

# Disseny, control i implementació d'un Quadcòpter

Ana Maria Corraliza Jimenez, Ruben Recasens Orós i Albert Garcia Junyent

Estudiants d'enginyeria electrònica industrial i automàtica a la universitat politècnica de Catalunya de Vilanova i la Geltrú

## Resum

En aquest projecte s'ha portat a terme un estudi sobre el model matemàtic d'un quadcòpter mitjançant les seves equacions diferencials, per així posteriorment aconseguir una planta utilitzant Matlab Simulink. Un cop aconseguit aquest objectiu s'ha dissenyat un controlador en realimentació d'espai d'estats. Per comprovar aquest primer estudi s'han fet un seguit de simulacions extraient el seu comportament. Tot seguit s'ha programat un mòdul Arduino amb l'esquema del controlador i s'ha construït un primer prototip d'un multicòpter. Finalment, s'han integrat perifèrics de visió, localització i WiFi, possibilitant el seu control i seguiment en temps real mitjançant Android i Matlab.

## 1. Introducció

En aquest article titulat Disseny, control i implementació d'un Quadcòpter, de tres estudiants de l'Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú és part del seu projecte final de grau realitzat durant el segon quadrimestre del curs 2013-14.

Cal fer notar que durant quan es parli de l'OVI es tractarà del quadcòpter dissenyat i controlat en el projecte. El nom OVI ha sortit a partir de buscar un possible nom pel quadcòpter, i així es va arribar a OVI: Objecte Volador Identificat.

S'han estudiat les parts relacionades amb el modelat, simulació, implementació i posterior control d'un quadcòpter.

S'ha començat per l'estudi del model de la planta física d'un quadcòpter a partir de les seves equacions diferencials per després poder obtenir una simulació acurada mitjançant Matlab-Simulink del seu comportament.

S'ha tancat el llaç de control amb una realimentació amb espai d'estats discreta i s'ha simulat la seva resposta, obtenint així un controlador en espai d'estats. Aquest controlador actuen sobre variables angulars i altura, i posteriorment s'ha implementat en un microcontrolador, muntat a un quadcòpter real.

Una altra part important del treball ha estat la programació de varies aplicacions Android, Wifi i de Radio-control des de zero. D'aquesta manera s'ha aconseguit saber en tot moment tant el seu comportament real com la seva posició en un mapa de Global Positioning System (GPS).

Tot el treball s'ha desenvolupat mitjançant la metodologia àgil d'SCRUM i s'ha anat recopilant al blog d'Agile d'EPSEVG [6].

## 2. Modelat del sistema

El primer pas per obtenir un controlador per la planta és arribar a un model matemàtic del seu comportament, d'aquesta manera es podran entrar unes consignes d'angles i velocitats a aquestes equacions i s'obtidran els resultats de trajectòria de l'OVI per tal de poder basar-se en aquesta a l'hora de fer tots els càlculs i proves adients.

### 2.1 Equacions de la planta

Per obtenir el model matemàtic de la planta del quadcòpter s'han estudiat les seves equacions diferencials, concretament les equacions de forces i parells [1]. Aquestes dos equacions són les que ens parametritzen els principals comportaments de l'estructura, les equacions relacionades amb les forces ens donen una idea de quines són les necessàries per mantenir la sustentació i controlar l'alçada de l'OVI. Mentre que les equacions relacionades amb els parells controlen el gir del sistema, ja que en un principi no es vol que hi hagi cap tipus de gir en cap eix.

A partir de dues referències d'eixos com es pot observar a la Fig1, una mòbil solidària al multicòpter ( $X_B, Y_B, Z_B$ ) i una altra fixa arran de terra ( $X, Y, Z$ ), s'han definit sis variables: tres posicions lineals i tres posicions angulars.

També és important obtenir una matriu de transformació entre les dues referències d'eixos, d'aquesta manera es pot fer un canvi de coordenades en tot moment segons les necessitats de l'estudi, això és degut a que alguns paràmetres estan identificats en un eix i la seva utilització s'ha d'establir en l'altre sistema de coordenades per aquest tema és important saber com passar aquests paràmetres d'un a l'altre.

