



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO INTEGRAL DE UN QUADROTOR: DISEÑO DE UN ALGORITMO DE CONTROL PARA LA POSICIÓN X-Y BASADO EN SEÑALES GPS.

AUTOR: Castillo Frasquet, Alberto

TUTOR: García Gil, Pedro

Curso Académico: 2013-14

A mis compañeros de laboratorio por todas las buenas horas que hemos pasado allí.

A Pedro García Gil y a Luis Ródenas, por toda su atención dedicada.

Este libro contiene, por orden, los siguientes documentos:

MEMORIA
PRESUPUESTO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO INTEGRAL DE UN
QUADROTOR: DISEÑO DE UN ALGORITMO
DE CONTROL PARA LA POSICIÓN X-Y
BASADO EN SEÑALES GPS.**

MEMORIA

AUTOR: Castillo Frasquet, Alberto

TUTOR: García Gil, Pedro

Curso Académico: 2013-14

Memoria

Índice

1. Introducción.	7
1.1. Motivaciones y estado del arte.	9
1.2. Objetivos.	13
1.3. Resultados obtenidos.	14
1.4. Estructuración del documento.	15
2. Modelo teórico de un quadrotor.	17
2.1. Variables generales que caracterizan a un quadrotor.	17
2.2. Marcos de referencia.	18
2.3. Acciones aplicadas al quadrotor.	21
2.4. Modelo cinemático general.	22
2.5. Modelo dinámico general.	23
2.6. Modelo simplificado del quadrotor.	26
3. Descripción de la plataforma.	29
3.1. El grupo de motores.	31
3.1.1. Estructura de soporte y tren de aterrizaje.	31
3.1.2. Motores	32
3.1.3. Variadores	33
3.1.4. Hélices.	34
3.1.5. Batería y su soporte en la estructura.	35
3.1.6. Conexión eléctrico.	37
3.1.7. La protección.	38
3.2. El módulo de estabilización.	40
3.2.1. La IMU.	40
3.2.2. El sensor de altura.	41
3.2.3. El microprocesador programable.	42
3.3. La placa para vuelos en exteriores.	44
3.4. El mini-PC u ordenador de abordo.	48
3.5. Resultados finales.	51
4. Localización por GPS.	53
4.1. Posicionamiento basado en trilateración.	54
4.2. Determinación de la posición de los satélites y distancia que nos separa de ellos.	56
4.3. Fuentes de errores.	57
4.4. Sistemas DGPS.	59
4.5. Sensores GPS estándar. Ultimate GPS breakout v3.	60

5. Descripción y resultados experimentales del sistema GNSS/INS.	67
5.1. Ideas fundamentales sobre observadores y filtro de Kalman.	69
5.2. Conversión de las coordenadas latitud y longitud a medidas de posición locales (x,y).	70
5.3. Obtención de las proyecciones de la velocidad en cada eje v_x, v_y a partir del módulo V dado por el GPS.	71
5.4. Rotación de las medidas de la IMU desde el sistema de referencia del quadrotor al de tierra.	72
5.5. Necesidad de estimar el <i>offset</i> del acelerómetro.	74
5.6. Retraso en las medidas del GPS. Necesidad de aplicarlo a la IMU.	75
5.7. Aplicación del algoritmo de Kalman.	76
5.8. Comprobaciones realizadas.	79
5.9. Resultados experimentales obtenidos.	81
5.10. Mejoras y trabajos futuros.	83
6. Desarrollo y resultados de la ley de control x-y.	85
6.1. Modelo de control de posición.	86
6.2. Modelo de control simplificado.	86
6.3. Ley de control.	87
6.4. Ley de control con control de velocidad.	88
6.5. Generalización de la ley de control para los dos ejes.	90
6.6. Generalización a tres ejes. ¿Se puede considerar el control de altura independiente?.	92
6.7. Rotación de coordenadas para hacer el control independiente del yaw.	92
6.7.1. ¿Porqué únicamente funciona el control para $\psi=0$?.	93
6.7.2. Solución al problema.	93
6.8. Gestión de WayPoints.	95
6.9. Algoritmo de control x-y final.	96
6.10. Resultados y simulaciones.	98
6.10.1. Control de posición de un desplazamiento recto.	98
6.10.2. Control de posición de un desplazamiento recto cuando $\psi \neq 0$	99
6.10.3. Control de posición de un desplazamiento en diagonal.	102
6.10.4. Rechazo de Perturbaciones. Cambio brusco de velocidad durante el desplazamiento.	105
6.10.5. Rechazo de Perturbaciones. Viento.	105
6.10.6. Ejemplo de control de trayectorias basadas en el algoritmo de seguimiento de WayPoints. Avance recto ejecutando diagonales.	107
6.10.7. Capacidad para aproximar trayectorias más complejas. Seno + semicírculo.	107
6.11. Resultados experimentales. Simulaciones en bucle abierto.	110
6.11.1. Ensayo bucle abierto 1: Con el quadrotor en parado.	110
6.11.2. Ensayo bucle abierto 2: Oscilando en torno a la posición de referencia y cambiando el yaw.	112

6.12. Conclusiones y trabajos futuros. 112

Bibliografía

Referencias, 115

Anexos

Artículo para las jornadas de automática, 116

Descripción del modelo empleado en el simulador MatLab, 125

Índice de figuras

1.1. RPAS comercial de tipo Quadrotor empleado para la grabación de vídeo aéreo	10
1.2. Century NEO 940C Hexa-Rotor	10
1.3. Multicopter de 8 rotores.	11
1.4. EVE UAV. Hexacopter	11
1.5. KK2.0 Multi-rotor LCD Flight Control Board	12
2.1. Definición del marco de referencia del quadrotor o Body Frame \mathcal{O}^a	19
2.2. Definición del ángulo de rotación ψ	19
2.3. Definición del ángulo de rotación θ	20
2.4. Definición del ángulo de rotación ϕ	20
2.5. Marco de referencia de tierra al cual se le aplica una rotación en ψ	21
2.6. Acciones sobre el quadrotor debidas a los motores.	22
2.7. Modelo completo de ecuaciones del quadrotor.	25
2.8. Modelo simplificado de ecuaciones del quadrotor	27
3.1. Esquema conceptual de la plataforma desarrollada	30
3.2. Estructura de soporte empleada	31
3.3. Imagen del motor Robbe Roxxy 2827-35	32
3.4. Generación de una tensión media constante mediante PWM	33
3.5. Variador YGE 30i	34
3.6. Hélices EPP1045 empleadas en el quadrotor	34
3.7. Batería LiPo 3S1P de la marca Gens Ace	35
3.8. Diagrama de descarga de una celda LiPo	36
3.9. Soporte para la batería LiPo.	37
3.10. Esquema eléctrico de alimentación del quadrotor	38
3.11. Aspecto de la estructura tras la primera fase de montaje	39
3.12. Aspecto final de la protección diseñada para el quadrotor	39
3.13. IMU 10DOF v2 de Drotek	41
3.14. Ultrasonic Distance Sensor de Parallax.	41
3.15. Arduino Due	43
3.16. Arduino Micro	45
3.17. Conversor de tensión (izquierda) y conversor de niveles lógicos (derecha)	46
3.18. Ultimate GPS breakout v3	46
3.19. Radio Xbee (derecha) y Receptor RC (izquierda)	46
3.20. Esquema de conexiones de la placa de exteriores	47
3.21. Foto de la placa para vuelos en exteriores una vez se hubieron instalado todos los elementos en una placa de prototipado.	48
3.22. Placa Igep v2	49
3.23. Quadrotor en vuelo.	51
3.24. Resultados experimentales. Seguimiento de referencias. Pitch y roll.	52
3.25. Resultados experimentales. Acciones enviadas a los motores.	52
4.1. Obtención de la posición de un punto en el plano conociendo las distancias que lo separan de otros 3.	55

4.2.	Fenómeno multipath.	58
4.3.	Cara delantera y trasera del <i>Ultimate GPS breakout v3</i>	61
4.4.	Captura Pantalla de <i>Google Maps</i> donde se ha introducido la información NMEA dada por el GPS	63
4.5.	Información de posicionamiento en formato NMEA dada por el <i>Ultimate GPS</i>	63
4.6.	Número de bytes leídos cada vez que se vacía el <i>buffer</i> del Arduino Micro.	64
5.1.	Spatial FOG	68
5.2.	Diagrama de bloques de un observador.	69
5.3.	Definición de la Latitud y Longitud.	71
5.4.	Offset del acelerómetro en el eje x.	74
5.5.	Resultados experimentales. Retraso en la señal GPS.	75
5.6.	Descomposición de una aceleración en dirección norte cuando el quadrotor se encuentra inclinado en pitch.	79
5.7.	Resultados experimentales. Rotación de las aceleraciones dadas por la IMU para expresarlas en el sistema de referencia de tierra. Variación en roll y pitch.	80
5.8.	Descomposición de una aceleración en dirección norte cuando el quadrotor se encuentra girado en yaw.	80
5.9.	Resultados experimentales. Rotación de las aceleraciones dadas por la IMU para expresarlas en el sistema de referencia de tierra. Variación en yaw.	81
5.10.	Resultados experimentales. Medidas x,dx,y,dy dadas directamente por GPS.	82
5.11.	Resultados experimentales. Medida de posición tras aplicar el EKF.	83
5.12.	Resultados experimentales. Corrección del offset.	83
6.1.	Ley de control de posición sobre el eje x.	88
6.2.	Ley de control de posición con control de velocidad sobre el eje x.	89
6.3.	Rotación del sistema de referencia de tierra.	94
6.4.	Definición de la zona de llegada al WayPoint.	95
6.5.	Resultados Simulación. Desplazamiento 5m en el eje x sin control de velocidad.	98
6.6.	Resultados Simulación. Desplazamiento 20m en el eje x con control de velocidad donde se ha fijado en 2m/s.	99
6.7.	Resultados Simulación. Desplazamiento 20m en el eje x con $\psi \neq 0$	100
6.8.	Resultados Simulación. Desplazamiento 20m en el eje x con $\psi = f(t)$	101
6.9.	Resultados Simulación. Desplazamiento diagonal donde no se han aplicado las ecuaciones que definen las saturaciones.	103
6.10.	Resultados Simulación. Desplazamiento diagonal donde sí que se han aplicado las ecuaciones que definen las saturaciones.	104
6.11.	Resultados Simulación. Rechazo de un cambio muy brusco de velocidad.	106
6.12.	Resultados Simulación. Rechazo del viento.	107

6.13. Resultados Simulación. Trayectoria, el quadrotor avanza siguiendo diagonales.	108
6.14. Resultados Simulación. Trayectoria compleja, el quadrotor sigue una trayectoria aproximada de un seno más un semicírculo.	109
6.15. Resultados experimentales. Simulación ley de control x-y en bucle abierto.	111
6.16. Resultados experimentales. Simulación de la ley x-y en bucle abierto. Cambios en yaw.	111
6.17. Los momentos de inercia del quadrotor se calculan asumiendo la siguiente geometría. Tiene en cuenta la masa y forma de los motores, la masa y longitud de las varillas de sujeción y la masa de toda la electrónica y batería (que se se asume como una esfera situada en el centro).	127

1. Introducción.

Este documento conforma la memoria de un trabajo de final de grado de la titulación de Grado de Ingeniería en Tecnologías Industriales, cursado en la Universidad Politécnica de Valencia durante los años 2010-2014.

El presente proyecto consiste en el desarrollo integral de una plataforma para el control completo de un sistema aéreo remotamente pilotado (*Remotely Piloted Aircraft Systems*, RPAS) de tipo quadrotor. Se enfoca en torno a la perspectiva de realizar vuelos en exteriores. Hay que mencionar que se ha partido de estudios ya realizados previamente por *Luis Ródenas L.*, *Ricardo Sanz D.*, y en general por todos los alumnos que han colaborado con el *Dr. Pedro García Gil* a lo largo de los últimos años. Existe un objetivo de continua mejora y de prueba de diversas configuraciones que vayan mejorando constantemente los resultados y capacidades del quadrotor.

Uno de los aspectos críticos que faltaba por tratar, era dotar al sistema con un control de la posición horizontal, de tal forma que no solo fuera capaz de sustentarse quieto en unas coordenadas concretas, sino que además fuera capaz de ejecutar trayectorias pre-programadas de forma autónoma. Para ello el aspecto más crítico, y la mayor limitación, es tener una medida suficientemente fiable de la posición horizontal (x-y). El presente proyecto se enfoca en torno a éstos dos objetivos, en primer lugar obtener en una buena medida de la posición, y en segundo lugar en el desarrollo de un algoritmo de control de posición que sea capaz de controlar el quadrotor de manera independiente al resto de variables ya que, como se verá a lo largo de este documento, existe una fuertísima relación de dependencia dinámica entre las variables que rigen al sistema.

La parte inicial del proyecto consistió en construir un quadrotor totalmente nuevo desde cero. Una vez se logró que sustentase de manera estable en vuelo «hover» se paso a la segunda parte. El estudiante *Pablo Albiol* se ha encargado de implementar el enlace de telemetría a través de un módulo de radio y un ordenador en tierra. El estudiante *Daniel Verdú* se ha encargado de llevar a cabo un control de altura mediante un sensor barométrico y de un control de orientación mediante un sensor magnetómetro. En este proyecto, como se ha comentado, se desarrolla una ley de control de posición x-y basada en señales GPS.

El trabajo ha sido realizado durante los meses de enero a julio del año 2014. El total del tiempo ha sido invertido tanto en tareas prácticas (montaje de la plataforma, diseño de la placa de baquelita, conexionado, soldaduras...) como en labores teóricas, como el estudio de la plataforma previamente desarrollada, estudio de algoritmos y códigos de programación, familiarización con nuevos entornos (como el entorno de programación de Arduino y Linux). Pero también ha sido necesario el estudio y aprendizaje de nuevos conceptos en diversos campos como estrategias de control, filtros avanzados, comunicación entre dispositivos digitales,

señales, etc; para los cuales ha sido fundamental la base adquirida en los estudios de grado.

Por último, destacar que el objetivo que nos planteó el tutor no fue el desarrollo de una plataforma comercial. Sino desarrollar una plataforma completamente modulada que nos permitiera adquirir la máxima cantidad de conocimientos multidisciplinarios. No se ha buscado la optimización de recursos, sino el empleo de éstos, para un fin didáctico y poder adaptarlos a un sistema real.

1.1. Motivaciones y estado del arte.

Las siglas UAS (*Unmanned Aircraft System*) sería el término más general para clasificar a un vehículo aéreo de tipo quadricóptero, o también llamado quadrotor. No obstante, teniendo en cuenta las últimas definiciones que se han dado en Europa al respecto, un quadrotor es un tipo particular de RPAS (*Remotely Piloted Aircraft Systems*), y un RPAS es un subconjunto de los UAS, los cuales se caracterizan porque constan de dos partes:

- RPA (*Remotely Piloted Aircraft*): Sistema automático que es capaz de ejercer un vuelo semiautónomo y es capaz de responder a las demandas del piloto, el cual no se encuentra a bordo.
- RPS (*Remotely Pilot Station*): Estación de tierra con la cual se tiene monitorizado el estado del vehículo en todo momento.

Los RPAS no sustituyen a los tradicionales vuelos tripulados en los que el piloto se encuentra a bordo, pero hacen posibles muchas labores que de la otra forma serían imposibles. Al encontrarse el piloto en tierra se puede construir una plataforma mucho más pequeña y ágil, apta para desplazarse por entornos urbanos o por lugares de difícil acceso. También se reducen mucho los costes y, además, pueden llevarse a cabo tareas mucho más arriesgadas sin que la vida del piloto corra ningún peligro.

Inicialmente los primeros RPAS se construyeron para fines militares debido al alto coste de sus componentes, pero en los últimos años los avances en electrónica han abaratado los precios de los componentes y se ha disparado enormemente su demanda para aplicaciones urbanas y comerciales. Los últimos diseños incorporan potentes ordenadores a bordo y algoritmos de control que son capaces de estabilizar automáticamente vehículos para los que se necesitarían unos cuantos años de práctica, y cierto talento innato, si se quisiera pilotarlos con destreza de forma manual.

La figura 1.1 muestra un prototipo comercial de Quadrotor, al cual se le ha incorporado una cámara de vídeo como complemento para tomar vídeos desde una perspectiva aérea. Esta imagen es una muestra de una aplicación en concreto, pero el potencial es altísimo ya que entre otras aplicaciones los RPAS son idóneos para tareas como: inspección de campos agrícolas, inspección de infraestructuras (tanto en interior como en exterior), supervisión de generadores aéreos, inspección de tuberías y oleoductos, investigación atmosférica, industria cinematográfica, fotografía deportiva, investigación de ecosistemas salvajes o asistencia en desastres naturales.

La cuota de mercado es altísima y por ello se ha disparado su uso actualmente. Es por esto que la Comisión Europea ha abierto recientemente un proceso legislativo para la regularización e incorporación de éstos vehículos al espacio aéreo



Figura 1.1: RPAS comercial de tipo Quadrotor empleado para la grabación de vídeo aéreo



Figura 1.2: Century NEO 940C Hexa-Rotor

Europeo [1, 2]. El fin es regular el uso masivo que están teniendo actualmente, de tal forma que no se ponga en peligro la seguridad del ciudadano, estableciendo unos requisitos de seguridad mínimos que el RPAS en cuestión debe cumplir.

A pesar de que el quadrotor sea el más extendido en investigación por ser el más sencillo, existen muchas variantes. Generalmente la diferencia fundamental son el número y distribución de los motores. La figura 1.2 muestra una variante de 6 rotores con una distribución circular, también llamados *hexarotors*. Aunque también se pueden encontrar variantes de 8 motores, como el de la figura 1.3, o incluso más (éstos últimos son apodados bajo el término *multicopters* o *multirotors*). La distribución de los motores tampoco tiene porqué ser circular, por ejemplo, la figura 1.4 muestra un hexarotor comercial llamado *EVE UAV* el cual no tiene los motores distribuidos circularmente, sino que los tiene en *H*.



Figura 1.3: Multicopter de 8 rotores.



Figura 1.4: EVE UAV. Hexacopter

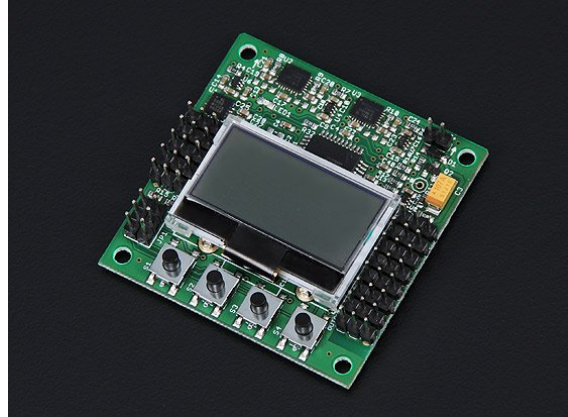


Figura 1.5: KK2.0 Multi-rotor LCD Flight Control Board

Por otro lado, también existen muchas variantes en torno a cómo se controlan estos vehículos, hay que destacar que la mayor parte de ellos cuentan con un sistema de control asistido. Esto quiere decir que disponen de un algoritmo de control que se encarga de garantizar la estabilidad del helicóptero, mientras que el control de posición lo hace el piloto desde tierra a través de una emisora de radiofrecuencia. Por ejemplo, la figura 1.5 muestra la placa electrónica KK2.0, ésta placa tiene un precio de unos 30\$ y viene ya programada con los algoritmos de control y con los sensores necesarios para garantizar la estabilidad de 20 tipos diferentes de multicopters. También hay quadrotors que cuentan con sistemas de navegación que permite hacer un control de la posición de manera automática, aunque siempre tienen la opción de cambiar a control asistido.

Para poder hacer un control de estabilidad se debe contar con los sensores necesarios. Generalmente se emplean unidades de medida inercial (IMU's) con las que se determina la orientación del quadrotor en el espacio. Dichas medidas de orientación se mandan al algoritmo de control que se encarga de procesarlas y mandar las señales de control necesarias a cada motor. Muchos de ellos emplean otros tipos de sensores también, por ejemplo, se suelen emplear sensores de ultrasonidos o sensores barométricos para obtener medidas de altura y así poder hacer también control sobre la altura, o magnetómetros y sensores GPS para monitorizar y controlar la posición horizontal.

La mayor parte de ellos disponen también de un ordenador de a bordo, o de un procesador, que se encarga de procesar la señal de los sensores y de ejecutar la ley de control. Funcionan a muy alta frecuencia, ya que son capaces de actuar sobre los motores a frecuencias de 200-500Hz, o incluso más, por lo que son capaces de actuar mucho más rápido ante una perturbación que lo que lo harían los reflejos de cualquier persona.

En cuanto a estrategias de control, lo más empleado por su sencillez y robustez son controladores PID (o derivados) para la estabilidad [5]. Para el control de posición se suele emplear control H_∞ debido a su capacidad de seguimiento de trayectorias [6]. No cabe olvidar que existen muchos otros tipos de controladores mucho más complejos que también son empleados en quadrotors, como técnicas de control predictivo, técnicas de control óptimo, algoritmos de aprendizaje, etc; pero que se escapan del alcance de éste proyecto.

1.2. Objetivos.

Los objetivos que se establecieron al inicio fueron cuatro. En primer lugar fue construir un quadrotor nuevo, desde cero. Este objetivo se daría por cumplido una vez se viera que la plataforma es capaz de mantener la estabilidad y rechazar perturbaciones durante el vuelo.

El segundo objetivo será dotar a la plataforma de un sensor GPS que permita obtener medidas de posición (x-y) en exteriores. Conforme se fue ensayando el sensor GPS se vio que sería prácticamente imposible hacer ningún control con solo un sensor GPS ya que la posición daba demasiado error, y la señal presentaba un aspecto muy escalonado (como es normal en los GPS corrientes). Por ello, para mejorar la precisión fue necesario programar un filtro de Kalman extendido (*Extended Kalman Filter*, EKF) que combinase las medidas del GPS con las de la IMU. Esta técnica suele llamarse sistema GNSS/INS.

El tercer objetivo se fijó en desarrollar una estrategia de control que fuera capaz de controlar la posición del quadrotor de una forma lo más independientemente posible del resto de variables, y que le permitiese seguir trayectorias pre-programadas basadas en WayPoints. Para poder evaluar la respuesta se tuvo que diseñar un simulador en MatLab que incorporase el modelo del quadrotor y donde se pudiera programar la ley de control y ver la respuesta antes de implementarla en el propio quadrotor.

