q.SOFT	u	Software Ai2	0 €
Q.ROT	u	Rotula	246,12 €
Q.ALIM	h	Fuente de alimentacion regulable	7,5 €
O.CAM	u	Cámaras Optitrack	600 €
O.PROC	u	Unidad de procesamiento Optitrack	1100 €
O.LIC	u	Licencia de uso de Motive	800 €
O.CON	u	Conexionado del sistema Optitrack	50 €
MA.ORD	h	Ordenador de gama media	2,5 €

Fig. Cap 2-1 Cuadro de precios de materiales



Es necesario hacer notar que no se han incluido los precios de las licencias de Matlab o Inventor. Esto es así porque la UPV ofrece esta licencia gratis a sus estudiantes, y por tanto, no se considera que tenga un costo real para el presente proyecto en particular.

CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO
MO.LIC	h	Graduado en ingeniería de tecnologías Industriales	35 €
MO.TEC	h	Técnico	25 €
MO.IMP	h	Hora de impresión 3D	7 €

Fig. Cap 2-2 Cuadro de precios de Mano de Obra

El cálculo de los precios por hora del Graduado en Ingeniería de Tecnologías Industriales y del Técnico de laboratorio se han obtenido mediante la consulta de varios proyectos y estudios salariales. Finalmente se han tomado como valores de cálculo los reflejados en el estudio sobre tendencias salariales de la empresa Randstad (Randstad S.A, 2018)

Dicho informe nos provee de una banda de salarios por cada uno de los puestos de trabajo. Se ha considerado que el Graduado está realizando un trabajo de “Técnico I+D” y el Técnico de Laboratorio la labor del mismo nombre. Para el Técnico de laboratorio la banda se sitúa para la zona geográfica de Valencia entre los 27.500 € y los 41.500 € anuales. Tomando un valor medio de 35.000 € los cuales, repartidos en 14 pagas mensuales, a razón de 80 horas mensuales (media jornada) nos da el valor arriba utilizado. Con idéntico cálculo y partiendo de 25.000€ se obtiene el valor del precio por hora del Técnico de laboratorio, cuya banda de salarios es de 22.000 €-33.000 €.

Para el cálculo de la hora de impresión 3D se ha recurrido a la empresa Printable 3D, con sede en la UPV

Capítulo 3 Cuadro de precios unitarios

Para la realización de este presupuesto se han diferenciado las siguientes Unidades de Obra

CODIGO	UNIDADES	DESCRIPCIÓN	PRECIO
UO.1	u	Montaje Quadrotor	2.873,50 €
UO.2	u	Instalación de Optitrack	6.283,00 €
UO.3	u	Creación servidor	1.596,50 €
UO.4	u	Diseño del Dispositivo de Control	57,68 €

Fig. Cap 3-1 Cuadro de precios Unitarios

Capítulo 4 Cuadro de precios descompuestos

A continuación, se pasan a desglosar las unidades de obra:

CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO	RENDIMIENTO	IMPORTE
Q.RASPI	u	Raspberry PI 2 Model b	35,00 €	1	35,00 €
Q.COOK	u	Shield Connection Bridge Raspberry pi to Arduino	40,00 €	1	40,00 €
Q.MARCO	u	Marco DJI F550	39,00 €	1	39,00 €
Q.IMU	u	IMU 10 DOF MP6050 + placa de circuitería para montaje	30,00 €	1	30,00 €
Q.CONV	u	Convertor tensión 12 to 5 V	4,21 €	1	4,21 €
QMOT	u	Motor brushless T motor	44,99 €	4	179,96 €
Q.HELIC	u	Helices de plástico 16 "	2,50 €	4	10,00 €
Q.WIFI	u	Wifi Extendido	9,80 €	1	9,80 €
Q.DRIVER	u	Driver YGE 30i	25,95 €	4	103,80 €
Q.TARJ	u	Tarjetsa Sd 8 GB	7,16 €	1	7,16 €
Q.SOFT	u	Software Ai2	- €	1	- €
Q.ROT	u	Rotula	246,12 €	1	246,12 €
Q.CABLEADO Y TORNILLERIA	u	cableado y tornillos varios	40,00 €	1	40,00 €
MO.LIC	h	Graduado en ingeniería de tecnologías Industriales	31,25 €	60	2.100,00 €
		Subtotal			2.845,05 €
		Gastos complementarios 1%			28,45 €
		Costes indirectos indirectos 2%			56,90 €
		Total Unidad de Obra			2.930,40 €