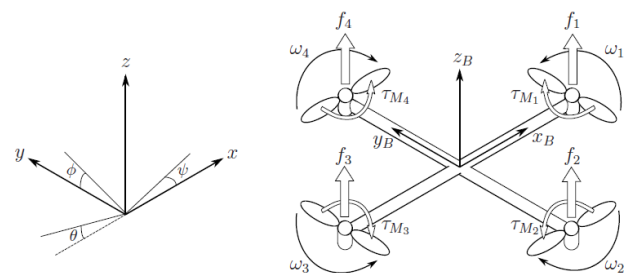


Fig 1: Esquema de les referències fixa i mòbil

Algun dels paràmetres que és poden veure a la figura, necessàries per tot l'estudi són els angles respecte l'eix fixa ( $\Phi, \theta, \psi$ ) ja que ens donaran informació de la inclinació del sistema. També observem les forces de torsió ( $T_M$ ) creades per la rotació dels motors i les hèlix. També com a informació mostra les direccions en cada eix de les velocitats angulars dels motors ( $\omega$ ) i les forces creades pel sistema ( $f$ ).

Amb les matrius d'inèrcia, velocitats i acceleracions (lineals i angulars) s'han obtingut dues equacions diferencials on es tenen en consideració les forces i parells que pateix l'estructura en cadascun dels tres eixos: X, Y i Z.

Per a la resolució del modelat del sistema obtingut s'ha seguit el mètode d'Euler-Lagrange (1), on es relacionen les energies del sistema.

$$L(q, \dot{q}) = E_{trans} + E_{rot} - E_{pot} \quad (1)$$

Ja que les components lineals i angulars no depenen unes de les altres, s'han estudiat de forma separada. Així, les primeres tenen relació amb les forces (3) composta per la multiplicació entre la massa i les acceleracions en cada eix ( $\ddot{\xi}$ ) i sumades a la gravetat, les segones tenen relació amb els parells (5) on es multiplica el Jacobia per les acceleracions angulars ( $\ddot{\eta}$ ) i se li sumen les perturbacions produïdes per temes de Coriolis [2] multiplicades per les velocitats angulars ( $\dot{\eta}$ ).

$$\xi = [X; Y; Z] \quad (2)$$

$$f = m * \ddot{\xi} + m * g * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

$$\eta = [\Phi; \Theta; \Psi] \quad (4)$$

$$\tau_B = J * \ddot{\eta} + C(\eta, \dot{\eta}) * \dot{\eta} \quad (5)$$

Un cop obtingudes les 6 equacions necessàries per tenir resultats del comportament del OVI, es podrà portar a terme l'estudi de les reaccions de la planta sota unes consignes determinades

## 2.2 Simulació de la planta

Enteses les 6 equacions diferencials que corresponen al comportament de la planta, s'ha passat a un esquema en Matlab Simulink per poder realitzar diferents proves.

Primer s'han escrit les dues equacions en dos blocs diferents, desglossant les forces en components lineals i els parells en components angulars.

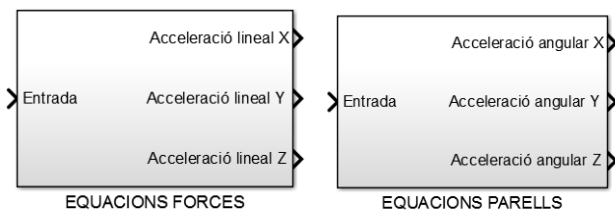


Fig 2: Blocs de Simulink on s'han desglossat les equacions

Seguidament s'han obtingut les 18 variables, les 6 corresponents a posicions angulars i lineals més de les obtingudes integrant les sortides de cadascun dels blocs dues vegades, per obtenir velocitats i posicions, tant lineals com angulars.

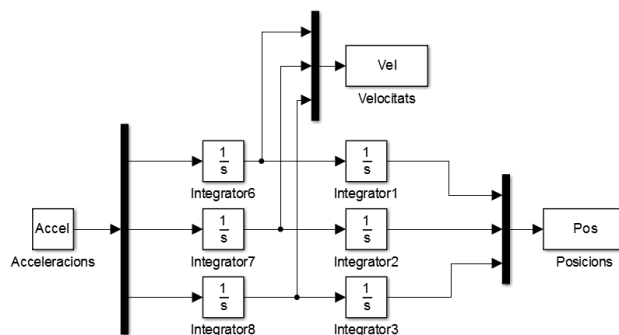


Fig 3: Esquema d'integradors per extreure variables d'estat

## 3. Disseny del controlador

Per aconseguir un correcte control de la planta s'ha escollit un mètode basat en espai d'estats[3]. Aquest mètode es basa en l'obtenció d'un sistema físic com un conjunt d'entrades, sortides i variables d'estat relacionades entre si per un sistema d'equacions diferencials.

En aquest cas, essent un sistema no lineal, s'ha decidit portar a terme una linealització per facilitar l'obtenció del controlador.