El cuarto y último objetivo sería lograr que el quadrotor hiciera un vuelo autónomo al aire libre en el que ejecutara una trayectoria pre-programada una vez estuviera todo lo anterior implementado.

1.3. Resultados obtenidos.

El primer objetivo se logró en los primeros meses de trabajo en los que aparecieron innumerables problemas, casi todos ellos derivados de los sistemas electrónicos, como problemas de comunicaciones, lectura de sensores, etc. Sin embargo, se solucionaron todos ellos y consiguió que la aeronave ejecutase vuelos tanto en interiores como en exteriores con un control en actitud casi excelente.

El tercer objetivo, al ser la parte más creativa del proyecto, se fue desarrollando de manera paralela al resto. Se partió de una ley de control de posición muy básica la cual se fue modificando y amoldando a nuestras demandas. Es quizás, para mí, la parte más satisfactoria de este trabajo, puesto que conforme se fue comprendiendo el problema fueron surgiendo soluciones que, todas ellas en conjunto, constituyen una ley que permite controlar de manera casi total un sistema muy inestable y con una fuertísima relación de entre sus variables.

El segundo objetivo se asumió como cumplido cuando tras aplicar el EKF y ajustarlo todo, se vio que no se podrían conseguir mejores resultados con el tiempo y material disponible. Los resultados de posición finales son bastante buenos ofreciendo precisiones de $\pm 1,5 m$ en posición y de $\pm 0,3 m/s$ en velocidad. Como se comentará más adelante para conseguir alta precisión con sistemas basados en GPS (0.1m para posición y 0.02m para velocidad) habría que prescindir de sensores comerciales que se limitan a dar las medidas de latitud y longitud para pasar a utilizar sensores GPS que ofrecen las señales directas que mandan los satélites.

El cuarto objetivo no ha llegado a lograrse en bucle cerrado, pero si que se vio un comportamiento satisfactorio en bucle abierto. La razón por la que no se cerró el bucle fue fundamentalmente por motivos de viento y cobertura GPS. El problema fundamental ha sido que los motores que había disponibles en el laboratorio, y que instalamos en nuestra plataforma no eran suficientemente potentes como para rechazar el viento. Por motivos de seguridad, se decidió no cerrar el bucle hasta solucionar estos problemas de hardware.

Como consecuencia de este trabajo se redactó un artículo para las *Jornadas de Automática* con la descripción de la plataforma, con la ley de control desarrollada y algunas gráficas con los resultados obtenidos con el GPS.

1.4. Estructuración del documento.

Se presentan aquí los puntos que conforman el proyecto, donde se ha resumido brevemente los aspectos que se van a tratar en cada uno de ellos.

1. **Introducción.**

Se detalla el objeto del proyecto y motivaciones. Se describen cuales son los objetivos planteados inicialmente y logrados finalmente, y se resume cual es el estado de arte en relación a los sistemas RPAS.

2. **Modelo teórico de un quadrotor.**

En este apartado se describe el modelo de comportamiento general de un quadrotor. Se definen las variables generales del quadricóptero, que necesitaremos conocer para poder desarrollar cualquier tipo de control, y los sistemas de referencia sobre los que se van a definir medidas durante todo el documento. También se deduce, a partir de las leyes de Newton, cuál es el modelo de comportamiento del quadrotor expresado en ecuaciones diferenciales, y se asumen ciertas hipótesis que nos permitirán reducir el modelo a uno mas sencillo que sea apto para poder simular y diseñar controladores.

3. **Descripción de la plataforma.**

Se detalla y justifica cuál ha sido la combinación de hardware y software que forma la plataforma final. Se divide en 4 módulos o grupos, cada uno con su función y que en conjunto hacen que funcione el sistema general. En este punto se detalla cada uno de éstos módulos

4. **Localización por GPS.**

Es un punto de carácter informativo, se describen los principios del posicionamiento por GPS así como los errores de medida y porqué son debidos. Describe también como corregir éstos errores y se nombran otros sistemas que se podrían aplicar para medir la posición (x,y) de un quadrotor.

5. **Descripción y resultados experimentales del sistema GNSS/INS.**

Describe el EKF que se ha diseñado para tratar la señal GPS. Se comenta la parte teórica en relación al filtro de Kalman y cómo se aplica a nuestro caso. Se describe el modelo empleado y todo lo que ha sido necesario tener en cuenta para el correcto desempeño. Se muestran, también, los resultados experimentales obtenidos y la comparativa de la señal sin aplicar el EKF y aplicándolo.

6. **Desarrollo y resultados de la ley de control x-y.**

Describe de manera ordenada como se fue desarrollando la ley de control para la posición x-y, los resultados obtenidos en simulación y los resultados experimentales que se obtuvieron en bucle abierto.

7. **Anexos.**

Se incluye el artículo que se redactó para las Jornadas de Automática. También se incluye información sobre el simulador MatLab con el que se ha simulado la ley de control.

2. Modelo teórico de un quadrotor.

En esta parte del documento se desarrolla el modelo de comportamiento teórico de un quadrotor. Dicho modelo ya es conocido y se puede encontrar amplia bibliografía al respecto, concretamente, en [5, Puntos 1, 2, 3 y 4] se ofrece una explicación muy detallada y con una distribución similar a la que se va a seguir en esta sección.

La obtención del modelo del quadrotor puede derivarse de dos maneras diferentes que, evidentemente, conducen al mismo resultado. Por un lado puede obtenerse a partir de un enfoque lagrangiano de la física, en el cual, las leyes del movimiento surgen tras aplicar las ecuaciones de Euler-Lagrange¹ al propio sistema y, por otro lado, puede derivarse también con el enfoque newtoniano tradicional. En este apartado se ha seguido este segundo enfoque por las siguientes razones:

- Da una idea mucho más clara sobre cuales son las acciones (fuerzas y momentos) que sufre el quadrotor, y como éstas le afectan al movimiento.
- Muchas de las matrices de rotación que se aplican se usarán posteriormente para el tratamiento de datos de sensores.

La derivación a través de la aplicación de la ecuación de Euler-Lagrange puede encontrarse en [6, Sección 2.5] o en [7, Sección 3].

Este punto está estructurado de la siguiente manera: En el punto 2.1 se detallan cuáles van a ser todas las variables que van a caracterizar al quadrotor, en el punto 2.2 se definen qué marcos de referencia se van a emplear, en el punto 2.3 se describen las fuerzas y momentos a los que está sometido el quadrotor y cuál es su causa, en los puntos 2.4 y 2.5 se desarrollan los modelos cinemático y dinámico generales que definen completamente el movimiento del quadrotor, finalmente en 2.6 se asumen ciertas hipótesis que simplifican el modelo a uno más sencillo y más apto para aplicar estrategias de control.

2.1. Variables generales que caracterizan a un quadrotor.

El quadrotor se puede entender como un sólido rígido libre en el espacio, por lo que tendrá 6 grados de libertad: $(x, y, z, \theta, \phi, \psi) \in \mathbb{R}^6$, siendo (x, y, z) la posición de su centro de masas (c.d.m. en adelante), y (θ, ϕ, ψ) los ángulos de Euler que definen su orientación.

¹Las ecuaciones de Euler-Lagrange en su forma más general se dan para un sistema en coordenadas generalizadas (q_1, q_2, \dots, q_n) y toman la siguiente forma:

$$\frac{d}{dt} \left(\frac{\partial L(q_i, \dot{q}_i)}{\partial \dot{q}_i} \right) - \frac{\partial L(q_i, \dot{q}_i)}{\partial q_i} = 0$$

Por lo tanto, para definir completamente la posición del quadrotor en un instante será necesario conocer esas 6 variables. A los ángulos de Euler se les denota como roll, pitch y yaw respectivamente. Es importante ver como se definen estos ángulos, pero para ello antes es necesario conocer los sistemas de referencia sobre los cuales vamos a referir la posición del helicóptero.

2.2. Marcos de referencia.

Vamos a definir un total de tres marcos de referencia que van a ser necesarios tanto para la definición del modelo del quadrotor, como para la estrategia de control x-y y el tratamiento de las señales de algunos sensores que comentaremos más tarde.

MARCO DE REFERENCIA INERCIAL O DE TIERRA \mathcal{O}^i .

Con \vec{i} apuntando al norte geográfico, \vec{j} apuntando al este y \vec{k} apuntando al centro de la tierra. Su origen se establece en unas coordenadas fijas, generalmente usaremos aquellas en las que esté ubicada la estación de control de tierra (RPS, *Remotely Pilot Station*).

La posición (x, y, z) del helicóptero queda definida en este marco. Así pues, x será la distancia en dirección norte a la cual se encuentra el c.d.m. del quadrotor, y será la distancia en dirección este y z será la altura. Generalmente z adquirirá un valor negativo ya que el eje \vec{k} apunta al centro de la tierra, por ello, para que sea más intuitivo, al referirnos a la altura hablaremos de $h = -z$.

MARCO DE REFERENCIA DEL QUADROTOR O BODY FRAME \mathcal{O}^q .

Este marco está fijo en el c.d.m. del quadrotor y se desplaza con él. Además también se encuentra rotado los ángulos de Euler (θ, ϕ, ψ) , es decir, que este sistema de referencia se desplaza y rota siguiendo al helicóptero. Su plano $x - y$ es paralelo al plano de las hélices, \vec{i} apuntará siempre en la dirección delantera del quadrotor y \vec{j} apuntará en una dirección lateral. La figura 2.1 muestra la estructura simplificada del quadrotor y su «Body Frame».

Para obtener el marco \mathcal{O}^q a partir del \mathcal{O}^i se aplicarán las siguientes transformaciones por orden:

1. Se toma el marco de referencia inercial \mathcal{O}^i y se traslada su origen al c.d.m. del quadrotor.
2. Se rota un ángulo ψ a lo largo del eje z, según muestra la figura 2.2. Este ángulo ψ es el primero de los tres ángulos de Euler que definen la orientación del helicóptero en el espacio.

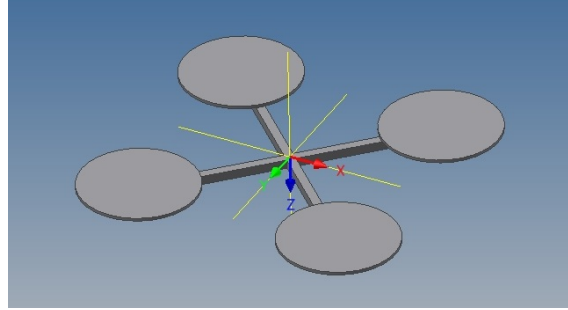


Figura 2.1: Definición del marco de referencia del quadrotor o Body Frame \mathcal{O}^q .

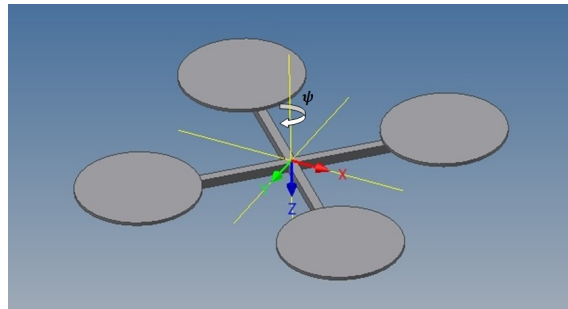


Figura 2.2: Definición del ángulo de rotación ψ .

3. Una vez aplicada la rotación en z , al nuevo marco de referencia se le aplica una rotación de un ángulo θ en el eje y según muestra la figura 2.3. De esta forma se define el ángulo θ que es el segundo ángulo de Euler.
4. Una vez aplicada esta segunda rotación, al marco doblemente rotado se le rota de nuevo, esta vez un ángulo ϕ en el eje x para obtener la orientación final del quadrotor. Figura 2.4.

Cualquier posible orientación del helicóptero puede reducirse a la composición de estas tres rotaciones. Por lo tanto, el conjunto $\{x, y, z, \theta, \phi, \psi\}$ define unívocamente la situación del quadrotor en el espacio.

Vemos ahora como pasar del marco \mathcal{O}^q , al marco \mathcal{O}^i . Imaginemos que tenemos un vector $v^q = (v_x^q, v_y^q, v_z^q)^T$ expresado en el sistema de referencia del quadrotor, y queremos expresarlo en el sistema de referencia de tierra. Si $v^i = (v_x^i, v_y^i, v_z^i)^T$ son las proyecciones del vector en el marco de tierra, la siguiente ecuación² nos dice que:

²La ecuación 2.1 es muy importante ya que nos permite trasladar las medidas de los sensores que están situados en el quadrotor (como la aceleración medida por la IMU) al sistema de tierra que es el que utilizará para el control x-y. Se puede encontrar la demostración en [5, sección 1.2].

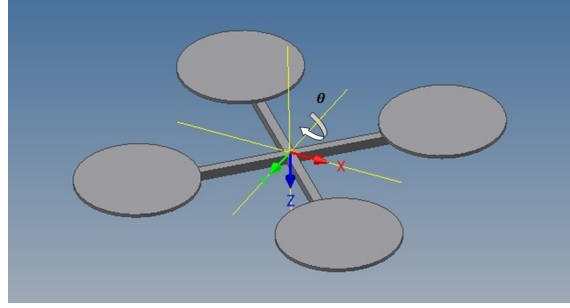


Figura 2.3: Definición del ángulo de rotación θ .

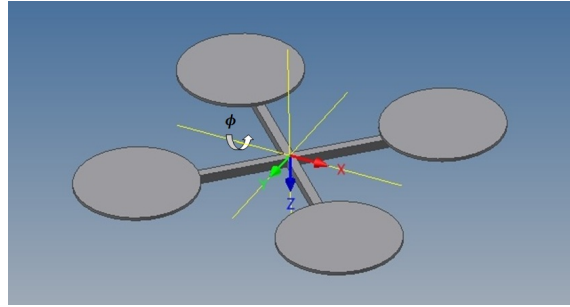


Figura 2.4: Definición del ángulo de rotación ϕ .

$$v^i = \begin{pmatrix} \cos\theta\cos\psi & \text{sen}\phi\text{sen}\theta\cos\psi - \cos\phi\text{sen}\psi & \cos\phi\text{sen}\theta\cos\psi + \text{sen}\phi\text{sen}\psi \\ \cos\theta\text{sen}\psi & \text{sen}\phi\text{sen}\theta\text{sen}\psi + \cos\phi\cos\psi & \cos\phi\text{sen}\theta\text{sen}\psi - \text{sen}\phi\cos\psi \\ -\text{sen}\theta & \text{sen}\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \cdot v^q$$

$$v^i = R_q^i \cdot v^q \quad (2.1)$$

MARCO DE REFERENCIA DE TIERRA ROTADO EN YAW \mathcal{O}^{T_ψ} .

Este marco, representado en la figura 2.5, es igual que \mathcal{O}^i solo que se encuentra rotado en el eje z el ángulo ψ . Lo definimos ahora aunque no se usará hasta que exponamos la estrategia de control x-y, donde mostraremos las razones por las que es necesario incorporarlo.

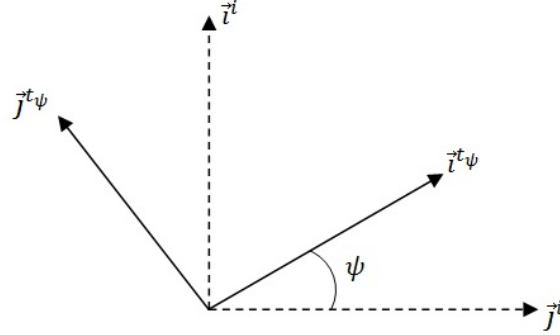


Figura 2.5: Marco de referencia de tierra al cual se le aplica una rotación en ψ .

2.3. Acciones aplicadas al quadrotor.

La figura 2.6 muestra las acciones que ejercen los motores sobre la estructura del quadricóptero. Dichas acciones se encuentran, por tanto, definidas en el sistema de referencia del quadrotor \mathcal{O}^q .

Cada motor ejerce una fuerza de empuje vertical que es proporcional al cuadrado de su velocidad angular:

$$F_i = k \cdot w_i^2$$

La velocidad de cada motor está controlada por un variador de velocidad, el cual es capaz de fijar la velocidad angular del motor proporcionalmente al ancho de un pulso eléctrico que se le manda, de forma que:

$$w_i = k_{var} \cdot \delta$$

De esta manera la fuerza que ejerce cada motor es proporcional al cuadrado del ancho del pulso eléctrico. La fuerza resultante que actuará sobre su c.d.m. será la suma de la fuerza de cada motor:

$$F_{mot} = F_1 + F_2 + F_3 + F_4 \quad (2.2)$$

Las fuerzas de los motores también ocasionan pares que tienden a hacer rotar la estructura, los cuales, son también proporcionales al cuadrado de la velocidad de los motores. Siguiendo con la figura 2.6, se crea un momento en el eje x (τ_x) y un momento en el eje y (τ_y):

$$\tau_x = (F_1 + F_2 - F_3 - F_4) \cdot l/2 \quad (2.3)$$

$$\tau_y = (F_1 + F_4 - F_2 - F_3) \cdot l/2 \quad (2.4)$$

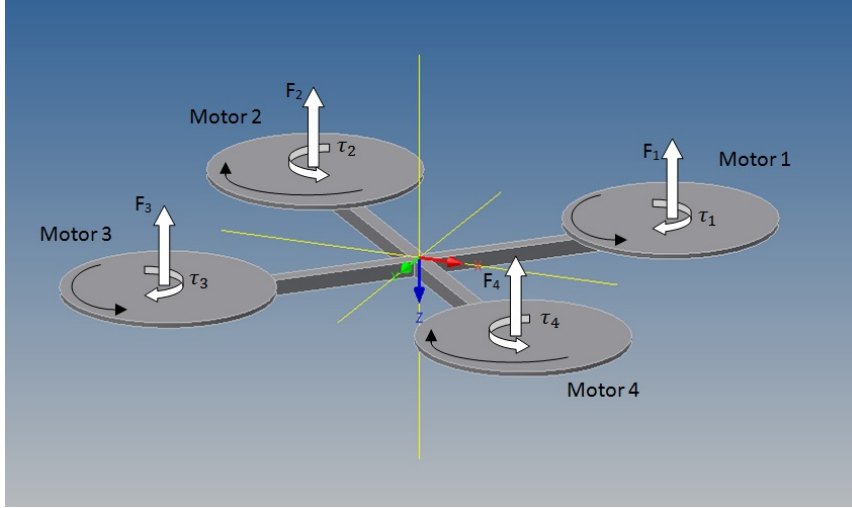


Figura 2.6: Acciones sobre el quadrotor debidas a los motores.

Siendo l la distancia que separa los ejes de rotación de los motores.

Además, cada motor crea también un par que es contrario a la velocidad de giro del motor. Este par surge inicialmente en las hélices ocasionado por la resistencia que ofrece el aire, pero se transmite a los motores y luego a la estructura a través del acoplamiento magnético entre el rotor y el estátor. El par total en el centro de masas será:

$$\tau_\psi = \tau_1 + \tau_3 - \tau_2 - \tau_4 \quad (2.5)$$

Este par es el que tiende a hacer girar al quadrotor en yaw. La existencia de este par es, además, lo que justifica que se invierta el giro de dos motores, de forma que pueda anularse si se escogen apropiadamente sus velocidades.

2.4. Modelo cinemático general.

Vamos a suponer que tenemos al quadrotor en una posición del espacio concreta, definida por las coordenadas (x, y, z) , y con una orientación determinada que viene dada por los ángulos (θ, ϕ, ψ) . Las fuerzas y momentos que sufre (exceptuando la fuerza de la gravedad) son consecuencia del empuje de cada motor, y por lo tanto se definen mucho más fácilmente en el marco de referencia del quadrotor, \mathcal{O}^q , como se ha detallado en el punto 2.3.

Por ejemplo, si queremos que ejecute un giro positivo a lo largo del eje x , se aumentará la velocidad de giro de los dos motores del semiplano $-y$, y se disminuirá la velocidad de giro de los motores del semiplano $+y$. Esto haría al quadrotor girar con una velocidad angular w_x^q creciente (donde el subíndice x indica que

la rotación es en el eje x y el superíndice q indica que está referida al marco de referencia del quadrotor). Pero ahora bien: ¿Cómo ese giro afecta a los ángulos de Euler?. La respuesta no es inmediata, y precisamente, el modelo cinemático es el que establece esas relaciones entre variables. Los resultados siguientes se encuentran debidamente demostrados en [5, sección 2.2], presentamos aquí únicamente los resultados.

Si $(v_x^q, v_y^q, v_z^q)^T$ es la velocidad de traslación del c.d.m. del quadrotor expresado en \mathcal{O}^q , la velocidad de traslación del c.d.m. expresado en el sistema de referencia de tierra \mathcal{O}^i vendrá dada por la ecuación 2.1, de modo que se tiene:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R_q^i \cdot \begin{pmatrix} v_x^q \\ v_y^q \\ v_z^q \end{pmatrix} \quad (2.6)$$

Por otro lado, si $(w_x^q, w_y^q, w_z^q)^T$ es la velocidad angular de rotación en un instante, expresada también en el marco del quadrotor, ésta queda relacionada con los ángulos de Euler $(\theta, \phi, \psi)^T$ a través de la siguiente expresión:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \text{sen}\phi \tan\theta & \text{cos}\phi \tan\theta \\ 0 & \text{cos}\phi & -\text{sen}\phi \\ 0 & \text{sen}\phi \text{sec}\theta & \text{cos}\phi \text{sec}\theta \end{pmatrix} \cdot \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} \quad (2.7)$$

2.5. Modelo dinámico general.

Para obtener el modelo dinámico partimos de las leyes de Newton. Para el movimiento de traslación la segunda ley de Newton establece que:

$$m \cdot \frac{dv}{dt} = F_r$$

Siendo m la masa total del helicóptero, $v = (v_x^q, v_y^q, v_z^q)^T$ la velocidad de traslación del quadrotor expresado en el sistema de referencia \mathcal{O}^q y $F_r = (f_x, f_y, f_z)^T$ la resultante de fuerzas expresada también en \mathcal{O}^q .