Fig. Cap 4-1 UO.1



CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO	RENDIMIENTO	IMPORTE
O.CAM	u	Cámaras Optitrack	600,00 €	6	3.600,00 €
O.PROC	u	Unidad de procesamiento Optitrack	1.100,00 €	1	1.100,00 €
O.LIC	u	Licencia de uso de Motive	800,00 €	1	800,00 €
O.CONEX	u	Conexión del sistema Optitrack	50,00 €	1	50,00 €
MO.TEC	h	Tecnico	22,32 €	15	334,80 €
MO.LIC	h	Graduado en ingeniería de tecnologías Industriales	31,25 €	5	156,25 €
Subtotal					6.041,05 €
Gastos complementarios 1%					60,41€
Costes indirectos indirectos 2%					120,82 €
Total Unidad de Obra					6.222,28 €

Fig. Cap 4-2 UO.2



CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO	RENDIMIENTO	IMPORTE
MA.ORD	h	Ordenador de gama media	2,50 €	60	150,00 €
MO.LIC	h	Graduado en ingeniería de tecnologías Industriales	31,25 €	40	1.250,00 €
Subtotal					1.400,00 €
Gastos complementarios 1%					14,00 €
Costes indirectos indirectos 2%					28,00 €
Total Unidad de Obra					1.442,00 €

Fig. Cap 4-3 UO.3

CÓDIGO	UNIDADES	DESCRIPCIÓN	PRECIO	RENDIMIENTO	IMPORTE
I.IMPRESIÓN	h	hora de impresión 3D	7,00 €	8	56,00 €
MA.ORD	h	Ordenador de gama media	2,50 €	10	25,00 €
MO.LIC	h	Graduado en ingeniería de tecnologías Industriales	31,25 €	10	312,50 €
Subtotal					56,00 €
Gastos complementarios 1%					0,56 €
Costes indirectos indirectos 2%					1,12 €
Total Unidad de Obra					57,68 €

Fig. Cap 4-4 UO.4



Capítulo 5 Presupuesto Parcial

CODIGO	UNIDADES	DESCRIPCIÓN	PRECIO	CANTIDAD	IMPORTE TOTAL
UO.1	u	Montaje Quadrotor	2.698,65 €	1	2.698,65 €
UO.2	u	Instalación de Optitrack	6.222,28 €	1	6.222,28 €
UO.3	u	Creación servidor	1.596,50 €	1	1.596,50 €
UO.4	u	Diseño del Dispositivo de Control	57,68 €	1	57,68 €

Fig. Cap 5-1 Presupuesto Parcial

Capítulo 6 Presupuesto de ejecución por contrata

CAPITULOS	PRECIO
CAP1	10.420,61 €
TOTAL	10.420,61 €

PRESUPUESTO DE EJECUCIÓN MATERIAL	10.420,61 €
GASTOS GENERALES 15%	1.563,09 €
BENEFICIO INDUSTRIAL 6%	625,24 €
PRESUPUESTO DE EJECUCIÓN POR CONTRATA	12.608,94 €
IVA 21%	2.647,88 €
PRESUPUESTO BASE LICITACIÓN	15.256,82 €

Fig. Cap 6-1 Presupuesto Base de licitación

Asciende el presente presupuesto a la expresada cantidad de:

QUINCE MIL DOSCIENTOS CINCUENTA Y SEIS EUROS CON OCHENTA Y DOS CÉNTIMOS.