### 3.1. Linealització de la planta

La planta del quadricòpter és no lineal ja que les sortides no estan directament afectades per les entrades. Per exemple, sota una entrada esglau no s'obté la sortida habitual d'un sistema de primer ordre davant aquest tipus de consigna. Per aquesta raó s'ha hagut de linealitzar el model de la planta. El mètode seguit ha estat la "Linealització aproximada al voltant d'un punt d'equilibri", aconseguint entendre el comportament d'un sistema no lineal al ser pertorbat al voltant del punt d'equilibri.

$$\dot{x} = f(x, u) \rightarrow \dot{x} = Ax + Bu \quad (6)$$

Per dur a terme aquesta linealització s'han seguit els següents passos:

- De les equacions no lineals s'han separat els components que depenen de les entrades dels que no en depenen.
- S'ha fet el Jacobia de les components que no depenen de les entrades respecte les variables d'estat (x). Amb aquest Jacobia s'ha trobat la matriu A
- S'ha fet el Jacobia de les components que depenen de les entrades respecte aquestes entrades (u). Amb aquest Jacobia s'ha trobat la matriu B

Les matrius del sistema continu linealitzat són les següents

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -0.00011 & 0 & 0.00011 \\ 0 & 0 & 0 & 0 \\ -0.00016 & 0 & 0.00016 & 0 \\ 0 & 0 & 0 & 0 \\ -0.000013 & 0.000013 & -0.000013 & 0.000013 \\ 0 & 0 & 0 & 0 \\ 0.000011 & 0.000011 & 0.000011 & 0.000011 \end{pmatrix}$$

La matriu C depèn de les variables presents a la sortida del sistema. En el d'aquest model és possible obtenir totes les variables d'estat del sistema a partir de la planta, per tant la matriu C és una matriu Identitat de 8x8.

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

En sistemes reals, la matriu D és igual a 0.

### 3.2. Discretització de la planta

El següent pas ha sigut el de discretitzar la planta a partir de les matrius obtingudes i un període de mostreig de 15ms escollit per temes de refresc de les consignes que posteriorment haurà de donar el microcontrolador.

Sabent que es pot obtenir la matriu d'estat discreta a partir de:

$$\Phi = e^{Ah} \quad (7)$$

$$\Gamma = \int_0^h e^{As} * B ds \quad (8)$$

El sistema discret en espai d'estats es pot aconseguir mitjançant la comanda "c2d" de Matlab, aquesta consisteix en convertir les matrius contínues en les seves equivalents discretes a partir de les fórmules esmentades (7) i (8) utilitzant un període de mostreig.

### 3.3. Obtenció del controlador

Trobades les matrius que defineixen el comportament del cos, s'ha realitzat una realimentació tancant el laç de control amb un primer controlador, trobat a partir d'uns pols determinats. El mètode utilitzat és el de ubicació de pols, aquest es basa en igualar el polinòmic característic de l'equació de la planta en llaç tancat a un polinomi amb uns pols que creen el comportament desitjat.

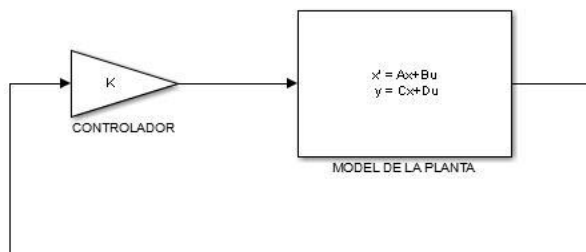


Fig 4: Llaç tancat de la planta amb el controlador

Durant la elaboració d'aquest projecte s'ha vist que el sistema amb el que es treballa es de tipus MIMO (Multiple Input Multiple Output), això significa que hi ha més d'una

entrada i més d'una sortida. Durant la titulació no s'havia treballat mai amb aquest tipus de sistemes pel qual durant la creació del controlador s'ha estudiat quins són els mètodes de treball amb aquesta peculiaritat.

S'ha vist que mitjançant l'ordre de Matlab "place" s'aconsegueix un controlador amb la tècnica d'ubicació de pols per a sistemes MIMO, ja que la comanda "acker" només permet els sistemes simple input.

El controlador obtingut mitjançant "place" és:

$$K = \begin{pmatrix} 0 & 0 & -40000 & -75000 & -16334.00 & -28680 & 1984.200 & 348.400 \\ -65000.00 & -125000 & 0 & 0 & 16334.00 & 28680 & 1984.200 & 348.400 \\ 0 & 0 & 40000 & 75000 & -16334.00 & -28680 & 1984.200 & 348.400 \\ 65000.00 & 125000 & 0 & 0 & 16334.00 & 28680 & 1984.200 & 348.400 \end{pmatrix}$$

D'aquesta manera s'ha trobat quin es el comportament realimentant els estats i s'han ajustat els seus paràmetres.