Las leyes de Newton únicamente se cumplen en sistemas de referencia inerciales, por lo que la derivada temporal $\frac{d}{dt}$ deberá estar expresada en el sistema de referencia inercial \mathcal{O}^i . Como el marco \mathcal{O}^q rota respecto de \mathcal{O}^i con una velocidad angular $w_q = (w_x^q, w_y^q, w_z^q)^T$, la ecuación de Coriolis determina que:

$$\frac{dv}{dt_i} = \left(\frac{dv}{dt_q} + w_q \times v \right)$$

Con lo que la segunda ley de Newton nos lleva a:

$$m \cdot \left(\frac{dv}{dt}_q + w_q \times v \right) = F_r$$

Desarrollando se concluye que:

$$\begin{pmatrix} \dot{v}_x^q \\ \dot{v}_y^q \\ \dot{v}_z^q \end{pmatrix} = \begin{pmatrix} -w_z^q v_y^q + w_y^q v_z^q \\ -w_x^q v_z^q + w_z^q v_x^q \\ -w_y^q v_x^q + w_x^q v_y^q \end{pmatrix} + \frac{1}{m} \cdot \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (2.8)$$

En nuestro caso, la resultante de las fuerzas F_r será la debida a la acción de gravedad y la fuerza resultante de los 4 motores, por lo que $F_r = F_g + F_{mot}$. La fuerza de la gravedad, expresada en el marco de referencia inercial será $F_r^i = (0, 0, mg)^T$, que se puede expresar en el marco del quadrotor teniendo en cuenta la expresión 2.1:³

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = (R_q^i)^T \cdot \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mg \operatorname{sen} \theta \\ mg \operatorname{cos} \theta \operatorname{sen} \phi \\ mg \operatorname{cos} \theta \operatorname{cos} \phi \end{pmatrix} \quad (2.9)$$

La fuerza de los motores será $F_{mot} = (0, 0, -F)^T$ ya expresada directamente en \mathcal{O}^q , ya que los motores ejercen empuje hacia arriba siempre.

La ecuación 2.8, por tanto, queda:

$$\begin{pmatrix} \dot{v}_x^q \\ \dot{v}_y^q \\ \dot{v}_z^q \end{pmatrix} = \begin{pmatrix} -w_z^q v_y^q + w_y^q v_z^q \\ -w_x^q v_z^q + w_z^q v_x^q \\ -w_y^q v_x^q + w_x^q v_y^q \end{pmatrix} + \begin{pmatrix} -g \operatorname{sen} \theta \\ g \operatorname{cos} \theta \operatorname{sen} \phi \\ g \operatorname{cos} \theta \operatorname{cos} \phi \end{pmatrix} + \frac{1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad (2.10)$$

Para la rotación la segunda ley de Newton toma la siguiente forma:

$$\frac{dL}{dt} = \tau$$

Donde $\tau = (\tau_\theta, \tau_\phi, \tau_\psi)^T$ es el par resultante y L es el momento angular. Siguiendo por el mismo camino, la derivada temporal deberá estar expresada en el marco de referencia inercial, por lo que:

³Una de las propiedades de las matrices de rotación es que son ortogonales, es decir, que son matrices cuadradas cuya inversa coincide con la traspuesta. De modo que:

$$(R_q^i)^{-1} = (R_q^i)^T$$

$$\begin{aligned}
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} &= \begin{pmatrix} \cos\theta\cos\psi & \text{sen}\phi\text{sen}\theta\cos\psi - \cos\phi\text{sen}\psi & \cos\phi\text{sen}\theta\cos\psi + \text{sen}\phi\text{sen}\psi \\ \cos\theta\text{sen}\psi & \text{sen}\phi\text{sen}\theta\text{sen}\psi + \cos\phi\cos\psi & \cos\phi\text{sen}\theta\text{sen}\psi - \text{sen}\phi\cos\psi \\ -\text{sen}\theta & \text{sen}\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \cdot \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} \\
\begin{pmatrix} \dot{v}_x^q \\ \dot{v}_y^q \\ \dot{v}_z^q \end{pmatrix} &= \begin{pmatrix} -w_x^q v_y^q + w_y^q v_x^q \\ -w_x^q v_z^q + w_z^q v_x^q \\ -w_y^q v_x^q + w_x^q v_y^q \end{pmatrix} + \begin{pmatrix} -g\text{sen}\theta \\ g\cos\theta\text{sen}\phi \\ g\cos\theta\cos\phi \end{pmatrix} + \frac{1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \\
\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} &= \begin{pmatrix} 1 & \text{sen}\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\text{sen}\phi \\ 0 & \text{sen}\phi\text{sec}\theta & \cos\phi\text{sec}\theta \end{pmatrix} \cdot \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} \\
\begin{pmatrix} \dot{w}_x^q \\ \dot{w}_y^q \\ \dot{w}_z^q \end{pmatrix} &= \begin{pmatrix} \frac{(-J_{zz}+J_{yy})}{J_{xx}} \cdot w_y^q w_z^q \\ \frac{(-J_{xx}+J_{zz})}{J_{yy}} \cdot w_x^q w_z^q \\ \frac{(-J_{yy}+J_{xx})}{J_{zz}} \cdot w_y^q w_x^q \end{pmatrix} + \begin{pmatrix} \frac{1}{J_{\varphi x}} \tau_\theta \\ \frac{1}{J_{yy}} \tau_\phi \\ \frac{1}{J_{zz}} \tau_\psi \end{pmatrix}
\end{aligned}$$

Figura 2.7: Modelo completo de ecuaciones del quadrotor.

$$\frac{dL}{dt_i} = \left(\frac{dL}{dt_q} + w_q \times L \right) \quad (2.11)$$

Donde $L = J \cdot w_q$ y J es la matriz de inercia. Al ser los ejes del quadricóptero principales de inercia J es diagonal y toma la forma:

$$J = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix}$$

Por lo que la ecuación 2.11 se desarrolla de esta forma:

$$\frac{dL}{dt_i} = \left(J \cdot \frac{dw_q}{dt_q} + w_q \times (J \cdot w_q) \right) = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix} \cdot \begin{pmatrix} \dot{w}_x^q \\ \dot{w}_y^q \\ \dot{w}_z^q \end{pmatrix} + \begin{pmatrix} (J_{zz} - J_{yy}) \cdot w_y^q w_z^q \\ (J_{xx} - J_{zz}) \cdot w_x^q w_z^q \\ (J_{yy} - J_{xx}) \cdot w_y^q w_x^q \end{pmatrix}$$

De modo que finalmente se tiene que:

$$\begin{pmatrix} \dot{w}_x^q \\ \dot{w}_y^q \\ \dot{w}_z^q \end{pmatrix} = \begin{pmatrix} \frac{(-J_{zz}+J_{yy})}{J_{xx}} \cdot w_y^q w_z^q \\ \frac{(-J_{xx}+J_{zz})}{J_{yy}} \cdot w_x^q w_z^q \\ \frac{(-J_{yy}+J_{xx})}{J_{zz}} \cdot w_y^q w_x^q \end{pmatrix} + \begin{pmatrix} \frac{1}{J_{\varphi x}} \tau_\theta \\ \frac{1}{J_{yy}} \tau_\phi \\ \frac{1}{J_{zz}} \tau_\psi \end{pmatrix} \quad (2.12)$$

La figura 2.7 muestra el conjunto de las ecuaciones 2.6, 2.7, 2.10, 2.12, que junto con la definición de las acciones en el punto 2.3 forman el sistema completo de ecuaciones que describen el movimiento completo del quadrotor. Puede observarse que son un sistema de ecuaciones diferenciales no lineales con 6 grados de libertad que en un principio parece difícil de tratar para aplicar alguna ley de control.

2.6. Modelo simplificado del quadrotor.

El modelo descrito en la figura 2.7 abarca el comportamiento íntegro del quadrotor. No obstante dicho modelo contempla más casos de los que se van a dar realmente en la práctica, por ejemplo, el modelo completo es válido sea cual sea la orientación del quadrotor, aunque realmente en la práctica nunca llegará a inclinarse más de 20 grados. Además el modelo completo contempla el término de Coriolis en la derivada temporal de la velocidad, cuando realmente en la práctica nunca se alcanzarán velocidades angulares lo suficientemente grandes como para que sea relevante dicho término.

Así pues se asumen estas dos hipótesis simplificativas:

1. Ángulos pequeños: Suficientemente válida si admitimos ángulos de inclinación máximos de 20 grados. De esta forma la ecuación 2.7 se puede simplificar de la siguiente manera:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \text{sen}\phi \tan\theta & \text{cos}\phi \tan\theta \\ 0 & \text{cos}\phi & -\text{sen}\phi \\ 0 & \text{sen}\phi \text{sec}\theta & \text{cos}\phi \text{sec}\theta \end{pmatrix} \cdot \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} \simeq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} = \begin{pmatrix} w_x^q \\ w_y^q \\ w_z^q \end{pmatrix} \quad (2.13)$$

2. Velocidades angulares pequeñas: De esta forma los términos de Coriolis pueden asumirse nulos. Las ecuaciones 2.10 y 2.12 quedan:

$$\begin{pmatrix} \dot{v}_x^q \\ \dot{v}_y^q \\ \dot{v}_z^q \end{pmatrix} = \begin{pmatrix} -g \text{sen}\theta \\ g \text{cos}\theta \text{sen}\phi \\ g \text{cos}\theta \text{cos}\phi \end{pmatrix} + \frac{1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ -F \end{pmatrix} \quad (2.14)$$

$$\begin{pmatrix} \dot{w}_x^q \\ \dot{w}_y^q \\ \dot{w}_z^q \end{pmatrix} = \begin{pmatrix} \frac{1}{J_{xx}} \tau_\theta \\ \frac{1}{J_{yy}} \tau_\phi \\ \frac{1}{J_{zz}} \tau_\psi \end{pmatrix} \quad (2.15)$$

Con 2.13 y 2.15 tenemos que:

$$\begin{pmatrix} \ddot{\theta} \\ \ddot{\phi} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{J_{xx}} \tau_\theta \\ \frac{1}{J_{yy}} \tau_\phi \\ \frac{1}{J_{zz}} \tau_\psi \end{pmatrix} \quad (2.16)$$

Por otro lado, con las ecuaciones 2.14 y 2.6, y tras operar debidamente se llega a:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} -\text{cos}\phi \text{sen}\theta \text{cos}\psi - \text{sen}\phi \text{sen}\psi \\ -\text{cos}\phi \text{sen}\theta \text{sen}\psi + \text{sen}\phi \text{cos}\psi \\ -\text{cos}\phi \text{cos}\theta \end{pmatrix} \frac{F}{m} + \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (2.17)$$

$$\begin{aligned}
\ddot{x} &= (-\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi)\frac{F}{m} \\
\ddot{y} &= (-\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)\frac{F}{m} \\
\ddot{z} &= g - \cos\phi\cos\theta\frac{F}{m} \\
\ddot{\phi} &= \frac{1}{J_{xx}} \cdot \tau_{\phi} \\
\ddot{\theta} &= \frac{1}{J_{yy}} \cdot \tau_{\theta} \\
\ddot{\psi} &= \frac{1}{J_{zz}} \cdot \tau_{\psi}
\end{aligned}$$

Figura 2.8: Modelo simplificado de ecuaciones del quadrotor

La figura 2.8 muestra las ecuaciones 2.16 y 2.17 que forman el conjunto de ecuaciones simplificadas del quadrotor. Éstas ya se ve que son algo más intuitivas y serán las que van a ser utilizadas tanto para el diseño de las estrategias de control, como para las simulaciones en ordenador.

3. Descripción de la plataforma.

En esta sección se describe cual ha sido la solución obtenida que constituye el estado final de la plataforma. El conjunto de toda la estructura, tal y como ha sido pensada, se divide en cuatro bloques, o módulos, independientes:

1. El grupo de los motores. Incluye los variadores, las hélices, la estructura de soporte, la batería, y, en general, todo aquello que es necesario para que cualquier quadrotor pueda volar.
2. El módulo de estabilización. Que consta de un microcontrolador junto con una unidad de medida inercial (IMU) y un sensor de altura. Garantiza que el quadricóptero pueda mantenerse en vuelo estable con un control sobre sus 4 grados de libertad (θ, ϕ, ψ, h) .
3. Placa para vuelos en exteriores. Que cuenta con un sensor GPS, un receptor de radio digital y un receptor de radio-control (rc) para el manejo con emisora.
4. Un mini-PC u ordenador de abordo. Donde se podrán programar algoritmos de control, y donde se ejecutarán las tareas de mayor coste computacional.

El tener una estructura modulada aporta muchas ventajas ya que cada módulo es independiente del resto. Por ejemplo, si se quiere volar en interiores se podría eliminar la placa para vuelos en exteriores aligerando el peso total; o, si se quieren probar nuevos algoritmos de control sobre el roll, pitch y yaw basta con implementarlos en el módulo de estabilización sin hacer ningún cambio en ningún otro sitio. También se facilita mucho la sustitución de componentes y detección de errores.

La figura 3.1 muestra un esquema conceptual de la plataforma desarrollada. Se puede apreciar que la estructura está jerarquizada. El nivel más básico sería el grupo de motores los con sus variadores, estos variadores reciben una señal de control δ_i y se encargan de controlar la velocidad del motor de tal forma que siempre sea proporcional a δ_i .

El segundo nivel es el módulo de estabilización; este módulo está pensado para que el microcontrolador reciba unas señales de referencia $(\theta_{ref}, \phi_{ref}, \psi_{ref}, h_{ref})$ como datos de entrada, y ejecute un algoritmo de control que genere las señales para cada motor $(\delta_1, \delta_2, \delta_3, \delta_4)$ que garanticen que el quadricóptero va a posicionar en esas referencias.

El tercer nivel es el control que se realiza sobre las referencias⁴ $\theta_{ref}, \phi_{ref}, \psi_{ref}, h_{ref}$. La plataforma se ha desarrollado para que se pueda actuar sobre ellas de dos maneras diferentes:

⁴Esto es lo que se llama un control en dos capas. Se tiene un algoritmo de control ejecutándose en un nivel mas bajo (en nuestro caso este algoritmo es el del módulo de estabilización), y sobre este se ejecuta otro control a un nivel superior.

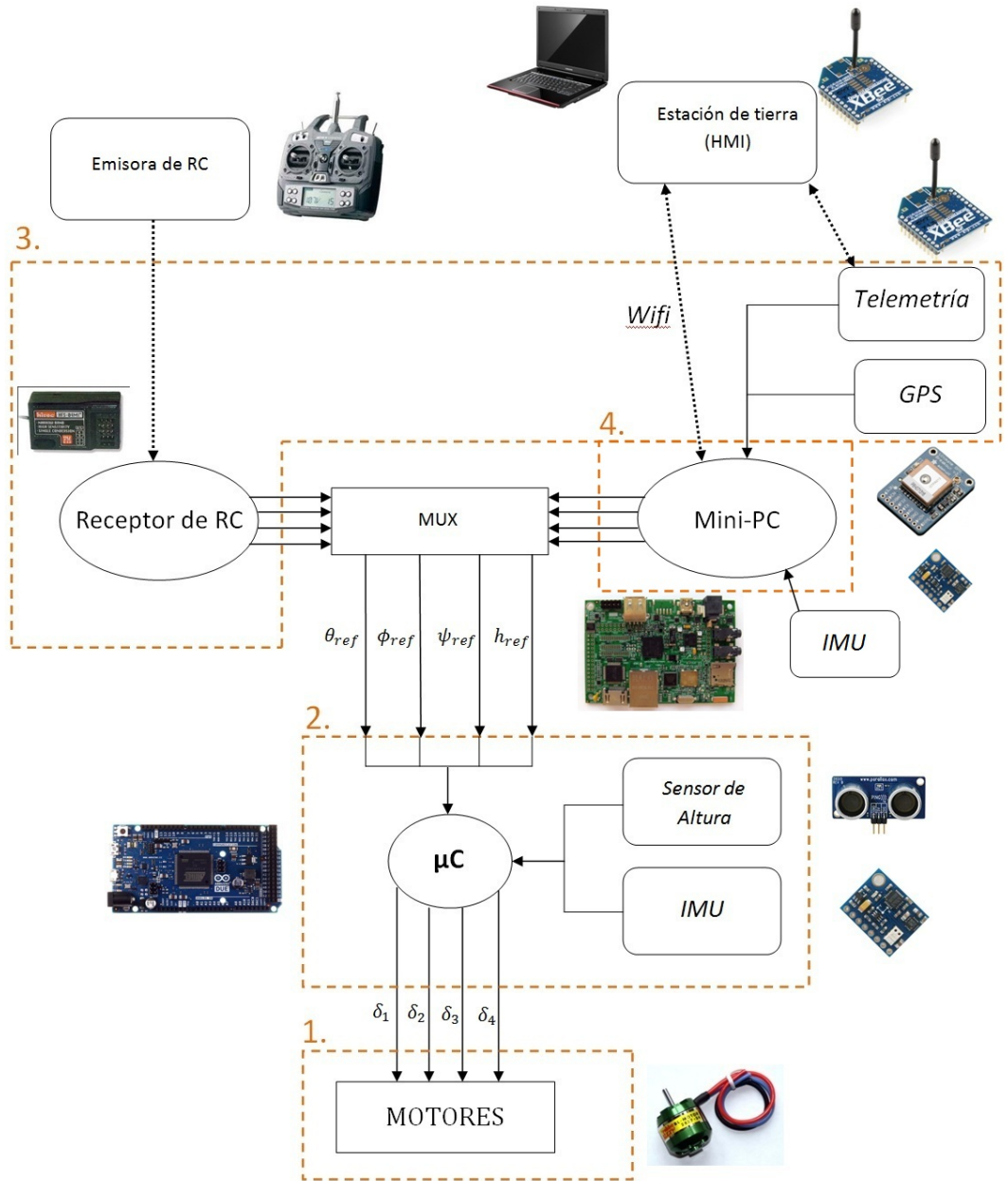


Figura 3.1: Esquema conceptual de la plataforma desarrollada



Figura 3.2: Estructura de soporte empleada

- A través de la emisora de radio-frecuencia, en la que cada referencia es asignada a un canal de la emisora.
- A través del mini-PC programando en él un algoritmo de control de posición. Generalmente estos algoritmos toman como entrada o bien unas coordenadas (x, y, z) , o bien un conjunto de coordenadas que formen una trayectoria. Y generan como salida las referencias $(\theta_{ref}(t), \phi_{ref}(t), \psi_{ref}(t), h_{ref}(t))$ que hacen que el quadrotor ejecute dicha trayectoria.

En lo que sigue se detalla cada uno de estos 4 módulos tanto a nivel de hardware como a nivel de software.

3.1. El grupo de motores.

3.1.1. Estructura de soporte y tren de aterrizaje.

En primer lugar es necesario encontrar una estructura que sea capaz de albergar toda la aparamenta electrónica que se tiene que montar. Se ha utilizado una estructura genérica comercial que se vende por un precio que está en torno a los 30 euros. Las barras que aguantan los motores son huecas y de sección cuadrada, hechas de aluminio aleado. El resto de componentes son de un material compuesto por fibra de carbono y resina epoxi. El tren de aterrizaje, por otro lado, es de plástico. La figura 3.2 muestra el aspecto final de la estructura una vez enlazados todos los elementos.



Figura 3.3: Imagen del motor Robbe Roxy 2827-35

3.1.2. Motores

Los motores incorporados son motores DC brushless (sin escobillas). Constan de imanes permanentes en el rotor y un bobinado con 14 polos en el estator. Son el modelo 2827-35 de la marca Robbe Roxy. Están diseñados para alimentarse con 2-4 celdas LiPo en serie, es decir, con una tensión que va desde los 7.4V (2 celdas) a los 14.6V (4 celdas) y soportan una corriente de 6-9 amperios en régimen nominal, aunque pueden aguantar picos de 10A.

Un aspecto importante de este tipo de motores es el factor Kv (r.p.m/V). Este parámetro nos da una idea de las revoluciones máximas a las que puede girar el motor por cada voltio de tensión aplicada. Teniendo en cuenta que la potencia máxima del motor está limitada y que $P(W) = \tau(Nm) \cdot \omega(rad/s)$, un valor bajo de Kv nos indica que, a una tensión concreta, el motor es capaz de generar más par a costa de girar más lento, mientras que con un valor alto ofrece una mayor velocidad de giro a costa de un menor par. Este razonamiento aplicado al aeromodelismo permite afirmar que un motor con bajo Kv está pensado para funcionar con hélices grandes, mientras que uno con alto Kv funcionará mejor con hélices pequeñas.

Este modelo en particular tiene una $Kv = 760 \text{ rpm/V}$, que es un valor bajo. De hecho el fabricante recomienda que se utilice con hélices de 10 pulgadas de diámetro y un paso de 4.5 pulgadas, que son unas hélices relativamente grandes para la potencia del motor.

La figura 3.3 muestra uno de los motores empleados sin montar en la estructura aún. Cabe observar que tiene 3 fases de entrada, esto es porque estos motores no pueden ser alimentados directamente por una tensión continua sino que se ha de alimentar con una corriente trifásica que varía su polaridad. De ésta tarea se encarga el variador.

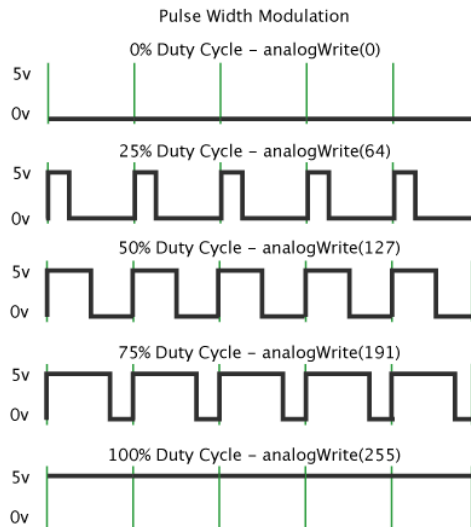


Figura 3.4: Generación de una tensión media constante mediante PWM

3.1.3. Variadores

El variador es un dispositivo electrónico que es capaz de controlar la velocidad de giro del motor. La velocidad de referencia deseada se le envía, generalmente, a través una referencia de voltaje, que suele expresar modulando el ancho de pulso (PWM) de una tensión digital, según muestra la figura 3.4. De esta forma un nivel de 0 de tensión se corresponde con el motor parado y un nivel de 5V se corresponde con la velocidad de giro máxima.

Los variadores garantizan que no se pierda el sincronismo en ningún momento ya que ello provocaría que el motor se parase. En los motores de naturaleza síncrona (como éstos) las causas que llevan a una pérdida de sincronismo son fundamentalmente dos: en los procesos de arranque o cambio de velocidad y cuando se dan variaciones del par resistivo.

La velocidad de giro en estos motores depende directamente de la tensión que llega a sus bornes. De modo que el variador realiza una doble tarea: por un lado cambia la polaridad de la tensión en cada fase de tal forma que nunca se pierda el sincronismo, y por otro lado ajusta la tensión para controlar la velocidad del rotor.

Todo ello lo hacen de manera automática y resulta transparente para el usuario. En nuestro caso los variadores empleados son de la marca YGE, y corresponden con el modelo YGE 30i. Soportan una corriente máxima de 30A, lo que está muy por encima del consumo del motor (10A máximo). La figura 3.5 muestra una foto.



Figura 3.5: Variador YGE 30i



Figura 3.6: Hélices EPP1045 empleadas en el quadrotor

3.1.4. Hélices.

Siguiendo con la recomendación del fabricante de motores (punto 3.1.2) se han empleado unas hélices de 10 pulgadas de diámetro y 4.5 pulgadas de paso.

Son unas hélices muy económicas y ello se refleja en los resultados. El plástico del que están hechas no es muy rígido y eso produce que flecten demasiado cuando están aguantando el peso del quadrotor.

Por otro lado se tuvieron que nivelar, ya que ninguna de ellas tenía el centro gravedad justo en el eje de rotación, sino que se encontraba desplazado, ocasionando un aumento de vibraciones en toda la estructura. La nivelación se hizo recortando ligeramente el extremo de la hélice con mayor masa hasta conseguir que el c.d.g. quede justo en el eje.

El motivo por el que se escogieron estas hélices fue porque, al ser de ese material, se evitan posibles daños. Está comprobado que esas hélices no cortan, por lo que por motivos de seguridad se ha escogido trabajar con ellas.

Han sido necesarias 4 hélices. Dos de ellas con giro en sentido horario y dos con sentido antihorario, para poder anular los momentos τ_ψ , como se comentó en el punto 2.3. La figura 3.6 muestra una fotografía de las hélices utilizadas.



Figura 3.7: Batería LiPo 3S1P de la marca Gens Ace

3.1.5. Batería y su soporte en la estructura.

Se han empleado diversas baterías LiPo 3S1P, que quiere decir que están formadas por 3 celdas en serie. En la elección de una batería existe un compromiso entre peso y capacidad, ya que las baterías de mayor capacidad son más grandes y pesan más.

En principio se puede emplear cualquiera siempre que la velocidad de descarga⁵ sea superior a 45A, que es el consumo máximo del quadrotor (10A por motor más 5A extra para la electrónica), aunque en la realidad, durante un vuelo estable, rara vez se superan los 25A.

Las que se han empleado en nuestro caso son baterías expresamente diseñadas para radio-control que soportan una alta velocidad de descarga. Por ejemplo, una con las que se funcionó durante la mayor parte del tiempo se muestra en la figura 3.7. De todas las disponibles esta es la que mayor capacidad ofrecía con 5300mAh. La velocidad de descarga es de 30C, es decir, de 159A, que sobrepasa con creces el consumo en régimen nominal del quadrotor.

Las baterías de LiPo pueden resultar dañadas si se descargan demasiado. Es por ello que algunas cuentan con un sistema de protección que corta el suministro de corriente cuando el voltaje de la batería llega al mínimo para que no resulten dañadas. No es el caso en baterías para radio-control, en cuyo caso, el corte instantáneo de corriente produciría la caída del quadrotor, por lo que no sería un

⁵La velocidad de descarga se define como la rapidez con la que la batería se puede descargar de forma segura. Se suele expresar en referencia a la capacidad: 2C, 3C, 25C... . De esta forma una batería de 2200mAh y una velocidad de descarga de 2C, podría soportar de manera permanente una descarga de hasta 4.4A.

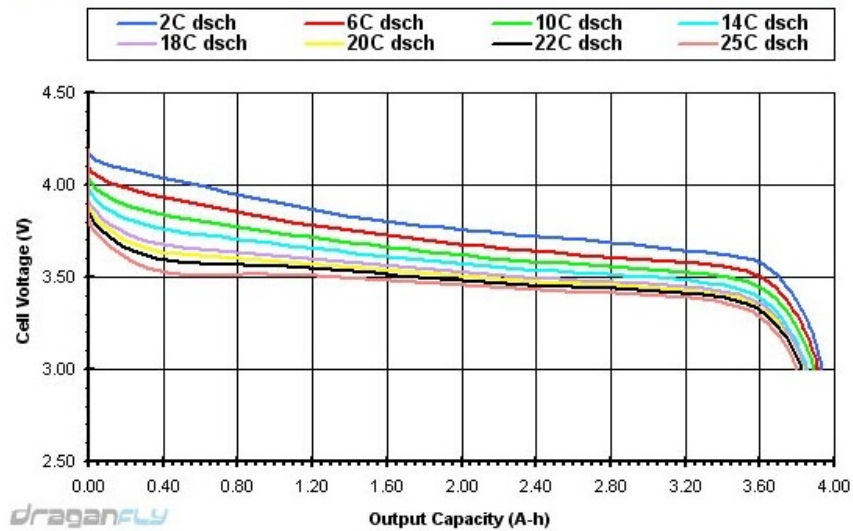


Figura 3.8: Diagrama de descarga de una celda LiPo

sistema de seguridad muy efectivo. En su lugar se dispone de un dispositivo que emite un fuerte pitido cuando alguna de las celdas alcanza el voltaje mínimo, en cuyo caso, se deberá proceder a aterrizar el quadricóptero.

A pesar de que la tensión nominal de la batería sea 11.1V, no siempre se encontrará a esa tensión. Esto hay que tenerlo en cuenta ya que, como se ha comentado en el punto 3.1.2 y 3.1.3, el par que pueden generar los motores depende de la tensión que les llega. La figura 3.8 muestra el diagrama de descarga típico de una celda LiPo. A máxima carga suelen estar en torno a los 4.2V (12.6V para 3 celdas en serie), pero se descargan rápidamente hasta su tensión nominal que está en torno a los 3.7V (11.1V para 3 celdas en serie), y permanecen en el rango de [3.4, 3.7]V durante todo el proceso de descarga hasta que finalmente la tensión cae de manera muy brusca cuando la batería se agota. Es entonces cuando hay que cortar el suministro si no queremos que quede dañada.

Por otro lado también hay que señalar que el tiempo que va a durar la batería antes de que se descargue depende de la capacidad y de la intensidad demandada. Esta relación se puede asumir lineal si la descargamos con una velocidad no muy superior a 1C, es decir que nuestra batería descargándose a 5.3A durará prácticamente una hora. En nuestro caso no es así ya que la estamos descargando a unos 5C, por lo que es de esperar que haya cierto comportamiento no lineal que haga que el tiempo de duración sea algo inferior al que nos indica la capacidad.

También es muy importante indicar que el mayor inconveniente que tiene este tipo de baterías es la dificultad del proceso de carga. Puede ocurrir que el pack de celdas se encuentre desequilibrado, esto quiere decir que una de ellas está más



Figura 3.9: Soporte para la batería LiPo.

cargada que el resto. En una batería convencional no habría problema puesto que las tres se comenzarían a cargar independientemente de su estado actual, la que está más cargada acabaría antes y, mientras acaban de cargarse las otras dos, desearía la energía en forma de calor. Pero en una batería LiPo no ocurre eso. Si seguimos dándole energía una vez cargada comienza a hincharse y puede llegar a explotar. Es por ello que hay que ir comprobando la tensión de cada celda durante todo el proceso de carga y ver si es necesario descargar algún elemento mientras se termina de cargar el resto. De eso se ocupan los cargadores especiales para baterías LiPo.

Se ha diseñado un soporte para la batería de tal forma que ésta quede protegida y sujeta bajo en el tren de aterrizaje como se muestra en la figura 3.9.

3.1.6. Conexionado eléctrico.

En este punto detallamos cuál es el conexionado eléctrico de más bajo nivel, así como los criterios de dimensionado de conductores que se han seguido.

Desde la batería va a ser necesario alimentar a los 4 motores, así como al micro-controlador (explicado en el punto 3.2), a la placa para vuelos en exteriores (punto 3.3) y al PC de abordaje (punto 3.4), por lo que se deberán disponer de 6 cables que salgan de la batería. Además se han puesto dos interruptores para conectar y desconectar toda la electrónica tal y como indica la figura 3.10.

Para dimensionar la sección de los cables el criterio térmico ha sido el más restrictivo, ya que al ser su longitud muy corta apenas había caída de tensión. Muchos fabricantes de cables ofrecen tablas de dimensionado que garantizan que nunca se alcanzará en el conductor una temperatura que pueda dañar al aislante. Tras comparar varias se concluyó que con una sección de 2mm de cobre sería

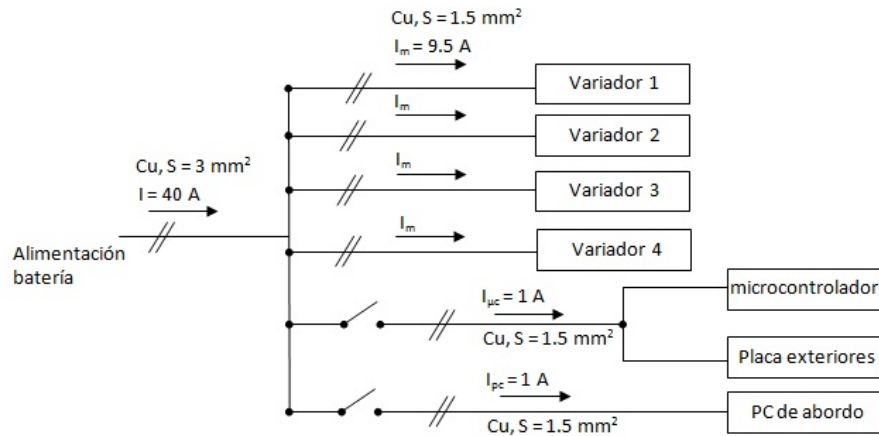


Figura 3.10: Esquema eléctrico de alimentación del quadrotor

suficiente para conducir 40A. Al disponer en el laboratorio de cable de cobre de 3mm y de 1.5mm, se dispuso el de 3mm para el tramo de mayor amperaje y el de 1.5 para el resto de subtramos.

Para toda la electrónica se ha estimado un consumo de 2A. Por lo que la red eléctrica de alimentación está algo sobredimensionada.

La figura 3.11 muestra el aspecto final de la estructura tras la esta primera fase de montaje. En ella se encuentran ya montados los motores, y están conectados a sus respectivos variadores. También está montado el compartimento de la batería y todo el conexionado eléctrico con sus interruptores.

3.1.7. La protección.

Por motivos fundamentalmente de seguridad, aunque también estéticos, se ha diseñado desde cero una protección para las hélices del quadricóptero. Muchos de los quadrotors comerciales no suelen llevar ningún tipo de protección para las hélices, y los que sí que la tienen suele ser de poliestireno expandido, que no es muy resistente a golpes e impactos.

En nuestro caso se hacía necesaria una protección ya que en muchas ocasiones se manipulaba el quadrotor durante el vuelo y a distancias muy cortas, con lo que había mucho riesgo de resultar dañado por alguna de las hélices.

La protección se ha diseñado a medida para éste quadricóptero en concreto. Está formada por tres capas formando un «sándwich». La primera capa es de un



Figura 3.11: Aspecto de la estructura tras la primera fase de montaje

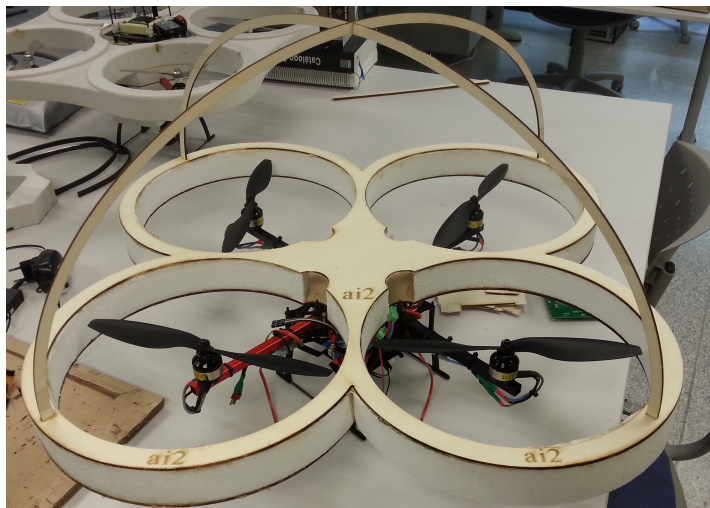


Figura 3.12: Aspecto final de la protección diseñada para el quadrotor

contrachapado de madera de chopo de 3mm, una madera muy ligera pero que es mucho más resistente que la típica madera de balsa que se usa para estas aplicaciones. La capa intermedia es una capa de 4.5cm de poliestireno expandido, y la capa superior es igual que la inferior.

El corte de la madera de contrachapado se efectuó con una cortadora láser de control numérico.

La figura 3.12 muestra el aspecto final del quadrotor con las hélices, motores, todo el conexionado eléctrico y la protección.

3.2. El módulo de estabilización.

Una vez montado todo el grupo de los motores, variadores, conexionado eléctrico, etc. se comenzó a desarrollar el módulo de estabilización. Tal y como se detalla en la figura 3.1, este módulo es el que se encarga de actuar sobre los motores para que el quadrotor siga a las referencias $\theta_{ref}, \phi_{ref}, \psi_{ref}, h_{ref}$ que se le pasan como parámetros de entrada.

Para ello se han empleado tres elementos:

- Una Unidad de Medida Inercial (IMU).
- Un sensor de altura.
- Un microprocesador programable.

En este punto se detallan cuáles son estos elementos concretamente, y cómo actúan.

3.2.1. La IMU.

Una unidad de medida inercial es un sensor que, como mínimo, incorpora un giróscopo de 3 ejes y un acelerómetro de 3 ejes. Aunque existen algunas que además tienen sensores barométricos, de temperatura o sensores magnéticos (o magnetómetros).

En la mayoría de estos sensores es muy sencillo acceder a los datos⁶ en bruto (a_x, a_y, a_z) de los acelerómetros (generalmente expresados en a/g e incluyendo la gravedad), y a las velocidades angulares (w_x, w_y, w_z) dadas por los giróscopos. Pero no resulta nada fácil obtener los ángulos de orientación (θ, ϕ, ψ) a partir de ellas. Generalmente se recurre a filtros de Kalman que combinan las medidas de los giróscopos con las proyecciones de la aceleración de la gravedad para obtener dichos ángulos.

⁶La mayoría de ellas se comunican por I2C y suelen traer librerías en C que incorporan las funciones necesarias para acceder a las medidas.



Figura 3.13: IMU 10DOF v2 de Drotek

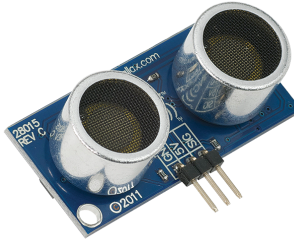


Figura 3.14: Ultrasonic Distance Sensor de Parallax.

Para esta parte se ha utilizado la IMU 10DOF v2 de Drotek [8] la cual se muestra en la figura 3.13. Es un chip que incorpora tres elementos. Por un lado la IMU MPU 6050 de InvenSense Inc [9], que es una IMU muy interesante ya que incorpora un procesador interno (Digital Motion Procesor, DMP) que nos da ya las medidas directas de roll, pitch y yaw así como sus derivadas. Por otro lado también tiene un magnetómetro de tres ejes (HMC5883), que se utilizará para obtener la medida del yaw a gracias a las mediciones del campo magnético terrestre, y un sensor barométrico que se utilizaremos para medir la altura [4].

3.2.2. El sensor de altura.

Para obtener una buena medición de la altura se han empleado dos sensores diferentes. Por un lado se obtiene una medición con el sensor barométrico que incorpora la IMU y por otro lado se ha incorporado también el *Ultrasonic Distance Sensor* de Parallax [10], o también llamado sensor *Ping*, que mide la altura emitiendo pulsos ultrasónicos (figura 3.14).

Las razones que impulsan a tener estos dos sensores son las siguientes:

- Los sensores barométricos tienen la ventaja de que pueden medir cualquier altura ya que se calcula a partir de la variación de presión, pero tienen dos

inconvenientes: el primero es que dan errores típicos de ± 1 m, con lo que resultaría peligroso hacer un control cuando el quadrotor esté a alturas bajas (de 1 o 2 metros); el segundo inconveniente es que la medida es absoluta y por tanto no pueden detectar los obstáculos que tengan por debajo.

- El sensor de ultrasonidos es un sensor muy preciso, con precisiones del orden de centímetros. Pero su mayor inconveniente es que solo da mediciones de altura en un rango de 2cm a 3m, además que necesitan tener debajo una superficie que refleje bien el pulso sónico⁷.

Es por ello que combinándolos eliminamos las desventajas de cada uno de ellos. Se utilizará el barométrico cuando el quadrotor este a alturas superiores a 3m y además el sensor ping estará siempre «vigilando» que no halla ningún obstáculo por debajo. Cuando el quadrotor esté a baja altura se medirá con el *ping* siempre y cuando dé medidas fiables. Esta fusión sensorial se explica detalladamente en [4].

3.2.3. El microprocesador programable.

Con la IMU y los sensores de altura ya se tienen accesibles tanto las medidas θ, ϕ, ψ, h , como sus derivadas $\dot{\theta}, \dot{\phi}, \dot{\psi}, \dot{h}$. Es preciso conocer todas estas magnitudes para poder aplicar cualquier algoritmo de control. Ahora queda incorporar un microprocesador que se encargue de tanto leer esas medidas de los sensores como de recibir las referencias, ejecutar el control y mandar las señales a los motores. El microprocesador deberá, por tanto, comunicarse con el PC de abordó y con la placa de vuelos en exteriores, que son quienes que le mandarán las referencias $\theta_{ref}, \phi_{ref}, \psi_{ref}, h_{ref}$. Por lo que se necesitará uno que tenga al menos las siguientes características:

- 4 salidas PWM. Que se conectaran a los variadores y por donde se envía la referencia de velocidad del motor.
- 2 o mas canales de comunicación serie (UART). Uno para comunicarse con el PC y otro para la placa de exteriores.
- Que resulte sencillo implementar el protocolo I2C. Necesario para leer la IMU.
- Que tenga pines digitales con capacidad para interrupciones. Muy útiles para leer de forma rápida el sensor ping.
- Que tenga al menos una entrada analógica. Para poder medir el voltaje de la batería.

⁷Por ejemplo los pulsos sónicos que emite apenas se reflejan en superficies como el césped, o una alfombra, en cuyo caso el sensor es incapaz de medir ninguna altura incluso estando a menos de 1m de distancia.

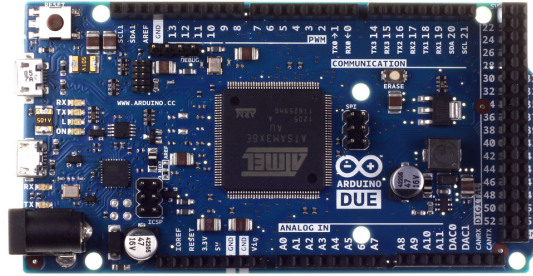


Figura 3.15: Arduino Due

Se ha instalado un *Arduino Due* [11] (figura 3.15). Este arduino cumple de sobra con las demandas, y además existe una amplia comunidad de usuarios activa, por lo que es muy sencillo encontrar información y hay muchísimas librerías actualizadas, sin errores y de uso libre.

Pero sin duda la mayor ventaja es su fácil programación. Se programa con un lenguaje basado en C. Para ello solo hay que descargarse un programa que está disponible en la página oficial de Arduino [11] que permite compilar y cargar el código y que, además, ya tiene incorporadas todas las librerías necesarias para acceder fácilmente a todas las características de hardware.

El programa que se ejecuta en el Arduino cuya misión es la descrita anteriormente se estructura en dos partes:

- *Setup()* (o inicialización): Solo se ejecuta una vez que es al encenderlo. Inicializa los puertos de comunicación, las variables, los pines digitales que se vayan a usar, la IMU, etc.
- *Main loop()* (o bucle principal): Se ejecuta cada 4ms exactamente (sistema de tiempo real) y realiza a cada iteración todas las funciones descritas en el Algoritmo 1.

Se ha tenido bastante en cuenta la seguridad. La primera medida de seguridad es que el control manual está por encima del automático. De esta forma si no respondiera bien el control automático que se ejecuta en el PC, se podría cambiar a control manual desde la emisora de radiofrecuencia y el programa, entonces, omitirá las señales que le envía el PC. (Toda la explicación detallada de cómo funciona el control manual a través de la emisora de radio-control está en [3]).

Otras medidas de seguridad son las siguientes: en caso de que el quadrotor se encuentre inclinado más de 45 grados quiere decir que se ha perdido el control o ha sufrido una colisión, en ese caso se apagan los motores, también se paran los motores en el caso de que el voltaje de la batería sea menor al mínimo (aquí, como trabajo futuro, se debería hacer que ejecutase una maniobra de aterrizaje

Algoritmo 1 Bucle principal Arduino Due.

- Lee el voltaje de la batería.
 - Lee el *buffer* donde se guardan los datos que le llegan del PC.
 - Lee el *buffer* donde se guardan los datos que le llegan de la placa de exteriores.
 - Actualiza los valores de referencia con los datos leídos. Exclusividad: si llegan las referencias de la placa de exteriores omite los del PC.
 - Lee las aceleraciones de la IMU.
 - Ejecuta el algoritmo de fusión sensorial entre el ping y el barómetro [4] y obtiene la medida de la altura.
 - Ejecuta algoritmo de control.
 - Comprueba medidas de seguridad.
 - Manda señales a los motores.
 - Envía información al PC.
 - Sleep(). Hasta que el tiempo de ciclo sean exactamente 4ms.
-

de emergencia ya que parar de golpe los motores, en este caso, no es la mejor alternativa).

3.3. La placa para vuelos en exteriores.

La placa con todos los elementos necesarios para volar en exteriores se ha diseñado desde cero y exclusivamente para este proyecto⁸. Las razones que impulsaron a desarrollarla fueron las siguientes:

- Era necesario disponer de sitio físico para albergar el receptor de radio digital así como el GPS, y el receptor de RC.
- Se invierte mucho tiempo de cómputo en leer la información que emite el GPS y el receptor de RC (casi 200ms en el peor de los casos), por lo que si se incorporasen estos elementos directamente a la placa de estabilización del Arduino Due ya no se podría tener un control de actitud a 250Hz.

⁸En lo referente al módulo de estabilización se contó con trabajos previos realizados por Luis Ródenas en su proyecto de final de carrera [7] que, entre otros, comprobó el funcionamiento de un quadrotor similar con varias IMUs de diferentes fabricantes, desarrolló las placas de conexionado de todo ése módulo y gestiona las comunicaciones con el PC. También desarrolló un algoritmo de lectura del sensor Ping, basado en interrupciones, que permite leer este sensor de manera casi instantánea, lo que permite tener el bucle principal del Arduino Due corriendo a 4Hz.