### 3.4. Tracking

Després de vaires reiteracions buscant el control més acurat, s'ha afegit un sistema de tracking, per aconseguir que la resposta final del sistema tendeixi a una consigna de referència determinada. Per dur a terme el seguiment s'afegeix una matriu Nx que converteix la senyal de referència a un conjunt d'estats d'equilibri. A més s'ha afegit una component integral de l'error de sortida respecte el de referència per contrarestar l'error en règim estacionari d'aquesta manera se suavitzen les pertorbacions i s'evita en més mesura la inestabilització.

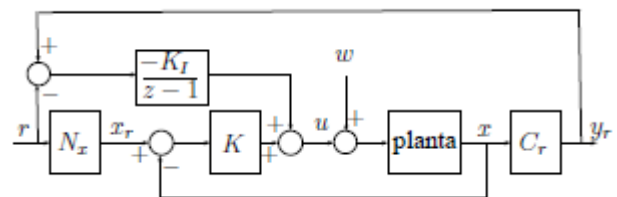


Fig 5: Esquema sistema discret en espai d'estats amb controlador tracking i integral de l'error

### 3.5. Simulacions

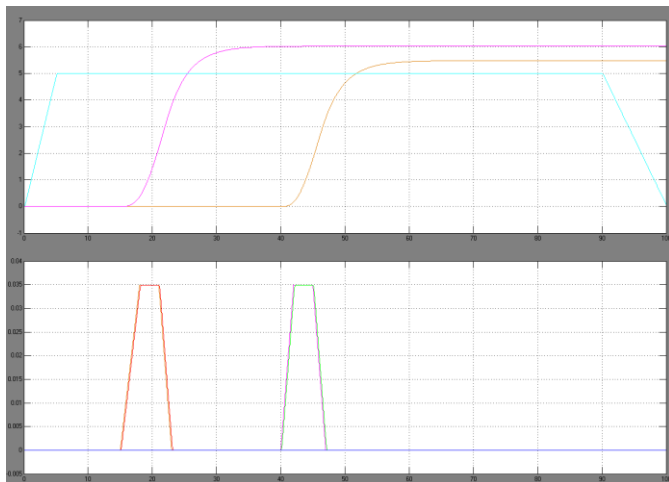
Tenint el sistema implementat en blocs en un Simulink i un controlador integrat a aquest, tancant la realimentació en espai d'estats, s'han arribat a les següents simulacions.

Totes les proves s'han fet amb una consigna a l'eix Z per arribar a una alçada de 5 metres, tot seguit s'ha efectuat una inclinació de 2 graus en l'eix X seguit d'una inclinació també de 2 graus en eix Y.

En les gràfiques que és mostraran és pot veure en la gràfica superior les posicions lineals de la planta un cop s'ha aplicat la senyal esmentada. En cian veiem la posició en Z, arribant a la posició desitjada de 5 metres en el temps que s'ha establert. En magenta veiem un moviment en Y, provocat per la inclinació en l'eix X de 2 graus. I en taronja un moviment en X, provocat per la inclinació en l'eix Y de 2 graus. Tant mateix en la gràfica inferior és veuen les posicions angular dels eixos, la primera forma correspon al gir en X i la segona al gir en Y, mentre que l'eix Z no pateix cap canvi d'orientació.

### Resposta simulada

En aquesta primera simulació un cop el sistema ha estat discretitzat amb el període de mostreig esmentat abans (15ms) s'observa que el comportament del quadricòpter és perfecte ja que segueix les trajectòries marcades amb una gran exactitud. Aquesta prova és la que ha demostrat que el control de la planta ha estat correcte.



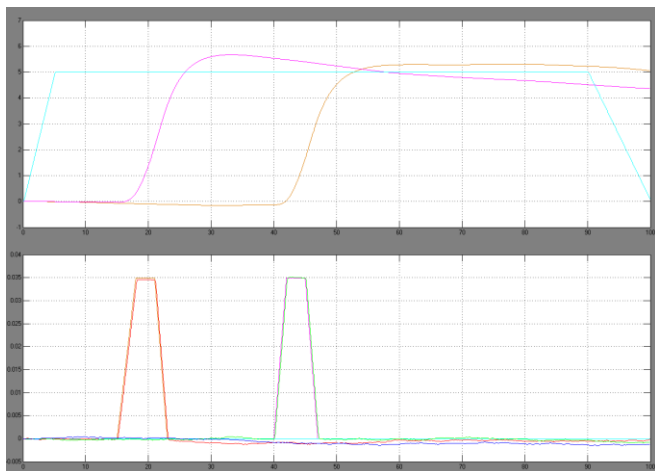
*Fig 6: Posicions lineals i angulars en el sistema continu discretitzat*

La diferència en les respostes de moviment en X i Y es deuen a que el temps de inclinació en cadascun dels eixos és diferent, això s'ha fet així per comprovar que ha diferents temps la trajectòria és l'esperada.

#### Sistema discret amb soroll

Les gràfiques del comportament obtingudes anteriorment tenen un comportament perfecte, per aquesta raó es va decidir afegir un soroll al sistema per simular les dades amb soroll pròpies dels sensors utilitzats.

Es pot veure a la gràfica que amb un soroll de l'ordre de 0.5 deg/s, les posicions angular no són constants en 0 quan no estan sent modificades pel controlador. Per aquesta raó es pot observar que les posicions lineals no es mantenen exactes un cop la consigna ha estat efectuada, i pateixen una deriva d'un metre en un minut.



*Fig 7: Posicions lineals i angulars en el sistema discret amb soroll*

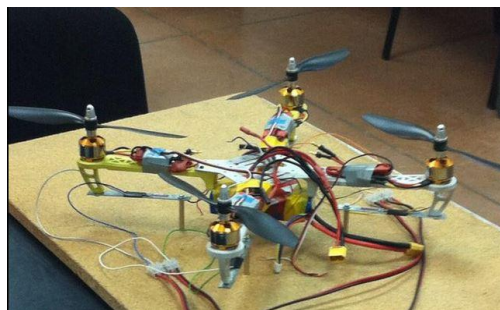
Per eliminar part d'aquest soroll s'ha utilitzat un filtre de Kalman, aquest és un observador òptim que filtre el soroll d'una variable del sistema, tot i que no s'ha portat a terme un estudi exhaustiu d'aquest.

## 4. Tests realitzats

En tot treball sempre és necessari passar de les simulacions a proves físiques, en aquest projecte s'han fet tres tipus de proves diferents relacionades amb els motors, les forces aplicades i el controlador.

### 4.1. Galgues extensiomètriques

La primera de les proves ha estat l'anclatge de l'OVI en un banc de proves construït a partir de quatre galgues extensiomètriques, posada una a cadascun dels braços per mesurar la força vertical dels motors. A més a més, com la subjecció no permetia l'elevació descontrolada de l'OVI es van poder mesurar les revolucions de les hèlix de cadascun dels motors a diferents consignes per treure la corba de comportament del motor.



*Fig 8: Banc de proves amb galgues.*

D'aquesta prova s'ha extret com a resultat que les forces de cadascun dels motors eren les adients a cadascuna de les consignes aplicades i això ha indicat que la velocitat de gir de les hèlix era la mateixa en tots quatre motors.

### 4.2. Junta de Cardan

La següent prova ha sigut l'anclatge de l'OVI a una junta de Cardan com es veu a la Fig 9. Una junta de Cardan és un element mecànic que permet la rotació en qualsevol dels tres eixos X, Y i Z. Aquesta prova ha servit per veure el comportament del sistema amb el controlador i sota unes consignes determinades, implicant el descontrol o el moviment no desitjat d'l'aparell. A més d'aquesta manera s'han evitat cops innecessaris i possibles trencaments de parts.



*Fig 9: Junta de Cardan*

### 4.3. Espai de vol

Unes de les proves realitzades durant tot el transcurs del projecte han estat efectuades a una zona habilitada amb xarxes, d'aquesta manera es podia portar a terme un vol sempre tenint en consideració la seguretat dels components del grup i la gent que estava mirant.



Fig 10: Espai de vol amb xarxes

Amb aquestes proves s'ha pogut extreure el punt d'equilibri en el qual el OVI es començava a elevar i en quin moment també la força en l'eix Z era la suficient com per mantenir-se en una posició determinada.

### 5. Prototip

Com a part del projecte s'ha redissenyat una estructura comercialitzada de manera que fos possible una construcció pròpia mitjançant la impressió 3D i perfils d'alumini i fusta. El primer pas ha sigut l'estudi de cadascuna de les parts que formen l'estructura:

- Braços
- Placa central
- Subjecció motors
- Unions

Fet aquest estudi es va arribar a la conclusió que era possible imprimir en 3D tant les subjeccions dels motors com les unions de les diferents peces de tot el conjunt. Es va escollir la impressió 3D ja que és una metodologia força nova però molt barata, i està en auge, a part de donar unes característiques mecàniques adequades a les necessitats.