Figura 3.16: Arduino Micro

- Se quería tener una plataforma completamente modulada.

Esta placa va a tener una doble función. Por un lado alimentar y albergar el módulo de radio digital, el cual, va conectado directamente al PC y, por otro, lado leer toda la información del GPS y del receptor de RC. Por lo que será necesario incorporar otro microcontrolador que gestione estas tareas. En este caso se ha optado por un *Arduino Micro*[11]. Tiene menos capacidad de cómputo y un hardware inferior al *Due*, pero es mucho más pequeño y ligero.

Concretamente, el *Arduino Micro* detecta y gestiona íntegramente los cambios de control automático a control manual y viceversa, lee los canales de la emisora, controla el funcionamiento del sensor GPS seleccionando la frecuencia e información que queremos recibir, también detecta si se ha perdido la señal GPS, o si el sensor está recibiendo correcciones diferenciales (DGPS) o no. Además codifica toda esta información en una trama de bytes, con una cabecera y un CRC, que manda al módulo de estabilización.

Para el correcto funcionamiento de la placa se han necesitado instalar, además, los siguientes elementos auxiliares:

- Un convertor de tensión de tipo *Buck*. Ya que era preciso convertir los 12V que da la batería a 3.3V, que es la tensión a la que se debe alimentar el módulo de radio digital.
- Un convertor de niveles lógicos 3.3/5V. Ya que el *Arduino Micro* tiene las comunicaciones a 5V, mientras que el *Due* las tiene a 3.3V, por lo que podría resultar dañado si no se incorporase.

Estos elementos se muestran, respectivamente, en la figura 3.17.

El módulo de GPS que se ha dispuesto es el *Ultimate GPS breakout v3* de Adafruit [12], del cual se comentarán sus características fundamentales en el punto 4.5.

Por otro lado, la figura 3.19 muestra el receptor de radio digital y el receptor RC. El funcionamiento de estos sensores y su integración en la plataforma se describen en [3].



Figura 3.17: Conversor de tensión (izquierda) y conversor de niveles lógicos (derecha)

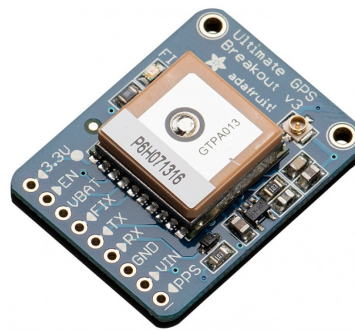


Figura 3.18: Ultimate GPS breakout v3

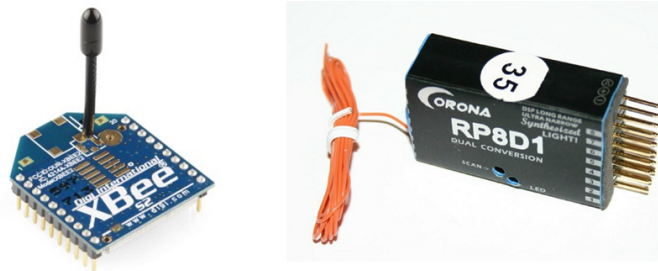


Figura 3.19: Radio Xbee (derecha) y Receptor RC (izquierda)

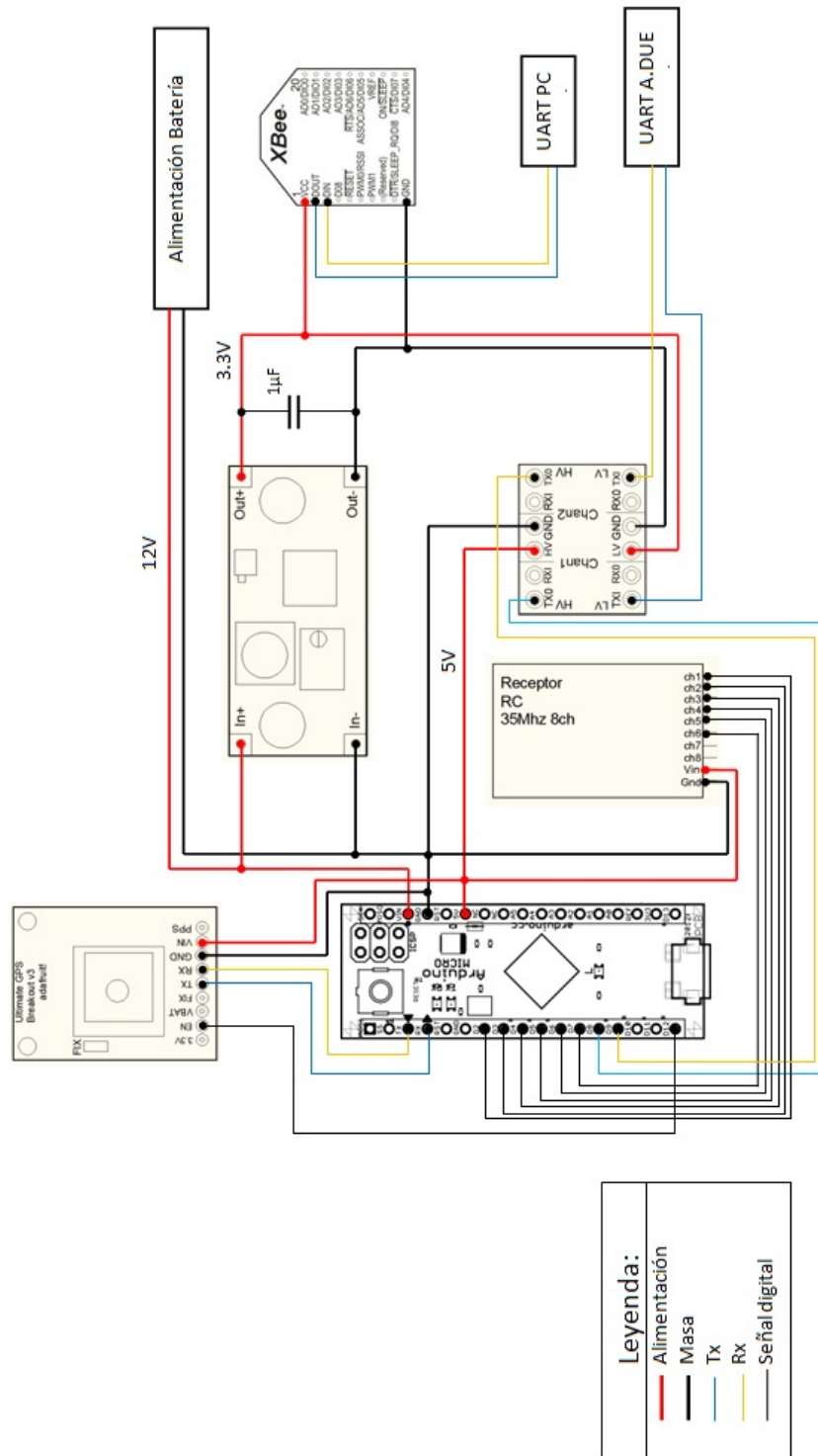


Figura 3.20: Esquema de conexiones de la placa de exteriores

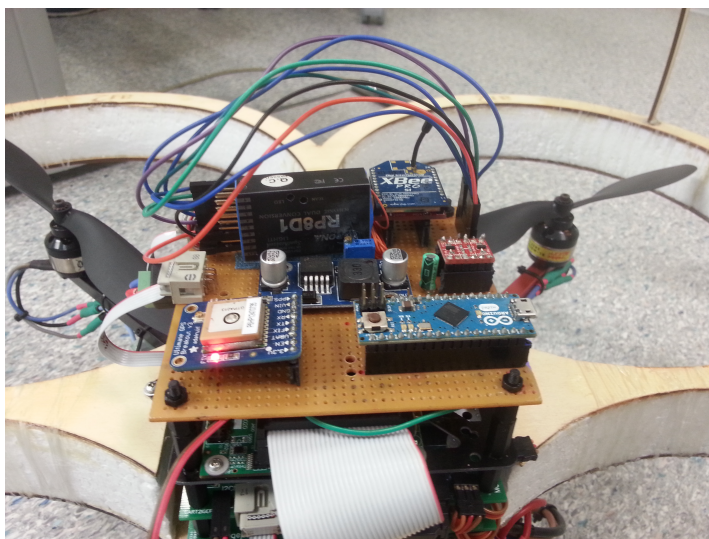


Figura 3.21: Foto de la placa para vuelos en exteriores una vez se hubieron instalado todos los elementos en una placa de prototipado.

Por otro lado, la figura 3.20 muestra como se ha llevado a cabo el conexionado de todos los elementos que conforman esta placa. Se ha instalado un condensador de $1\mu\text{F}$ con objeto de filtrar las pequeñas componentes no continuas que pueda dar el conversor aunque esto no es estrictamente necesario.

El programa que se ejecuta en el *Arduino Micro*, al igual que el del *Due*, tiene dos partes:

- *Setup()* (o inicialización): Le dice al GPS que información queremos recibir y, si queremos que emita a 1Hz o 5Hz (configurable). Además inicializa todas las variables, puertos de comunicación, etc.
- *Main loop()* (o bucle principal): Se ejecuta cada 200ms exactamente. Lo que hace en cada iteración se detalla en el Algoritmo 2.

3.4. El mini-PC u ordenador de abordo.

Como PC de a bordo se ha seleccionado una placa *Igep v2*[13]. Esta placa está desarrollada por la empresa Española Isee, y cuenta con las características de un pequeño ordenador portátil aunque con un menor consumo.

Este elemento es, sin duda, el más caro de todos los que se han incorporado ya que su precio está en los 179 euros, sin embargo, tiene muchas características que lo justifican. Por un lado es capaz de ejecutar cualquier sistema operativo que sea compatible con la arquitectura ARM del procesador, además la CPU tiene

Algoritmo 2 Bucle principal Arduino Micro.

- Comprueba el control. Si el usuario activa la palanca de control manual pone `control_automático = 1`, si no: `control_automático = 0`.
 - `if (control_automático == 1) { «Lee los canales de la emisora.»}`.
 - Comprueba si ha llegado un nuevo mensaje del GPS. Pone `GPSmsg = 1` si hay nuevo mensaje.
 - `if (GPSmsg == 1) { «Obtiene información de posición»}`.
 - Genera la trama de bytes con el mensaje que contiene la información.
 - Le incorpora `#M` como cabecera.
 - Suma todos los bytes e incorpora el resultado al final de la trama como CRC.
 - Envía toda la trama al Arduino Due.
 - `Sleep()`. Hasta que el tiempo de ciclo sean exactamente 200ms
-

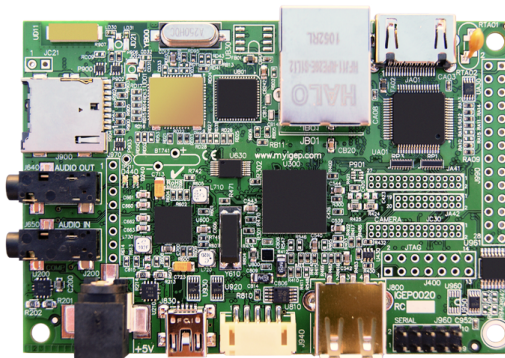


Figura 3.22: Placa Igep v2

una velocidad de cómputo de 1GHz (muy superior a los 84MHz del *Arduino Due*) y también tiene conectividad Wifi, algo muy útil cuando queremos comunicarnos con el quadrotor a cortas distancias ya que es mucho más rápido que la radio.

Actualmente la Igep tiene Xenomai como sistema operativo, que es una versión de Linux en tiempo real. Este S.O. se ejecuta en la Igep desde una micro SD. Toda la configuración del núcleo de Xenomai así como su instalación en la Igep ha sido desarrollada por diversos estudiantes que han colaborado con Pedro García Gil, y que han dejado una guía para poder llevar ésta tarea a «nivel de usuario». Algo de enorme utilidad ya que esta placa está orientada a programadores y no es tan sencilla de empezar a programar como el Arduino, además tampoco existe mucha documentación al respecto.

La Igep se encuentra conectada vía serie (UART) con el Arduino Due y con la radio XBEE. También se encuentra conectada vía I2C a una segunda IMU (exactamente igual que la del módulo de estabilización).

Al igual que con la placa de exteriores, se han tenido que instalar tanto un convertor de tensión, ya que se debe alimentar a una tensión estable de 5V, como un convertor de niveles, ya las comunicaciones son a 1.8V. Éstos son exactamente iguales que los de la figura 3.17.

El programa que se ejecuta en la Igep es *multithread*, es decir, que tiene varios hilos ejecutándose simultáneamente. Es en este dispositivo donde se programan y se ensayan nuevos algoritmos de control, por lo que el programa que ejecuta está en constante cambio, y se añaden y quitan hilos constantemente, sin embargo, existen 5 hilos fundamentales:

- Hilo del Algoritmo de Control. Donde se ejecutará el control x-y.
- Hilo de comunicación con el *Arduino Due*. Que gestiona este canal de comunicación.
- Hilo de telemetría. Que gestiona la comunicación por radio. Descrito en detalle en [3].
- Hilo del Wifi. Que gestiona la comunicación por Wifi.
- Hilo de la IMU. Que se encarga de leer este sensor.

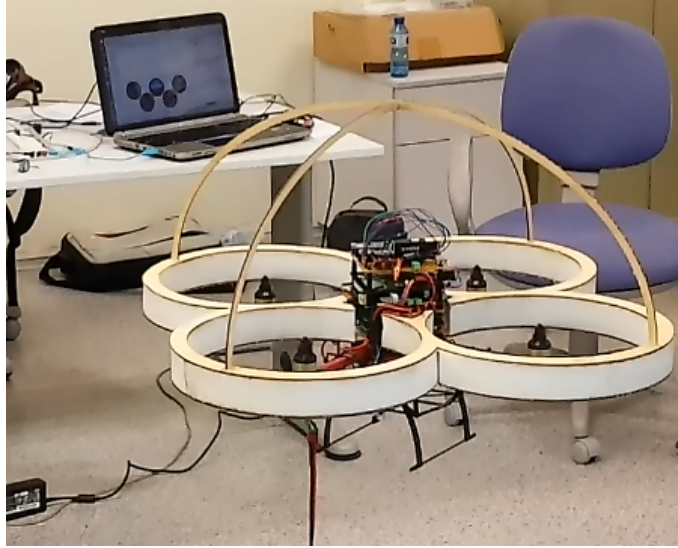


Figura 3.23: Quadrotor en vuelo.

3.5. Resultados finales.

La figura 3.23 muestra el aspecto del quadrotor una vez terminada la primera fase. Se encuentra ejecutando un vuelo en hover.

Puede verse toda la electrónica montada y el ordenador mostrando las variables del sistema de fondo.

Por otro lado, en la figura 3.24 se muestra una gráfica donde se ha representado la medida de pitch θ y roll ϕ , así como sus referencias. Se han cortado de las gráficas los 70 primeros segundos puesto que constituían la parte de arranque de motores, despegue y puesta a punto, de esta forma lo que se ve en las gráficas es cuando el quadrotor está ya en vuelo estable y controlado por el joystick. Se puede ver así la capacidad del quadrotor para seguir a sus referencias en pitch y roll. Se ha señalado en rojo los instantes donde se le introdujeron perturbaciones al sistema para comprobar su capacidad para rechazarlas.

La figura 2.6 muestra las acciones que son aplicadas a los motores para obtener el comportamiento en pitch y roll de la figura 3.24. Se comprueba que los motores se encuentran en torno al 80-90 % de su capacidad, es decir, que están entregando prácticamente el total de su potencia. Como trabajo futuro se plantea sustituir estos motores por otros mas potentes de forma que en régimen nominal estén en torno al 50-60 % (o menos).

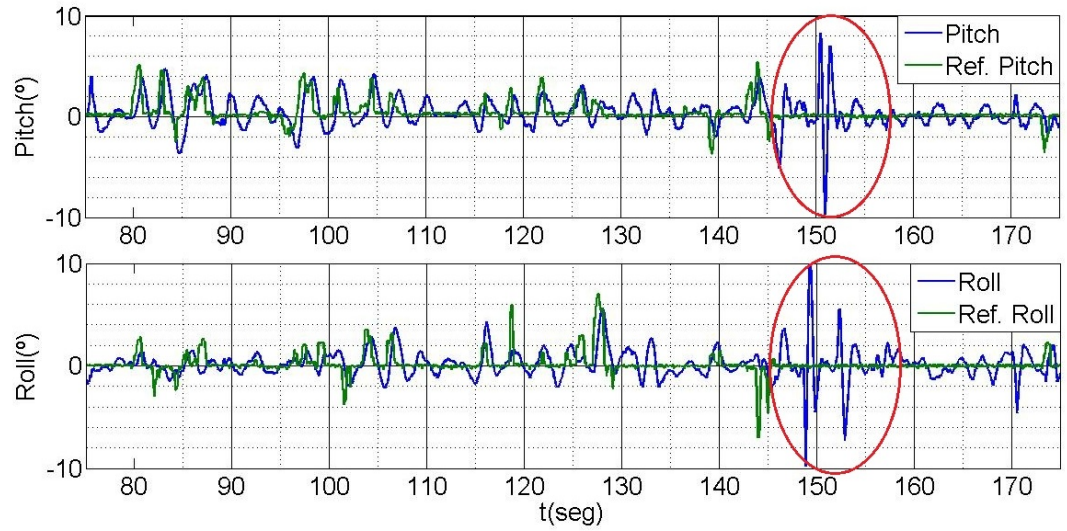


Figura 3.24: Resultados experimentales. Seguimiento de referencias. Pitch y roll.

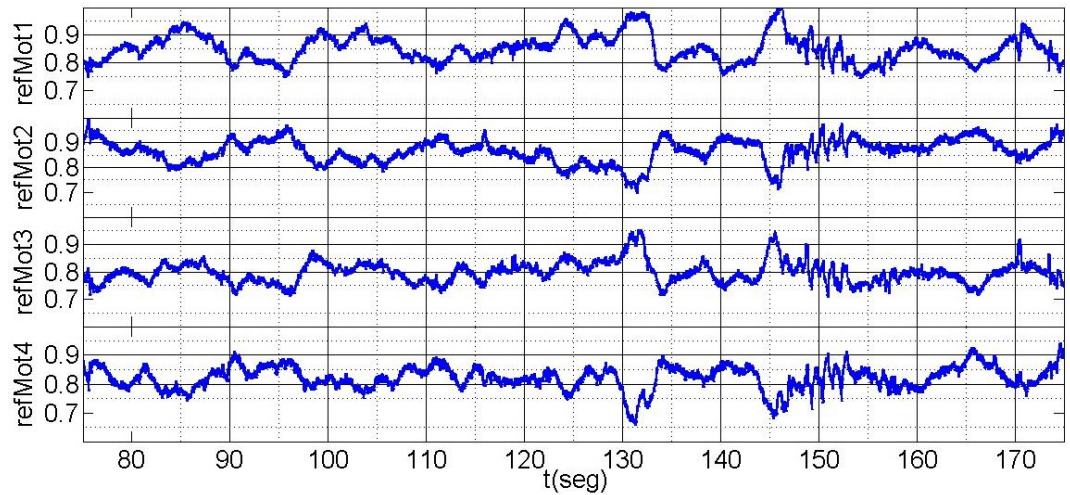


Figura 3.25: Resultados experimentales. Acciones enviadas a los motores.

4. Localización por GPS.

Esta sección tiene como objetivo dar una idea sobre cómo funcionan los sistemas de posicionamiento basados en la trilateración, y más concretamente sobre cómo esto se aplica al GPS. Aunque aquí se va a tratar con carácter introductorio, se puede encontrar una guía donde comenzar a informarse más detalladamente sobre el posicionamiento GPS en [14].

El sistema GPS se lleva utilizando desde hace varios años para obtener medidas de posición en tiempo real de cualquier tipo de vehículo u objeto ya que su principal ventaja es que son válidos en cualquier punto del planeta, y, que tan solo es necesario un único receptor para obtener una medida de posición.

Esto lo convierte en una herramienta de enorme utilidad para conocer y graficar en un mapa la posición de un RPAS concreto. Incluso se utiliza en aviones no tripulados para hacer un control de seguimiento de trayectorias basadas en *WayPoints*. Es un sistema que ofrece una precisión de medida que ronda desde los 2m, en condiciones de buena visibilidad y alta captación de satélites, hasta los 20m en unas condiciones un poco más desfavorables. Por lo que es la solución idónea para RPAS que se van a desplazar en grandes áreas a cielo abierto, en los que un error máximo de 20 metros no es relevante (como aviones o vehículos terrestres).

Sin embargo, al querer aplicarlo a quadrotors los inconvenientes afloran. Por un lado un quadrotor no está pensado para moverse en grandes distancias, ya que su autonomía es reducida y su velocidad de desplazamiento no es muy alta, de hecho la mayor ventaja del quadricóptero es su capacidad para moverse en entornos reducidos, incluso inferiores a 20 metros. Además un quadrotor tampoco está pensado para volar a grandes alturas, donde la captación de señal GPS mejora notablemente ya que se tiene una visión de 180° del cielo y no existe el fenómeno *multipath*. Por otro lado, gran cantidad de veces se requerirá su uso en interiores, donde el GPS queda obsoleto.

Es por ello que, desde mi punto de vista, es necesario replantearse si el GPS es la mejor solución para el problema de posicionamiento en quadrotors. En este caso, su principal ventaja es su sencillez y bajo coste. Por ejemplo el *Ultimate GPS breakout v3* [12] cuesta 40\$ y comienza a dar medidas automáticamente en cuanto capta satélites, algo que lo hace extremadamente sencillo de incorporar al sistema. También es cierto que existe tecnología GPS más compleja y que da mayor precisión como sensores que captan tanto satélites GPS como GLONASS o sensores programables con los que se podrían llegar a corregir los errores empleando técnicas DGPS en tiempo real, incluso se venden estaciones portátiles que generan ya automáticamente esas correcciones DGPS. En estos casos la precisión mejora, pero tanto la complejidad como el precio se disparan, y, además, no se soluciona el problema de posicionamiento en interiores. Como trabajo futuro se

plantea el estudio en detalle de éstas alternativas para comprobar las mejoras y ver si realmente son interesantes para nuestra causa.