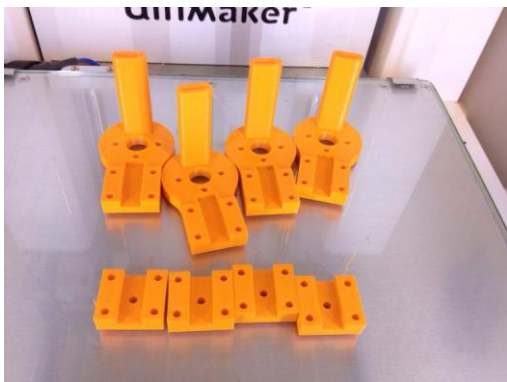


Fig 11: Subjeccions i unions impreses en 3D.

La part central constarà de dues plaques paral·leles de fusta per tal d'augmentar la rigidesa del muntatge. S'ha proposat un disseny quadrat però un dels avantatges de la fusta és que es pot adaptar a la forma desitjada fàcilment.

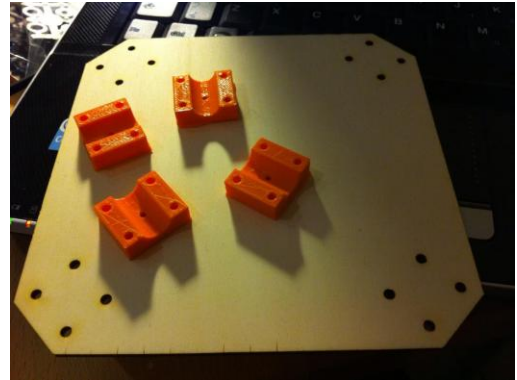


Fig 12: Subjeccions i placa central de fusta.

Els braços s'han tret a partir d'un perfil d'alumini tallat a la llargada desitjada. S'ha triat l'alumini ja que és resistent, fàcil de mecanitzar i aguanta força bé la torsió.



Fig 13: Prototip acabat.

## 6. Aplicacions

### 6.1. Controlador en arduino

El microcontrolador escollit per a governar el quadcòpter ha estat un Arduino Mini Pro. S'han estudiat diverses opcions per a ser cervell del quadcòpter: Raspberry PI, Arduino Mini Pro i Arduino Uno. S'ha triat Arduino Mini Pro ja que té suficient capacitat de processament per a dur a terme el control, les seves mesures són òptimes per integrar-les al quadcòpter i no es perd cap de les seves funcionalitats.

S'ha dissenyat un programa de control a partir dels valors i simulacions obtinguts via Matlab-Simulink que serà l'encarregat de dirigir el prototip. El programa es basa en el següent:

- Abans de res, fa Checklist de tot el necessari pel correcte funcionament del quadcòpter
  - Comprova totes les connexions
  - Configura els offsets de l'IMU
  - Configura els ESC's (Electronic Speed Controller)
  - Espera la connexió WiFi
  - Habilita les senyals lumíniques

- Passat el Checklist, comença una rutina que només s'interrompra en cas de parada d'emergència:

- Obtenció de les variables de sortida actuals mitjançant les lectures del giroscopi i l'acceleròmetre, i un filtre de Kalman. Les variables són: 3 posicions angulars, 3 velocitats angulars, altura i velocitat d'ascens o descens de l'OVI.

- Lectura de les senyals provinent de l'emissora Radio-control per saber els angles desitjats, així com l'altura i velocitat verticals

- Actualització de les velocitats de rotació de cadascun dels 4 motors mitjançant l'aplicació de la senyal de control als ESC's encarregats de governar-los

- Comunicació via i2c a l'Arduino encarregat de transmetre les dades via WiFi allà on faci falta (PC o mòbil) per a les lectures en temps real de tot el que està passant.

A continuació es mostra una gràfica del comportament en temps real d'una de les proves de vol mitjançant aquest tipus de control. Les de color magenta i verd són les dues consignes, i les de color rosa i blau són les respostes de l'OVI. S'ha fet un canvi de consigna d'angle primer respecte l'eix X (color magenta) per després tornar-la a 0 i es pot veure com l'OVI el segueix (color rosa). Posteriorment, s'ha fet el canvi a l'eix Y i es pot tornar a veure com l'OVI segueix aquesta consigna.

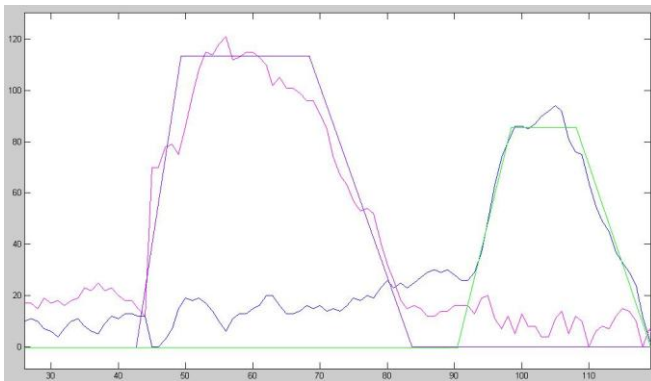


Fig 14: Posicions angulars del quadrotor en temps real

El programa complet, així com totes les llibreries utilitzades es poden trobar als annexos de la següent referència[4].

### 6.2. WiFi

Per poder establir una connexió entre dispositius s'ha configurat un mòdul Adafruit WiFi CC3000 (Fig 15) com a servidor connectant-lo a un Arduino per poder rebre i transmetre en temps real.



Fig 15: Mòdul WiFi CC3000

D'aquesta manera el sistema està preparat per rebre la trajectòria a seguir des de un comandament que també tingui un sistema WiFi.

Aquest sistema de comandament està principalment pensat per utilitzar tablets i smartphones amb sistema operatiu Android[5], un exemple d'aplicació és la desenvolupada en aquest treball:



Fig 16: Aplicació comandament en Android

Una aplicació composta de dos joystick, un per comandar direcció i l'altre pensat per portar l'alçada i el yaw.

A la vegada a més del control de la trajectòria la incorporació de un mòdul de comunicació WiFi permet portar a terme lectures de les diferents variables de la maqueta durant les proves del controlador en temps real.

### 6.3. Geolocalització

La geolocalització s'ha establert amb el mòdul Adafruit Ultimate GPS Breakout (Fig 17) està configurat per rebre la longitud i latitud de la seva ubicació, a més també mostra l'hora i la precisió de les seves dades.



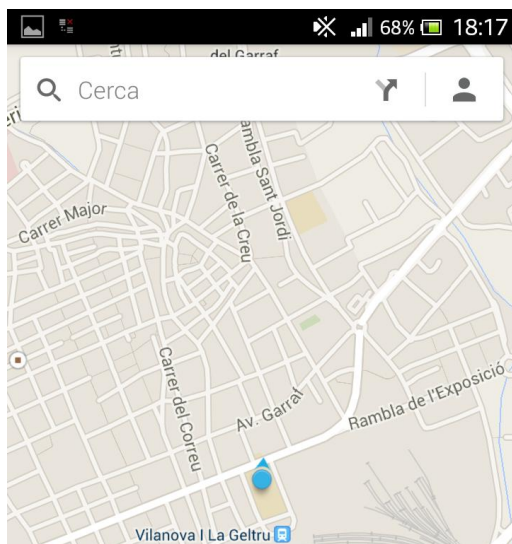
Fig 17: Mòdul Adafruit Ultimate GPS Breakout

Es poden rebre les dades de localització per port sèrie o connectant aquest mòdul al sistema de comunicació. Per aquesta raó també s'ha fet una aplicació mòbil que mostra la ubicació i la pot situar a Google maps. En la següent captura Fig 18 és pot veure la ubicació mostrada per l'aplicació:



*Fig 18: Aplicació comandament en Android*

Un cop pres el boto d'accedir a Maps ens mostra la ubicació en el google Maps:



*Fig 19: Aplicació comandament en Android*

#### 6.4. Visió

Com altres dades, aquest cop visuals, s'ha adquirit una càmera i s'ha estudiat el seu funcionament, per d'aquesta manera facilitar la captura d'imatges des del quadmotor.

La càmera utilitzada és una petita "Wireless CCTV Cam", fàcilment integrable a l'OVI pel seu petit tamany i poc pes. A més a més és fàcil d'utilitzar ja que compta amb un receptor de radiofreqüència que es connecta directament al PC i es pot obtenir les imatges de vídeo o fotos juntament amb so en temps real.



*Fig 20: Càmera amb receptor de RF*

#### 6.5 Radiocontrol

Un tema molt important, encara que complementari a la aplicació Android programada, és el de poder fer proves sense cap risc, per aquest tema es va arribar a la conclusió que la forma més segura, amb un estàndard molt aprovat en el món de l'aeromodelisme, és el radiocontrol. Ja que mentre es portava a terme el tema de la programació de l'aplicació i la seva comunicació es podien anar fent proves amb aquest sistema.



*Fig 21: Imatge del emissor i receptor del radiocontrol*

En el prototip s'ha instal·lat una receptora de radiofreqüència per fer proves de vol amb control sobre potencia i parada d'emergència de forma segura i robusta, d'aquesta manera es té un control molt acurat del sistema físic. Podent aturar els seus motors en cas de descontrol, dirigir la direcció del OVI i fins i tot la potencia dels motors per definir una alçada desitjada.