Otra posible alternativa es el empleo de técnicas de Sistemas de Posicionamiento Local (*Local Positioning System*, LPS). La mayoría están basadas también en la trilateración. Son de especial interés aquellas basadas en posicionamiento por radio UWB (*Ultra WideBand*), ya que ofrecen una alta precisión, son de mayor alcance que las basadas en ultrasonidos y son válidas tanto para interiores como exteriores. Se plantea también como trabajo futuro el estudio de viabilidad de éstas alternativas para el posicionamiento de quadrotors.

Últimamente la tendencia se ha decantado por usar técnicas GNSS/INS. Estas técnicas están basadas en combinar el GPS con la IMU, y consisten en realizar una primera estimación de posición a partir de las mediciones de la IMU, y posteriormente corregir esa estimación cuando llega una medida del GPS (se detalla su funcionamiento en el punto 5). Diversos usuarios obtienen precisiones que rondan los 0.5m con estos sistemas, pero para ello hay que disponer de IMU's muy precisas, altamente calibradas, compensadas por temperatura y en entornos con pocas vibraciones. Además algunos sistemas incorporan algoritmos complejos de rechazo de medidas erróneas del GPS y compensación de errores, pero para ello es necesario tratar las señales de los satélites «en crudo», por lo que dejan de ser útiles todos los sensores comerciales de bajo coste que directamente dan las mediciones de latitud y longitud tras conectarlos a la alimentación.

Concretamente en este proyecto se ha desarrollado un algoritmo que obtiene una medida de posición empleando técnicas GNSS/INS aprovechando la disponibilidad de la IMU en la plataforma y empleando un sensor GPS comercial. Se puede decir que debido al bajo coste tanto de la IMU como del sensor GPS se trata de un sistema de posicionamiento x-y de muy bajo coste. El algoritmo queda detallado en la sección 5. En este punto, por otro lado, describimos en que se basan todos estos sistemas de posicionamiento, cuáles son las fuentes de error de la señal GPS y cómo podrían corregirse.

4.1. Posicionamiento basado en trilateración.

La trilateración es un método matemático que se usa para determinar una posición relativa a varios objetos. Para ello partimos de que conocemos las posiciones de dos o más puntos de referencia, y las distancias que nos separan de ellos.

La figura 4.1 muestra un ejemplo. Si conocemos la distancia que nos separa en el plano de los puntos O_1 y O_2 , nuestra posible ubicación son los dos puntos que correspondería a las intersecciones de las dos circunferencias de radio r_1, r_2 . Al añadir un tercer punto de referencia el problema se soluciona ya que de los dos

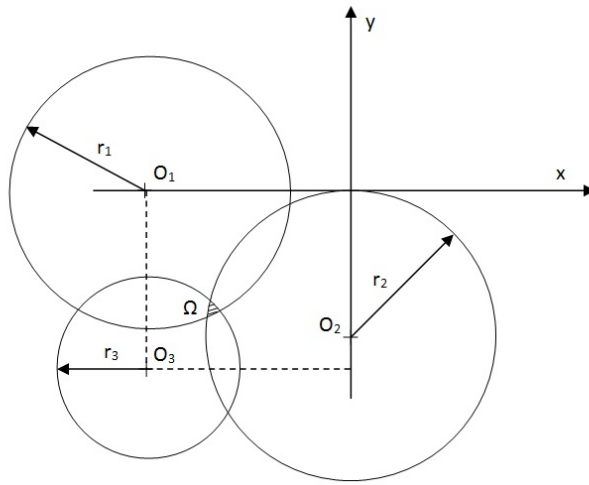


Figura 4.1: Obtención de la posición de un punto en el plano conociendo las distancias que lo separan de otros 3.

posibles uno queda eliminado⁹. Si se conocieran las distancias exactas la solución sería única, pero generalmente esto no ocurre y por ello se encuentra un conjunto de puntos Ω que es la solución al problema. Además, si se tienen más objetos de referencia O_4, O_5, \dots se puede estimar y reducir el error.

En esto se basa el posicionamiento por GPS, así como la mayor parte de sistemas de posicionamiento local (LPS).

En caso de usar un LPS, nosotros escogemos los puntos de referencia O_1, O_2, O_3 situando un sensor en cada uno de ellos. Se pondría otro sensor en el quadrotor de modo que pudiéramos obtener r_1, r_2, r_3 , y, se debería programar un algoritmo que resolviese el problema de posicionamiento con el que obtendríamos la posición (x, y, z) del quadrotor.

Si queremos usar el sistema GPS el problema se complica, ya que la posición de los satélites varía porque que están en órbitas no geoestacionarias, y, a priori, dicha posición no es conocida.

Sin embargo debido al gran desarrollo de esta tecnología durante los últimos años ya existen sensores muy baratos que internamente resuelven el problema, por lo que resulta transparente para el usuario. Aunque si se quiere obtener medidas altamente precisas o corregir los errores se deberá prescindir de estos sensores y

⁹Aquí es importante caer en que los centros de las circunferencias no deben estar alineados ya que, en ese caso, la tercera circunferencia seguiría intersectando en los mismos 2 puntos. Es por ello los objetos de referencia se sitúan de manera triangulada.

adentrarse un poco en la programación de los GPS. En lo que sigue se detalla a grandes rasgos como los GPS obtienen la posición del usuario.

4.2. Determinación de la posición de los satélites y distancia que nos separa de ellos.

Para poder estimar nuestra posición en la tierra debemos conocer la posición de 4 satélites y conocer además la distancia que nos separa de cada uno de ellos.

Los satélites se encuentran constantemente emitiendo señales que, de una forma u otra, llevan codificada la hora atómica en la que la señal fue enviada desde el satélite y, además, también ciertos parámetros orbitales con los que el receptor puede calcular la posición del satélite. Los parámetros orbitales generalmente se suelen llamar *broadcast ephemeris* e incluyen toda la información necesaria para posicionar al satélite en el marco de referencia WGS84.

Los receptores GPS también incluyen relojes atómicos, de modo que conocen la hora atómica a la que se ha recibido la señal del satélite en tierra, t . Como dicha señal lleva codificada la hora a la que salió del satélite, t_s , se sabe el tiempo que ha tardado en recorrer la distancia que los separa $\Delta t = t - t_s$. Las señales son señales electromagnéticas, de modo que se propagan a la velocidad de la luz, por lo que la distancia de separación queda determinada por:

$$d = c \cdot \Delta t$$

El problema no es tan sencillo como medir Δt y multiplicar por c . Es necesario estimar el error del reloj del receptor de tierra y corregir el del satélite, ya que ninguno de ellos es perfecto. Como la onda se propaga a la velocidad de la luz, un pequeño error de reloj se convierte en cientos de metros de error de posición. Por ejemplo, si suponemos que el receptor tiene un error de reloj $1\mu s$, ello quiere decir que se cometerá un error de medida de 300 metros aproximadamente, que, evidentemente, resulta inadmisibile. La hora a la que la señal sale del satélite será:

$$t_s = t_s^{real} + \delta t_{sat}$$

La hora a la que la señal llega al receptor será:

$$t = t^{real} + \delta t_{receptor}$$

Siendo δt_{sat} y $\delta t_{receptor}$ los errores de reloj del satélite y receptor respectivamente.

La Δt realmente contendrá esos dos términos de error:

$$\Delta t = t_s - t = \Delta t^{real} + \delta t_{sat} - \delta t_{receptor}$$

El error de reloj del satélite es estimado por el propio satélite y lo envía en la propia señal en forma de polinomio, por lo que se puede corregir. El error del receptor permanece como incógnita.

La distancia de separación del satélite se puede modelar como:

$$d = c \cdot \Delta t + c \cdot \delta t_{receptor} - c \cdot \delta t_{sat}$$

Nuestras posibles ubicaciones estarán en una esfera de radio d , por lo que se tiene que:

$$c \cdot \Delta t = \sqrt{(x_s(t_s) - x(t))^2 + (y_s(t_s) - y(t))^2 + (z_s(t_s) - z(t))^2} - c \cdot \delta t_{receptor} + c \cdot \delta t_{sat} \quad (4.1)$$

Siendo $(x_s(t_s), y_s(t_s), z_s(t_s))$ la posición del satélite en el instante en el que mandó la señal y $(x(t), y(t), z(t))$ nuestra posición en el instante actual. Vemos que tenemos entonces 4 incógnitas $(x, y, z, \Delta t_{error})$, por lo que por ello es necesario tener al menos 4 satélites en visión. De esta forma se tiene un sistema con 4 ecuaciones como la 4.1 que si se resuelve apropiadamente obtendremos nuestras coordenadas x, y, z en el marco de referencia WGS84. Luego habrá que transformarlas para expresarlas como medidas de latitud y longitud.

4.3. Fuentes de errores.

Hay que mencionar que la ecuación 4.1 es la ecuación más simplificada que se puede obtener ya que no tiene en cuenta ninguna fuente de error más que los relojes. Sin embargo hay muchas otras que no se han contemplado y que afectan a la precisión de manera significativa.

En primer lugar la velocidad c no es constante a lo largo de todo el recorrido ya que depende de la capa de la atmósfera que atraviesa (por ejemplo c varía en la ionosfera) y de las condiciones atmosféricas existentes. Por otro lado la posición del satélite $(x_s(t_s), y_s(t_s), z_s(t_s))$ tiene también un error ya que se estima a partir de ciertos datos proporcionados. También nos encontramos con errores debidos al *multipath*, ya que puede ocurrir que la señal que analicemos no provenga del satélite directamente sino que venga reflejada de alguna superficie. Por supuesto,

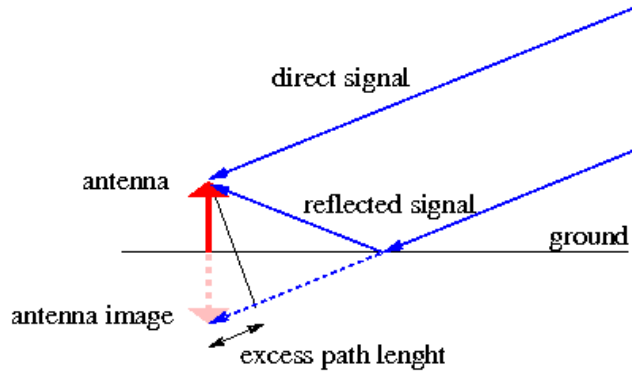


Figura 4.2: Fenómeno multipath.

siempre existe también cierto ruido aleatorio.

La ecuación 4.1 se puede reescribir como:

$$p^s(t, t_s) = \sqrt{(x_s(t_s) - x(t))^2 + (y_s(t_s) - y(t))^2 + (z_s(t_s) - z(t))^2}$$

$$c \cdot \Delta t_{medida} = p^s(t, t_s) - c \cdot \delta t_{receptor} + c \cdot \delta t_{sat} + c \cdot \Delta t_{ionos} + c \cdot \Delta t_{atm} + \delta sat(t) + \delta mult(t) + \eta(t) \quad (4.2)$$

Donde $c \cdot \Delta t_{ionos}$ es el error cometido por la variación de la velocidad c en la ionosfera, $c \cdot \Delta t_{atm}$ es el error debido a condiciones atmosféricas particulares, $\delta sat(t)$ representa el error en la posición del satélite, $\delta mult(t)$ es el error debido al *multipath* y $\eta(t)$ es un ruido aleatorio en la medida.

Es por ello que los sensores varían mucho su precisión dependiendo del lugar donde te encuentres, o si te encuentras en entornos urbanos, donde el fenómeno *multipath* se potencia enormemente, o en días nublados. Ha habido investigadores y universidades que han desarrollado modelos atmosféricos que permiten corregir los errores debidos a la ionosfera y la troposfera, y existen también algoritmos de detección y corrección del *multipath*. Todas estas técnicas mejoran la precisión pero son bastante complejos. No obstante la solución que más se está llevando a cabo para el problema del error de posicionamiento debido a su relativa simplicidad y alta eficacia son las técnicas DGPS o GPS diferencial.

4.4. Sistemas DGPS.

La técnica DGPS (*Differential GPS*) se basa en la siguiente idea. Si conocemos exactamente las coordenadas donde tenemos situado un receptor GPS, entonces

$$p^s(t, t_s) = \sqrt{(x_s(t_s) - x(t))^2 + (y_s(t_s) - y(t))^2 + (z_s(t_s) - z(t))^2}$$

es conocido, por lo que podemos despejar de la ecuación 4.2 todos los términos de error.

Además el error cometido en ese instante será prácticamente igual para todos los usuarios cercanos (en un radio de unos 50-100 km) ya que se debe mayoritariamente a las condiciones atmosféricas. Entonces la idea que surge es que se determine el error cometido en una estación de referencia fija, con coordenadas conocidas, y dicho error se transmita por radio en tiempo real a los usuarios cercanos, los cuales aplican la corrección.

Esta técnica se está aplicando últimamente, la idea es crear una red de estaciones de referencia del estado, que cubran todo el territorio, y que sean accesibles por cualquier ciudadano. De esta forma se pueden lograr precisiones de 1-4 metros. Existen, por otro lado, numerosas estaciones que no aplican las correcciones en tiempo real, sino que las guardan para que los usuarios puedan descargárselas y aplicarlas luego en post-proceso y, de esta manera, corregir las medidas que tomaron previamente.

Para que funcionen correctamente las estaciones de referencia tienen que cumplir ciertos requisitos:

- No deben aplicar ningún modelo atmosférico que elimine los errores ionosféricos y troposféricos.
- Deben eliminar de las correcciones el error de reloj del satélite y el suyo propio.
- Deben situarse en un lugar donde no se vean afectadas por el *multipath*.
- Las mediciones deben ser simultáneas entre la estación de referencia y los receptores.

De esta forma cuando llega una señal del satélite, la estación de referencia computa de igual manera que un receptor la posición del satélite $(x_s(t_s), y_s(t_s), z_s(t_s))$. Como su posición $(x_{est}, y_{est}, z_{est})$ es conocida, entonces se conoce:

$$p_0 = \sqrt{(x_s(t_s) - x_{est})^2 + (y_s(t_s) - y_{est})^2 + (z_s(t_s) - z_{est})^2}$$

Si a la ecuación 4.2 le restamos p_0 obtenemos el error que se está cometiendo, que es:

$$c \cdot \Delta t_{medida} = -c \cdot \delta t_{est} + c \cdot \delta t_{sat} + c \cdot \Delta t_{ionos} + c \cdot \Delta t_{atm} + \delta sat(t) + \delta mult(t) + \eta(t)$$

Ahora es donde entran los requisitos de la estación de tierra, que debe eliminar $c \cdot \delta t_{est}$ y $c \cdot \delta t_{sat}$, y debe estar situada donde $\delta mult(t) \simeq 0$. Lo que no debe eliminar es $c \cdot \Delta t_{ionos}$ ni $c \cdot \Delta t_{atm}$ que es precisamente el error que se quiere corregir y que será aproximadamente igual para usuarios cercanos. De esta forma la estación de referencia emite la corrección siguiente:

$$\Delta_{DGPS} = c \cdot \Delta t_{ionos} + c \cdot \Delta t_{atm} + \delta sat(t) + \eta(t) \quad (4.3)$$

Hay que destacar que la ecuación 4.3 es válida para un único satélite. Por lo que la estación de referencia deberá obtener una corrección para cada satélite en visión. Por otro lado el receptor deberá aplicar cada corrección a su respectivo satélite.

Esto nos dice que se podríamos crear nuestra propia estación correctora y así obtener medidas GPS muy precisas, pero para ello sería necesario un receptor GPS que nos permitiese acceder a la Δt_{medida} de cada satélite, así como a los datos orbitales¹⁰. De modo que se debería programar todo el proceso de obtención de posición. Se dejaría un sensor en tierra con una radio que transmitiera en tiempo real cada corrección al quadrotor. Se plantea como trabajo futuro ver la viabilidad de esta alternativa.

4.5. Sensores GPS estándar. Ultimate GPS breakout v3.

Los sensores GPS como el *Ultimate GPS breakout v3*[12] son sensores que ya vienen programados. Internamente ejecutan todos los cálculos descritos en los puntos anteriores y ofrecen los resultados. Vamos a particularizarlo para el *Ultimate* ya que es el que se ha instalado en la plataforma, aunque la mayoría son bastante parecidos.

La figura 4.3 muestra el aspecto del sensor una vez lo compras. Tiene diversas características que pueden resultar de utilidad como la posibilidad de alimentarlo con una batería de botón, o una memoria flash interna que permite almacenar datos de posición cada 15 segundos para posteriormente descargarlos. También

¹⁰Los sensores comerciales no suelen dar estos datos. Directamente muestran la solución al problema de posicionamiento y todos los cálculos internos resultan transparentes para el usuario.

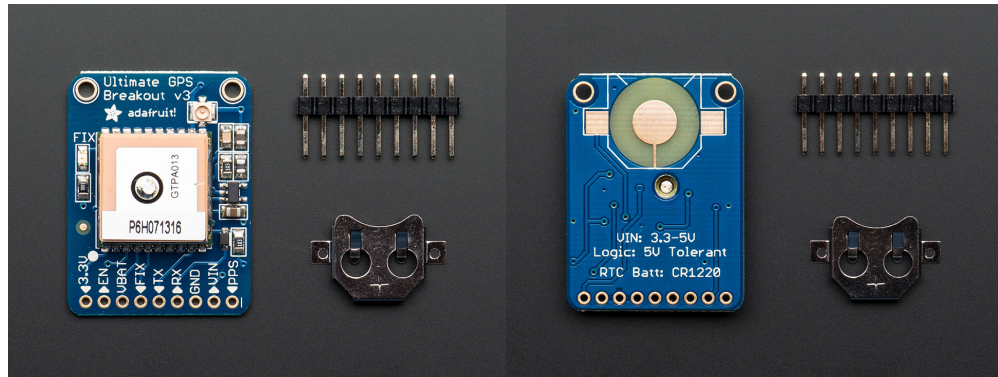


Figura 4.3: Cara delantera y trasera del *Ultimate GPS breakout v3*.

tiene una entrada de antena externa por si no se quiere usar la del propio sensor. Además es capaz de dar información de posición a una velocidad configurable de 10Hz, 5Hz o 1Hz. Para interactuar con él tiene en total 9 pines digitales:

- Pin de «3.3V»: Ofrece una tensión constante a 3.3V por si se quisiera utilizar para algo. Puede dar hasta 100mA.
- Pin «En» ó *Enable*: Es un pin que cuando se le aplica una tensión igual a la de *ground*, el GPS se apaga. Se enciende cuando esta a una tensión igual a *Vin*.
- Pin «VBat»: Está conectado directamente a la entrada de batería de botón. Se utiliza por si no se quiere poner la batería detrás del GPS y se quiere poner en otro lado del circuito.
- Pin «Fix»: Este sensor tiene un led incorporado que parpadea cada segundo si el GPS está buscando satélites, o parpadea cada 15 segundos si el GPS tiene ya los satélites suficientes como para posicionarnos. Este pin funciona de forma exactamente igual, por si se quiere poner el led en cualquier otra parte del circuito.
- Pin «Tx»: Por donde envía toda la información de posicionamiento, satélites en visión, velocidad, etc. Se manda vía serie (UART) y en formato NMEA.
- Pin «Rx»: Donde podemos mandar ciertas tramas de bytes que el sensor entiende y le dicen que información queremos que nos dé y a que velocidad.
- Pines «Gnd» y «Vin»: Con los que se alimenta al sensor. Tiene incorporado un regulador de tensión que permite alimentarlo a una tensión variable entre 3.3-5 V.
- Pin «pps»: Es un pin que da un pulso de 50ms cada segundo. Se utiliza por si se quiere sincronizar el microcontrolador con los relojes de los satélites.

Otra característica muy interesante del sensor es que es capaz de recibir y aplicar las correcciones DGPS si existe algún emisor cercano. Por otro lado también tiene un algoritmo de detección y compensación del fenómeno *multipath*.

El GPS ofrece la información de salida en formato *NMEA 0183*. Es un protocolo definido por la *National Marine Electronics Association*, que se basa en el uso de caracteres ASCII para transmitir la información. Tiene una serie de reglas concretas como cualquier protocolo, por ejemplo, cada mensaje comienza con el símbolo del dólar, \$, y los siguientes caracteres dan información sobre el mensaje que viene. Por otro lado, cada dato útil va separado por una coma del resto. Por ejemplo esta es una trama de información generada por el *Ultimate GPS* que se leyó desde el Arduino Micro:

```
$GPGGA,125849.000,3928.6977,N,00020.0164,W,2,8,1.04,54.8,M,51.9,M,0000,0000*79
```

- \$GP: indica que la información viene de un GPS, *GL*, si fuera GLONASS.
- En general \$GPGGA indica que esta cadena es la «Global Positioning System Fix Data». Que incluye toda la información relevante de posicionamiento.
- 125849.000 (12:58:49) es la hora a la que fueron tomados esos datos.
- 3928.6977,N indica que nuestra latitud es 39^o 28.6977' Norte.
- 00020.0164,W indica que nuestra longitud es 000^o 20.0164' Oeste.
- 2, indica la calidad del fix. Un 2 quiere decir que el sensor está recibiendo correcciones diferenciales, un 1 sería que tiene señal GPS no corregida, y un 0 indica que no hay señal GPS.
- 8, son el número de satélites en visión.
- 1.04 es un parámetro que indica la precisión de la medida de posición que está dando.
- 54.8, M, es la altura en metros.
- El resto es el CRC e información sobre la estación DGPS si se están recibiendo correcciones.

Si introducimos esas coordenadas en cualquier mapa, nos dará la ubicación donde fue tomada esa lectura. Se puede ver en la figura 4.4. Existen muchas más frases NMEA que dan otro tipo de información, se puede encontrar más información sobre el protocolo NMEA, por ejemplo, en [15].