#### 7. Conclusions

Després dels quatre mesos en els que s'ha elaborat aquest projecte, es poden extreure una sèrie de conclusions sobre la elaboració del treball i tota la feina realitzada.

Una de les conclusions i possiblement la primera a la que es va arribar un cop obtingudes les gràfiques de comportament, comentades anteriorment, va ser que l'estudi del model de la planta s'ha efectuat correctament. A més, també es pot concloure, que efectuar un control de la trajectòria aplicat sobre les posicions angulars funciona adequadament, permetent la programació d'una ruta en les simulacions efectuades.

Un altre aspecte final és el de la implementació del controlador en un microprocessador, en aquest cas es va escollir l'Arduino Mini Pro. Aquesta elecció ha estat l'adequada per temes de pes espai i el més important de capacitat de processament. Un altre aspecte a tenir en consideració en aquest camp és la programació del dispositiu que s'ha portat a terme adequadament, ja que en les proves efectuades el microprocessador ha respost adequadament mirant les proves en llaç obert del controlador obtingut mitjançant la realimentació d'estats.

Aconseguir l'obtenció de dades del controlador en temps real, mitjançant el mòdul WiFi, ha aportat una informació molt útil del comportament de l'aparell. A partir d'aquestes dades s'ha pogut anar regulant i ajustant molts dels paràmetres que corresponen al control i a la modelització.

A més a més s'ha après a programar aplicacions Android en codi Java. Aquesta tasca que va portar molts problemes en un inici, s'ha convertit en un mètode fàcil de interactuar i de crear funcions útils en el projecte.

Per acabar comentar que la introducció del sistema de visió i geolocalització ha estat una bona idea per completar la percepció del entorn, i completar una possible aplicació futura d'aquest tipus d'aparells.

Per últim, cal dir que s'han trobat moltes més dificultats de les que en un principi eren esperades, degut a que el nivell de teoria necessari per la seva realització ha estat bastant més alt que la rebuda durant el grau.

## 8. Agraïments

Si ha estat possible la realització del projecte és sobretot gràcies al nostre tutor Pau Martí i Antonio Camacho en tema de modelat i control, ja que amb les seves explicacions s'han pogut sobrepassar alguns dels entrebancs trobats.

També agrair a Cristobal Raya per fer arribar la seva teoria i coneixements en tema de programació d'Android i altres consells rebuts per la seva part.

A German Morillo, Manuel Liebanas, Juanjo Vázquez i Balduí Blanqué per les seves recomanacions, estructures muntades per portar a terme proves i coneixements compartits.

A Cecilio Angulo i a Wilber Aguilar pels seus consells i bons moments passat conjuntament. Des de Barcelona, a Manuel Velasco i Gabri, que estan desenvolupant també un controlador per un quadcòpter, amb els quals s'ha compartit coneixement i resultats

Agrair també a Victor Cano, de "Un ojo en el cielo" [6] per les seves estones amb nosaltres compartint consells de temes de vol i visió de quadcòpters i per deixar-nos material de gran utilitat del projecte

A BeachLAB Sitges[8] per deixar-nos la maquinària necessària per a portar a terme els prototips i pels seus consells referents a impressió 3D.

A tot el grup Agile de la EPSEVG [7] per escoltar els nostres progressos i portar un seguiment del treball en tota la seva duració.

I per últim a tots aquells que s'han interessat en el nostre projecte i algun cop s'han preguntat:

"Ja vola?"

## Referències

- [1] Teppo Luukkonen, "Modelling and control of quadcopter", *Independent research project in applied mathematics*, Aalto University School of Science, Espoo, August 22, 2011
- [2] Wikipedia, "Coriolis effect" [*en linea*]. Actualizat Agost 6, 2014. Disponible en: [http://en.wikipedia.org/wiki/Coriolis\\_effect](http://en.wikipedia.org/wiki/Coriolis_effect)
- [3] Pau Martí "Sistema lineals en temps discret" *Enginyeria de control*, Escola politècnica superior de Vilanova i la Geltrú, Setembre 13, 2013
- [4] Repositori Github "OviCopter"  
"https://github.com/agjunyent/OVICopter"
- [5] Cristobal Raya "Android" Programació de dispositius mòbils, Escola politècnica superior de Vilanova i la Geltrú, Octubre 17, 2011
- [6] Equip Agile EPSEVG:  
"blocs.epsevg.upc.edu/agileupcvilanova"
- [7] Un Ojo en el cielo:  
<http://unojoenelcielo.es/>
- [8] BeachLab Sitges:  
<http://fablabsitges.org/>