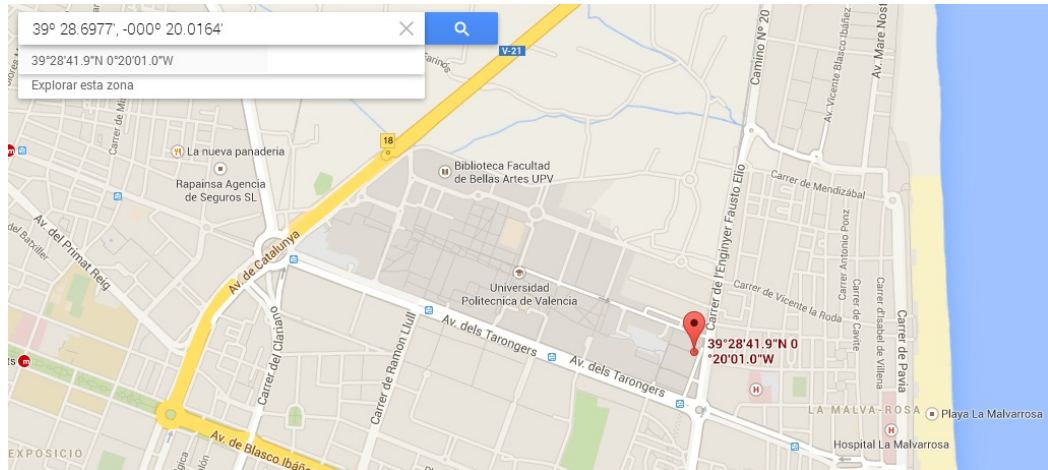


Figura 4.4: Captura Pantalla de *Google Maps* donde se ha introducido la información NMEA dada por el GPS

```

$PGTOP,11,2*6E
$GPGGA,125849.000,3928.6977,N,00020.0164,W,2,8,1.04,54.8,M,51.9,M,0000,0000*79
$GPGSA,A,3,15,09,05,10,08,07,28,26,,,,,1.34,1.04,0.84*05
$GPGSV,3,1,11,26,66,339,45,05,65,206,42,28,51,092,37,09,46,046,40*7F
$GPGSV,3,2,11,33,41,203,40,08,39,045,37,15,34,301,34,10,17,156,39*76
$GPGSV,3,3,11,24,13,243,18,07,10,054,35,02,01,203,*42
$GPRMC,125849.000,A,3928.6977,N,00020.0164,W,0.02,254.33,270314,,D*76
$GPVTG,254.33,T,M,0.02,N,0.03,K,D*3A

```

Figura 4.5: Información de posicionamiento en formato NMEA dada por el *Ultimate GPS*.

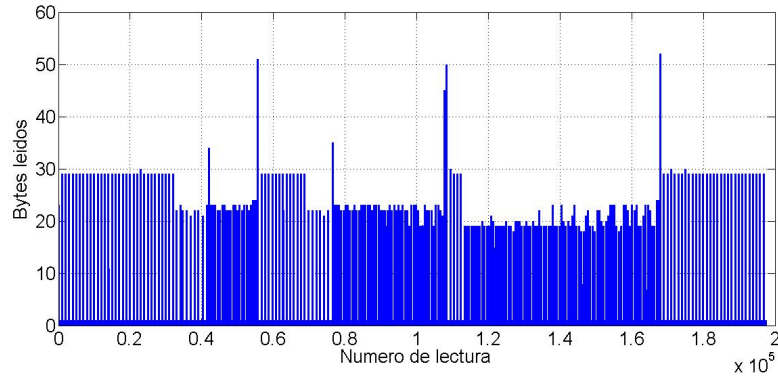


Figura 4.6: Número de bytes leídos cada vez que se vacía el *buffer* del Arduino Micro.

Hay que destacar un matiz importante que es el tiempo que tarda en transmitirse toda la información desde el GPS al microcontrolador¹¹. Aunque en la frase NMEA aparezcan números en realidad están codificados como caracteres, por lo que cada cifra ocupa 8 bits ya sea una coma, un número cualquiera o una letra. Además el sensor no manda únicamente la cadena \$GPGGA, ya que puede mandar hasta un total de 8 cadenas con información diferente. En total pueden ser unos 450 caracteres de 8 bits (3600 bits en total), que se escriben a una velocidad de 9600 bits/seg. Esto quiere decir está durante 375 ms mandando información continuamente hasta que acaba.

Por otro lado, el Arduino Micro tiene una capacidad de *buffer* de 64 bytes (512 bits), por lo que es necesario leerlo al menos cada 50 ms o corre el riesgo de que se llene, en cuyo caso se perderían los nuevos datos entrantes. Este fue un problema que ocurrió de verdad y que costó detectarse. Aparentemente el GPS funcionaba y mandaba información, pero el algoritmo de decodificación nunca encontraba una frase NMEA completa y, por tanto, no podía decodificar nada. Esto era porque se leía el *buffer* del micro una vez por ciclo, es decir, cada 200ms, lo que provocaba que se llenara y se perdiera parte de la información. La solución fue hacer más de una lectura por ciclo de modo que nunca estuviese más de 50ms sin ser leído. La figura 4.6 muestra el número de bytes que se leen del *buffer* del Arduino Micro tras aplicar la solución, se comprueba que en casi 200.000 lecturas nunca se alcanzan los 64 bytes.

Solo queda mencionar que una vez el Arduino Micro recibe la trama de caracteres, ejecuta un algoritmo que devuelve ya como variables de tipo *float* la latitud, longitud, velocidad y como variables de tipo *int* la calidad del fix, numero de satélites, etc. Estas variables se envían a la IGEP que junto con las medidas

¹¹Como se detalló en el punto 3.3 la información del GPS actualmente se lee desde un Arduino Micro, el cual ejecuta el ciclo del programa cada 200ms.

(a_x, a_y, a_z) de la IMU, los ángulos de euler (θ, ϕ, ψ) y a través de un filtro de Kalman Extendido se obtienen las medidas finales de posición y velocidad (x, \dot{x}, y, \dot{y}) . Todo este proceso se detalla en el punto siguiente.

5. Descripción y resultados experimentales del sistema GNSS/INS.

Como se ha comentado al inicio del punto 4 existen varias formas de obtener medidas de posición. Casi todas ellas se basan en el principio de trilateración que se comenta en el punto 4.1. La tendencia se ha decantado por el uso de sistemas que combinan medidas de posición y velocidad GPS junto con medidas de posición y velocidad obtenidas a partir de una IMU. Para ello también es necesario el uso de magnetómetros ya que es preciso obtener la orientación de la IMU respecto al norte geográfico y, muchos de estos sistemas, también usan sensores barométricos para medir la altura, ya que la altura dada por un GPS contiene más error que la medida horizontal.

La fusión de GPS+IMU+Magnetómetro+Barómetro para obtener un posicionamiento horizontal se suele designar como sistema GNSS/INS y es, sin duda, el sistema más ampliamente usado en UAV's y RPAS.

Estos sistemas suelen hacer la fusión sensorial mediante filtros de Kalman Extendidos (EKF). Un EKF, en teoría de control, no es más que un observador que a través de un proceso iterativo y convergente selecciona la matriz de ganancia de la realimentación de tal forma que se minimice la covarianza del error de la estimación. Aunque aquí se plantearan los puntos básicos sobre el EKF necesarios para poder explicar el sistema GNSS/INS implementado se puede encontrar información más extendida en [16] y [18].

Se pueden encontrar en el mercado numerosos sensores de posicionamiento que incorporan esta tecnología. Muchos desarrolladores e investigadores incorporan en sus proyectos uno de ellos. Las fundamentales ventajas son que funcionan de manera casi excelente, además se encuentran altamente protegidos frente a vibraciones, y muy calibrados¹². Son sensores que se conectan y ya están listos para dar medidas altamente precisas. Su mayor inconveniente es el precio, ya que rondan desde los 2.000\$ hasta los 15.000\$ o incluso más.

Por ejemplo, en la figura 5.1 muestra la *Spatial FOG* [17], que es uno de los mejores sistemas GNSS/INS que existe en el mercado. Su precio está en los 37.000\$ y el fabricante asegura obtener precisiones de 0.8m horizontalmente bajo cualquier circunstancia de captación de satélites¹³ y precisiones de 0.008m en los mejores casos. Para ello hay que destacar que el sistema GPS que incorpora es capaz de captar satélites tanto GPS como GLONASS y BeiDou, así como capacidad para el futuro Galileo. La empresa, evidentemente, no menciona nada sobre el software de procesamiento de la señal, pero hay que señalar que estos sistemas tan precisos

¹²sobretudo en lo referente al magnetómetro. Pueden verse las dificultades que entraña la correcta calibración de este sensor en [4]

¹³Cabe destacar que un sensor GPS corriente suele oscilar en torno $\pm 10m$, o más, cuando tiene pocos satélites en visión.



Figura 5.1: Spatial FOG

no se limitan a aplicar el EKF, sino que además suelen incorporar complejos algoritmos de rechazo de medidas malas de GPS y corrección de errores, entre otros.

Una de las ideas de este proyecto, y que se desarrolla a continuación, es desarrollar un sistema GNSS/INS a partir de sensores de bajo coste, con objeto de ver hasta que punto está justificado el alto precio de estos sensores en el mercado. La idea es posteriormente poder aplicar esas medidas para controlar la posición del quadrotor fabricado.

Para ello se parte del Ultimate GPS descrito en 4.5. Se ha desarrollado todo el conexionado del GPS y del microprocesador según se indica en 3.2, por lo que una vez se cerraron las comunicaciones y se comprobó que funcionaban correctamente, ya se disponía de las medidas de latitud, longitud y velocidad (en módulo) en la IGEP donde se puede programar con ellas. Por otro lado tenemos en la plataforma también la IMU que nos da medidas de aceleraciones lineales además de los ángulos de orientación θ, ϕ, ψ . Y también está el magnetómetro que, tras los trabajos de Daniel Verdú en [4], da una medida de la orientación ψ del quadrotor respecto al norte.

Este punto se estructura de la siguiente manera. En la sección 5.1 se hace una breve introducción a los observadores y filtro de Kalman, en 5.2 se detalla cómo se transforma desde las medidas de latitud y longitud dadas por el GPS a medidas de posición local (x,y). En 5.3 se explica un algoritmo que se ha desarrollado para obtener una estimación de las velocidades en los ejes x e y, ya que el GPS únicamente da la velocidad en módulo. Para poder fusionar las medidas del GPS junto con las estimadas por la IMU, ambas deben estar expresadas en el mismo sistema de referencia, por ello, en 5.4 se explica cómo se lleva a cabo esa transformación. Por otro lado, en 5.5 se justifica porqué es necesario incluir el Offset del acelerómetro en el modelo. En el punto 5.6 se comenta el retraso que se observa en la señal de GPS, ello ha empujado a tener que retrasar las medidas del acelerómetro de tal forma que estén en el mismo margen temporal. Los puntos 5.7, 5.8, 5.9 y 5.10 detallan el algoritmo de Kalman aplicado, los resultados obtenidos y posibles

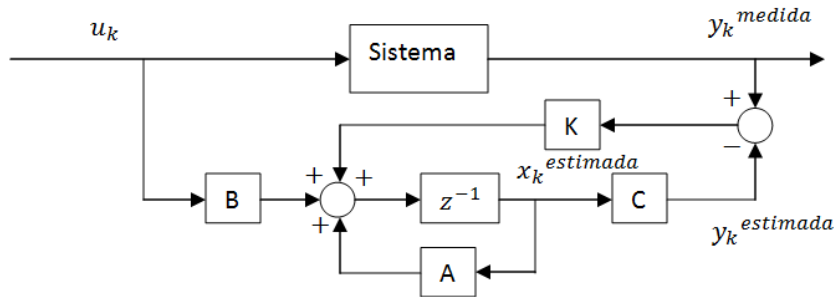


Figura 5.2: Diagrama de bloques de un observador.

mejoras de cara al futuro.

5.1. Ideas fundamentales sobre observadores y filtro de Kalman.

En teoría de control la idea de un observador es, precisamente, la de observar el proceso con objeto de obtener una o más variables que no puedan ser medidas directamente. Para ello se ha de disponer de un modelo del sistema suficientemente aproximado a la realidad. Además han de conocerse las acciones de control u y también se ha de medir, al menos, una de las salidas del sistema.

Un modelo de un sistema puede expresarse de manera discreta según la siguiente ecuación matricial:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \quad (5.1)$$

Donde x_k es el conjunto de todas las variables del sistema e y_k son las variables medidas.

Se puede utilizar el modelo para estimar la medición y_k , y después corregirla con la medida real dada por el sensor. La figura 5.2 muestra la estructura en bucle cerrado del observador de un sistema. La ecuación viene dada por:

$$x_{k+1} = Ax_k + Bu_k + K(y_k - Cx_k) \quad (5.2)$$

Con esta realimentación la medida estimada converge a la real si los autovalores de $(A - KC)$ están dentro del círculo unidad en el plano complejo. Hay que destacar que si x_k converge a la medida real quiere decir que se conocerán todas las variables del sistema, a pesar de estar midiendo únicamente y . Ése es el punto

verdaderamente interesante de los observadores.

Realmente el sistema 5.1 es una aproximación del sistema real. En el sistema real, generalmente, existirá un ruido aleatorio tanto en el modelo como en la medición, por lo que puede reescribirse como:

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k\end{aligned}\tag{5.3}$$

Siendo w_k y v_k ruidos aleatorios. De esta forma el algoritmo de Kalman permite determinar la matriz K de tal forma que se minimice la covarianza del error de medida ocasionada por w_k y v_k . Como se ha comentado antes se puede encontrar información más en detalle en [16] y [18].

5.2. Conversión de las coordenadas latitud y longitud a medidas de posición locales (x,y).

El GPS da medidas de latitud y longitud en grados. Sin embargo para posicionar el quadrotor resulta más interesante expresar su posición en coordenadas (x,y), siendo el (0,0) el punto donde está ubicada la estación de tierra. Para hacer esta conversión es necesario conocer que representan físicamente la latitud y longitud.

La latitud es el ángulo, en grados, que hay desde nuestra posición al ecuador. Mientras que la longitud es el ángulo que hay desde nuestra posición al meridiano de Greenwich. El radio medio de la tierra tiene un valor de 6.371 km. De esta forma la longitud de una circunferencia con su centro en el centro de la tierra, radio $r = 6371$ y que pase por los polos es:

$$l = 2\pi r = 40030,17359 \text{ km}$$

Esto quiere decir que cada grado de latitud equivale a una distancia, d , de:

$$2\pi r/360 = 111,194926 \text{ (km)}$$

O dicho de otra forma, cada segundo de latitud equivale a 30.887 metros. Esto es independiente de dónde esté ubicado el punto, ya que el radio de los meridianos se puede asumir constante.

Para la longitud esto no ocurre. Observando la figura 5.3 se ve que a medida que aumenta nuestra posición en latitud el ángulo de longitud se reduce porque el radio de los paralelos va disminuyendo. Si la longitud es 0 y asumimos la tierra como esférica¹⁴ la relación es igual que para la latitud: cada segundo de

¹⁴En este caso el radio del ecuador es igual al radio polar.

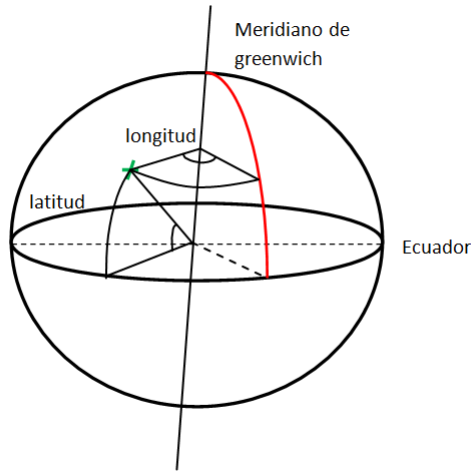


Figura 5.3: Definición de la Latitud y Longitud.

longitud equivale a 30.887 m. A medida que la latitud aumenta, el radio del paralelo disminuye de manera proporcional al coseno de la latitud. De modo que se tiene que un segundo de longitud equivale a:

$$30,887 \cdot \cos(\textit{latitud}) \textit{ (m)}$$

Lo primero que hace el programa es tomar las medidas de latitud y longitud iniciales antes del despegue y guardarlas. Luego en cada instante calcula la diferencia entre la latitud y longitud actuales y latitud y longitud iniciales, y aplica las conversiones. De esta forma se obtienen las coordenadas relativas (x,y).

5.3. Obtención de las proyecciones de la velocidad en cada eje v_x, v_y a partir del módulo V dado por el GPS.

La velocidad podría obtenerse directamente a partir de las variaciones de latitud y longitud. De hecho, esa es la idea que surge de manera natural si nos planteamos cómo obtener una medida de velocidad. Si se hiciera de esta manera, la velocidad V sería una magnitud dependiente, es decir, que errores cometidos en posición se trasladarían a errores de velocidad.

Actualmente muchos de los sensores GPS obtienen la velocidad de manera independiente a la posición. Se basan en el efecto Doppler de las señales electromagnéticas que llegan de los satélites¹⁵. De esta manera posición y velocidad se

¹⁵Realmente suelen emplear técnicas que combinan la medida obtenida por efecto Doppler con la medida que se obtendría observando las variaciones de posición.

obtienen por medios diferentes, con lo que resulta interesante incluir ambas.

Es necesario conocer (\dot{x}, \dot{y}) , pero el dato que proporciona nuestro sensor es $V = \sqrt{\dot{x}^2 + \dot{y}^2}$. Por ello se ha de recurrir de manera forzada a tener en cuenta las variaciones de latitud y longitud. Esto se hace la siguiente forma: Se mira la variación entre nuestra posición actual y la posición dos segundos atrás¹⁶ y, en función de esa variación, se obtiene un coeficiente que pondera a V .

Si definimos:

$$k = \frac{\dot{x}}{\dot{y}}$$

Tras sustituir en $V = \sqrt{\dot{x}^2 + \dot{y}^2}$ y desarrollar se obtienen las componentes (\dot{x}, \dot{y}) como:

$$\dot{x} = \frac{k}{\sqrt{k^2 + 1}} \cdot V \quad (5.4)$$

$$\dot{y} = \frac{1}{\sqrt{k^2 + 1}} \cdot V \quad (5.5)$$

La constante k se puede calcular de la siguiente forma:

$$k \cong \frac{\frac{x_k - x_{k-2}}{\Delta t}}{\frac{y_k - y_{k-2}}{\Delta t}} = \frac{x_k - x_{k-2}}{y_k - y_{k-2}} = \frac{\text{latitud}_k - \text{latitud}_{k-2}}{(\text{longitud}_k - \text{longitud}_{k-2}) \cdot \cos(\text{latitud}_k)} \quad (5.6)$$

Cuando llegue un nuevo mensaje del GPS con medidas de latitud, longitud y velocidad (V), en primer lugar se obtiene la constante k a través de la ecuación 5.6, seguidamente se pondera el módulo según las ecuaciones 5.4 y 5.5. Posteriormente se calculan las posiciones relativas (x, y) tal y como se ha descrito en el punto 5.2. Así se tienen disponibles las medidas (x, y, \dot{x}, \dot{y}) dadas por el GPS.

Ahora, paralelamente, es necesario hacer una estimación de ellas a partir de las aceleraciones de la IMU y fusionarlas a través del filtro de Kalman extendido.

5.4. Rotación de las medidas de la IMU desde el sistema de referencia del quadrotor al de tierra.

Para poder comparar las medidas estimadas por la IMU con las del GPS es necesario que estén expresadas en el mismo sistema de referencia, es decir, en el

¹⁶No tiene porqué ser necesariamente 2 segundos. Se tomo este criterio con objeto de que se filtrara un poco la variación de posición, ya que sino el coeficiente k oscilaba de manera muy brusca.

de tierra \mathcal{O}^i (definido en el punto 2.2).

La IMU está intencionadamente situada en el c.d.m. del quadrotor con sus ejes orientados según los que se marcan en la figura 2.1. Por lo que da medidas en el sistema de referencia del quadrotor \mathcal{O}^q . Tal y como se detalló en el punto 2.2, un vector expresado en el marco del quadrotor puede expresarse en el marco de tierra aplicando la ecuación 2.1.

Llamando $(a_x, a_y, a_z)^T$ las aceleraciones dadas por la IMU y $(a_{xg}, a_{yg}, a_{zg})^T$ las aceleraciones expresadas en el sistema de referencia de tierra, se cumple que:

$$\begin{pmatrix} a_{xg} \\ a_{yg} \\ a_{zg} \end{pmatrix} = \begin{pmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \cdot \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (5.7)$$

Aquí es donde entra el papel del magnetómetro, ya que es necesario tener una medida de ψ que nos de el ángulo de desviación respecto al norte geográfico. Las medidas de θ, ϕ las proporciona la IMU y son muy buenas.

Por otro lado hay que destacar que la IMU también mide la gravedad, por lo que si el quadrotor se encuentra quieto se cumplirá que $\sqrt{a_x^2 + a_y^2 + a_z^2} = 9,81$. Tras aplicar 5.7 la gravedad se mide íntegramente en a_{zg} , por lo que para eliminarla bastará con restarle 9.81 a esa componente.

Es necesario comprobar que los signos estén como toca, tanto los de las aceleraciones, como los de θ, ϕ, ψ . Esto quiere decir que $a_{xg} > 0$ si aceleramos dirección norte, $a_{yg} > 0$ si aceleramos hacia el este, $a_{zg} > 0$ si nos acercamos al centro de la tierra. θ, ϕ, ψ deben estar definidos tal cual se definió en 2.2.

Al aplicar la rotación se observó que a pesar de que θ, ϕ, ψ tuviesen bien los signos, los signos de $(a_{xg}, a_{yg}, a_{zg})^T$ finales no salían como toca. Al final se vio que era porque la IMU expresaba las aceleraciones en un sistema de referencia con z apuntando hacia arriba y no hacia abajo, con lo que solucionó rotando el sistema de referencia de la IMU.

En el punto 5.8 se muestran los resultados que se obtuvieron tras realizar dos ensayos con objeto de comprobar el correcto funcionamiento de ésta rotación una vez programada en el quadrotor.

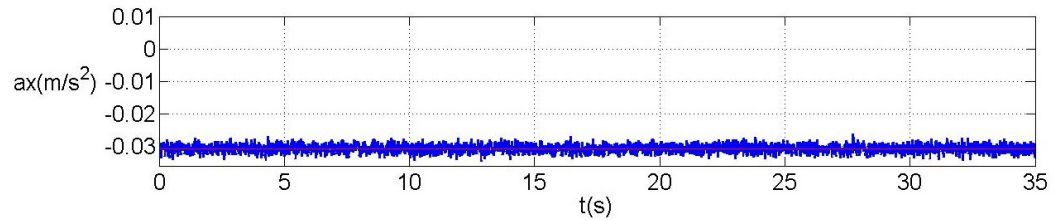


Figura 5.4: Offset del acelerómetro en el eje x.

5.5. Necesidad de estimar el *offset* del acelerómetro.

A medida que se fueron haciendo pruebas se vio que era completamente necesario hacer algo para que el *offset* del acelerómetro se corrigiera de forma automática. El *offset* es el valor promediado que da el acelerómetro cuando supuestamente debería dar cero. Es decir, que si se deja el acelerómetro completamente quieto, en un entorno sin ninguna vibración, la media de los valores que marca no es cero. Ese valor medio es lo que se llama *offset*.

La figura 5.4 muestra el *offset* del acelerómetro en el eje x. Se debe corregir, ya que sino al intentar calcular la posición integrando ésta irá creciendo de forma cuadrática únicamente debido al *offset*. Eso va a provocar que a los pocos segundos la posición se haya desviado bastantes metros del origen. Generalmente lo que se hace es tomar datos durante un rato, calcular la media y restársela. De esta manera se fuerza que se posicione en cero.

Esa solución en nuestro caso no es válida ya que estamos en un entorno muy dinámico. Por ejemplo, si los motores cambian de velocidad, el *offset* cambia¹⁷ o, si se cambian los motores por otros, o se quiere instalar otra IMU, el *offset* cambiará también. No hay que olvidar que se quiere desarrollar un algoritmo que pueda ser adaptado a otra plataforma de manera sencilla y rápida.

Además hay que recordar que estas aceleraciones están transformadas por la ecuación 5.7, de modo que una aceleración en el eje x (norte) puede ser medida por el eje y de la IMU si el quadrotor se encuentra rotado en yaw, con lo que los *offsets* se pueden ir cambiando de ejes durante el vuelo.

El *offset* no es medible directamente, no obstante, se puede estimar si se incluye como variable de estado en el modelo del observador. Se detalla en el punto 5.7.

¹⁷Esto no quiere decir que aumente, simplemente cambia de valor.

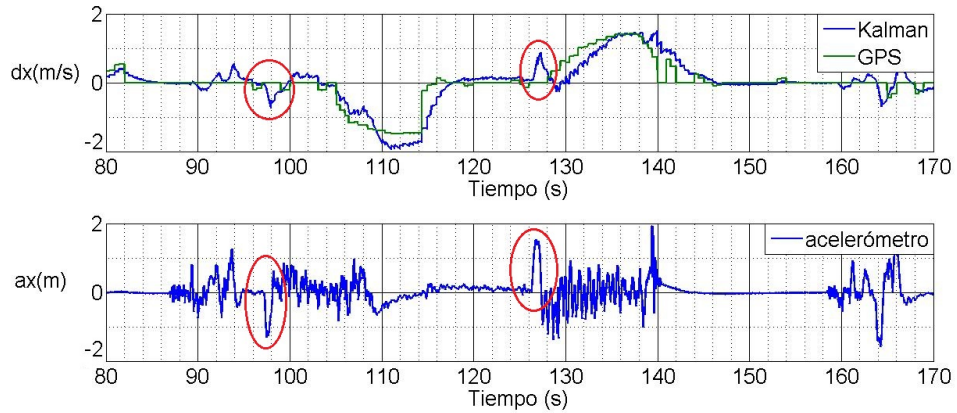


Figura 5.5: Resultados experimentales. Retraso en la señal GPS.

5.6. Retraso en las medidas del GPS. Necesidad de aplicarlo a la IMU.

Otro problema que surgió tras las primeras pruebas fue que parecía que existía un cierto retardo en la señal del GPS. Esto quedó confirmado tras repetir los ensayos en diversas condiciones y lugares. La señal del GPS no comenzaba a variar hasta unos segundos después de iniciar el movimiento.

Este problema es más grave que el del offset, ya que un retardo en la señal medida dificulta enormemente el control. La solución más sencilla consiste en hacer un control menos agresivo bajando las constantes, ya que se corre el riesgo de sufrir fuertes oscilaciones, o incluso inestabilidad, si no se toman medidas. Otra solución es aplicar técnicas de control que tratan de adelantar la señal, como el Predictor de Smith.

En cualquier caso, para aplicar el filtro de Kalman correctamente, ambas medidas tienen que evolucionar paralelamente en el tiempo. Así que la solución que se tomó fue la de determinar cuánto retardo existía y retrasar la señal del acelerómetro de tal forma que se igualara con la del GPS. Haciendo esto parece que en lugar de solucionar el problema, que sería tratar de adelantar la medida del GPS, lo estamos empeorando ya que estamos metiendo intencionadamente un retraso en la medida de la IMU. Sin embargo esto es únicamente para que las dos señales discurren paralelamente al aplicar el algoritmo de Kalman, después se pueden aplicar técnicas de predicción para tratar de adelantarla.

En la figura 5.5 se han señalado los picos de aceleración que marcan el inicio del movimiento real. Estos fueron medidos por la IMU. En la gráfica superior de velocidad se ve cómo la velocidad determinada por el Kalman comienza a subir como consecuencia de la aceleración. Sin embargo la del GPS comienza a variar

unos instantes más tarde.

Tras realizar numerosos ensayos se vio que el retardo no es constante, si no que varía entre 2-5 segundos. La mayor parte de las veces se vio que está en torno a 3.5s por lo que se aplicó un retraso constante de 3.5 segundos a la medida de la IMU. Con esto los resultados mejoraron notablemente.

5.7. Aplicación del algoritmo de Kalman.

En este punto se explica cuál ha sido el modelo empleado para el filtro de Kalman.

El modelo no es más que un modelo cinemático. En ese caso lo describimos para el eje x , aunque es exactamente igual para el eje y . Si miramos la figura 5.2, nuestra u_k sería la aceleración a_{xg} . Y la y_k serían las medidas dadas por el GPS, es decir:

$$y_k = \begin{pmatrix} x_{gps} \\ \dot{x}_{gps} \end{pmatrix}$$

Se le debe aplicar un retraso a a_{xg} de 3.5s tal y como se ha comentado previamente, además es necesario incluir el offset, δa_{xg} , en el modelo. En este punto y por simplicidad a_{xg} ya hará referencia a la aceleración retrasada, no a la actual. El modelo que relaciona la entrada a_{xg} con la salida se puede obtener de forma sencilla.

Por un lado, ya que se ha de incluir δa_{xg} :

$$u(k) = a_{xg}(k) - \delta a_{xg}(k)$$

Por la propia definición de la velocidad se tiene que:

$$\dot{x}(t) = \int_{t_0}^t (a_{xg}(\tau) - \delta a_{xg}(\tau)) d\tau$$

Esta ecuación continua se puede transformar en una ecuación en diferencias. Para ello basta conocer que la IMU actualiza su medida de forma discreta a 300Hz, es decir, que durante h segundos, con $h = 1/300$, la medida es constante. Si escogemos el intervalo de integración de tal forma que $t = t_0 + h$, las aceleraciones se mantendrán constantes en todo el intervalo, por lo que:

$$\dot{x}(t_0 + h) = \dot{x}(t_0) + h \cdot (a_{xg}(t_0) - \delta a_{xg}(t_0))$$

Dicho de otra forma, la velocidad en el instante $k + 1$ depende de la velocidad en el instante k y de la aceleración medida por la IMU en el instante k :

$$\dot{x}(k + 1) = \dot{x}(k) + h \cdot a_{xg}(k) - h \cdot \delta a_{xg}(k) \quad (5.8)$$

Para la posición se tiene:

$$x(t) = \int_{t_0}^t \int_{t_0}^t (a_{xg}(\tau) - \delta a_{xg}(\tau)) d\tau$$

Si se aplica el mismo razonamiento se concluye:

$$x(k + 1) = x(k) + h \cdot \dot{x}(k) + \frac{h^2}{2} \cdot a_{xg}(k) - \frac{h^2}{2} \cdot \delta a_{xg}(k) \quad (5.9)$$

El *offset* se asume que permanecerá constante entre un intervalo y el siguiente:

$$\delta a_{xg}(k + 1) = \delta a_{xg}(k) \quad (5.10)$$

El conjunto de ecuaciones 5.9, 5.8 y 5.10 forman el modelo del sistema. Se pueden expresar en notación matricial de la siguiente manera:

$$\begin{pmatrix} x(k + 1) \\ \dot{x}(k + 1) \\ \delta a_{xg}(k + 1) \end{pmatrix} = \begin{pmatrix} 1 & h & -\frac{h^2}{2} \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x(k) \\ \dot{x}(k) \\ \delta a_{xg}(k) \end{pmatrix} + \begin{pmatrix} \frac{h^2}{2} \\ h \\ 0 \end{pmatrix} \cdot a_{xg}(k) + w_k \quad (5.11)$$

$$\begin{pmatrix} x_{gps}(k) \\ \dot{x}_{gps}(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x(k) \\ \dot{x}(k) \\ \delta a_{xg}(k) \end{pmatrix} + v_k$$

Que vemos que coincide con la representación del modelo 5.3 donde $A = \begin{pmatrix} 1 & h & -\frac{h^2}{2} \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{pmatrix}$; $B = \begin{pmatrix} \frac{h^2}{2} \\ h \\ 0 \end{pmatrix}$; $C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$; $y = \begin{pmatrix} x_{gps}(k) \\ \dot{x}_{gps}(k) \end{pmatrix}$ y $x = \begin{pmatrix} x \\ \dot{x} \\ \delta a_{xg} \end{pmatrix}$.

Algoritmo 3 Algoritmo del filtro de Kalman Extendido.

- Se proyecta el estado: $x_{k+1} = Ax_k + Bu_k$
 - Se proyecta la matriz de covarianza del error: $P_{k+1} = AP_kA^T + Q$
 - if (llega nueva medida del GPS) {
 - Se calcula la ganancia de Kalman: $K = P_{k+1}C^T (CP_{k+1}C^T + R)^{-1}$
 - Se corrige la medida: $x_{k+1} = x_{k+1} + K(y_k - Cx_{k+1})$
 - Se corrige la covarianza del error: $P_{k+1} = (I - KC)P_{k+1}$
- }
-

El GPS actualmente se encuentra dando mediciones cada segundo, por lo que durante las 300 mediciones de aceleración que da la IMU en cada segundo se irá propagando el modelo 5.11 y cuando llegue una medida de GPS se corregirá con la ecuación 5.2:

$$x_{k+1} = Ax_k + Bu_k + K(y_k - Cx_k)$$

Siendo K la matriz de ganancia de Kalman.

En el modelo 5.11 se han incluido los términos $w_k = (w_k^x, w_k^{\dot{x}}, w_k^{\delta a_{xg}})^T$ y $v_k = (v_k^{x_{gps}}, v_k^{\dot{x}_{gps}})^T$ que hacen referencia a ruidos aleatorios. w_k es lo que se conoce como *ruido de planta* e incluye la componente ruidosa de cada variable del sistema ocasionada por un ruido en la entrada. v_k incluye los *ruidos en la medida* del sensor. Las hipótesis respecto a estos ruidos son que son ruidos blancos con media cero y no correlacionados entre sí.

El filtro de Kalman es un observador óptimo de selecciona la matriz de ganancia K de tal forma que da la mejor estima del sistema rechazando al mismo tiempo el efecto de los ruidos w_k y v_k . Para ello es necesario conocer las covarianzas de cada uno de los ruidos, en nuestro caso, como muchas de ellas no son accesibles directamente, se han de ajustar hasta que se obtenga el comportamiento deseado. Se puede mirar de la siguiente manera: si le damos mucha covarianza a, por ejemplo, $v_k^{\dot{x}_{gps}}$, estamos diciendo que la medida de velocidad del GPS es muy mala, de modo que el filtro la tratará con menos importancia. Dichas covarianzas se suelen poner al principio con un valor intuitivo, aunque luego se ha de realizar un ajuste fino.

Se puede encontrar la demostración del filtro de Kalman en numerosos lugares, por ejemplo en [18] se da una. El algoritmo se muestra en el algoritmo 3. Donde

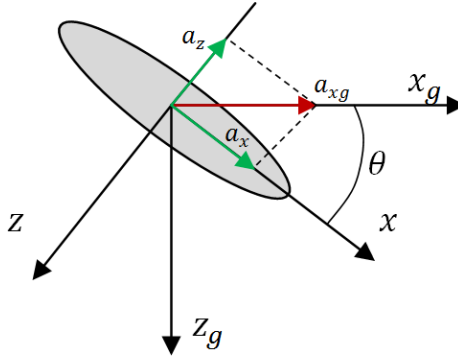


Figura 5.6: Descomposición de una aceleración en dirección norte cuando el quadrotor se encuentra inclinado en pitch.

$Q = \text{diag}(\sigma_x, \sigma_{\dot{x}}, \sigma_{\delta a_{xg}})$ y $R = \text{diag}(\sigma_{x_{gps}}, \sigma_{\dot{x}_{gps}})$ son las matrices estimadas de covarianzas de w_k y v_k , y $P \in \mathcal{M}_{3 \times 3}$ es la matriz de covarianza del error. Generalmente P se inicializa a un múltiplo de la matriz identidad, aunque el algoritmo de Kalman asegura que converge siempre que sea semidefinida positiva¹⁸.

5.8. Comprobaciones realizadas.

Se describen aquí algunas comprobaciones que se tuvieron que hacer durante el desarrollo de este sistema para comprobar que se estaba haciendo de manera correcta.

El primer ensayo que se realizó fue para comprobar el correcto funcionamiento de la ecuación 5.7 aplicada a las medidas de la IMU. El ensayo consistió en orientar al quadrotor en dirección norte y moverlo hacia delante y detrás, primero en el eje x (norte), luego en el eje y (este) y finalmente en z . Después se hizo lo mismo pero esta vez no se situó de manera horizontal, sino que se puso inclinado -45° en pitch, como muestra la figura 5.6, y se repitió el proceso. Se repitió 3 veces más pero inclinandolo $+45^\circ$ en pitch, $+45^\circ$ en roll y, por último, -45° en roll.

El objetivo era ver que cuando el quadrotor se encuentra inclinado en pitch o roll y sufre una aceleración en una dirección cualquiera del sistema de referencia de tierra, por ejemplo a_{xg} , la IMU ve esa aceleración proyectada en dos ejes, que en este caso son a_x y a_z . Sin embargo una vez aplicadas las rotaciones se debería visualizar directamente a_{xg} .

Se ha llamado (a_x, a_y, a_z) a las aceleraciones que lee la IMU, y (a_{xg}, a_{yg}, a_{zg})

¹⁸Se puede decir que una matriz es semidefinida positiva si todos sus autovalores son mayor que cero.

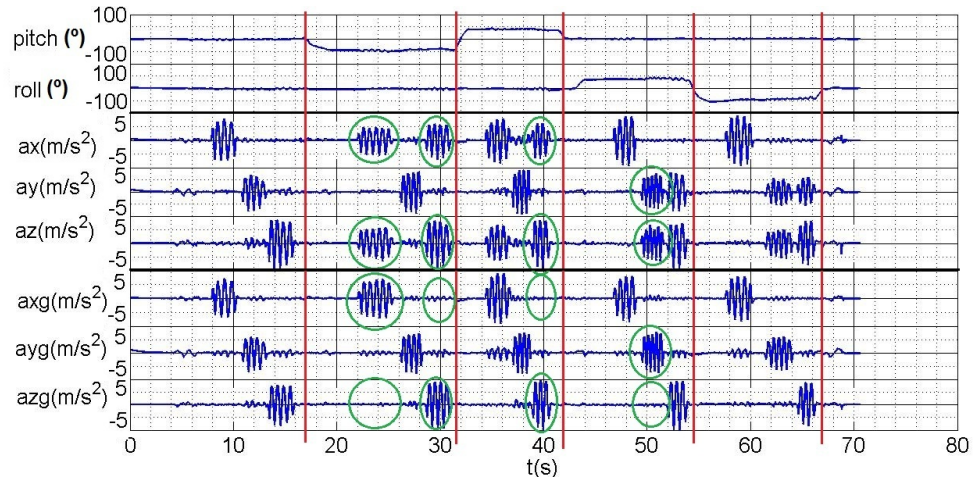


Figura 5.7: Resultados experimentales. Rotación de las aceleraciones dadas por la IMU para expresarlas en el sistema de referencia de tierra. Variación en roll y pitch.

a las aceleraciones obtenidas tras aplicar la ecuación 5.7. Se puede ver en la figura 5.7 que cuando hay pitch o roll la IMU ve dos proyecciones pero, una vez aplicada la rotación, una de ellas desaparece y siempre obtenemos la misma secuencia: primero aceleración en x, luego en y y luego en z independientemente de cuál sea la inclinación. Que fue efectivamente tal cual se llevó a cabo en el ensayo.

El segundo ensayo fue para ver si corregía apropiadamente en yaw. Si, por ejemplo, el quadrotor se encuentra rotado 90° , una aceleración en a_{xg} positiva la IMU la verá como una aceleración en y (figura 5.8). Se repitió el mismo proceso que en el ensayo anterior, solo que esta vez se posicionó el yaw en 0° , 90° , 180° y 270° . Se observa en la figura 5.9 cómo la rotación corrige las aceleraciones de la IMU correctamente.

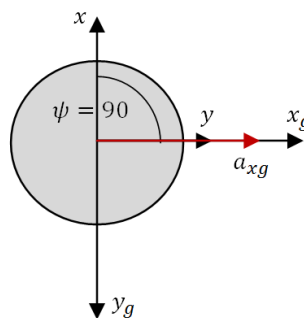


Figura 5.8: Descomposición de una aceleración en dirección norte cuando el quadrotor se encuentra girado en yaw.

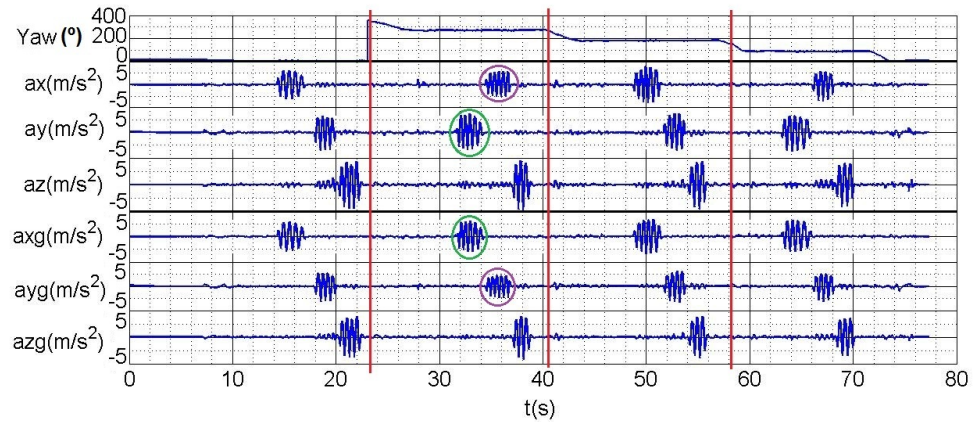


Figura 5.9: Resultados experimentales. Rotación de las aceleraciones dadas por la IMU para expresarlas en el sistema de referencia de tierra. Variación en yaw.

5.9. Resultados experimentales obtenidos.

Se muestran aquí los últimos resultados obtenidos tras programar y hacer funcionar este pequeño sistema GNSS/INS. Los resultados son bastante buenos aun a falta de realizar el ajuste fino de parámetros que se ha de ir conformando poco a poco a medida que se vaya usando. Antes de presentar los resultados hacemos un balance sobre cuáles han sido las ventajas obtenidas así como los resultados menos satisfactorios:

- El incorporar la IMU suaviza notoriamente la medida del GPS que se puede ver que es de carácter escalonado y los bruscos escalones nunca son deseados a la hora de aplicar controladores.
- La IMU detecta cambios de velocidad bruscos y de corta duración que el GPS es incapaz de detectar.
- A pesar de incluir un retraso en las aceleraciones, las actuales siguen disponibles y se conoce además su *offset*, con lo que pueden ser aplicadas para el control.
- El *offset* del acelerómetro se estima de manera casi excelente.
- Se pasa de tener una medida de posición cada segundo, a tener 300 por segundo.
- El problema fundamental es que la precisión final depende del GPS. Por lo que, en gran parte, se sigue estando condicionado por la calidad de este sensor.

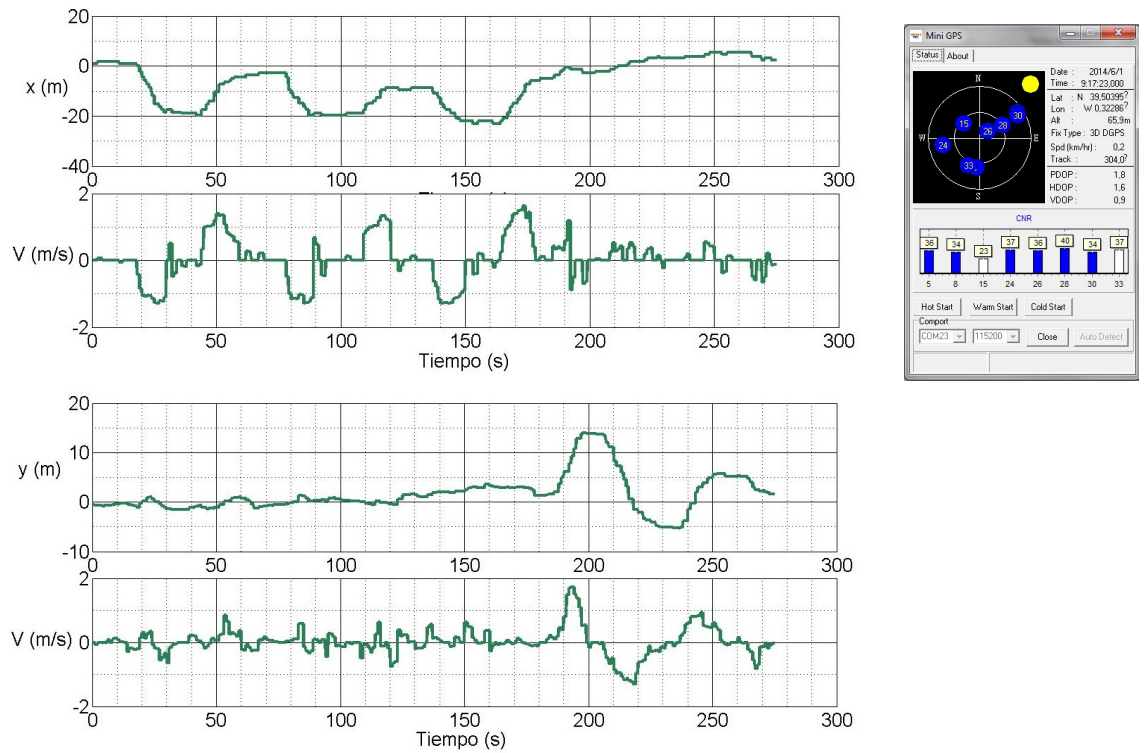


Figura 5.10: Resultados experimentales. Medidas x, dx, y, dy dadas directamente por GPS.

- Se sigue teniendo un retraso de unos 3.5 segundos en la medida.

La figura 5.10 muestra las medidas de posición local obtenidas por el GPS un día con buena captación de señal de satélites. Se adjunta el mapa de satélites en visión que hubo en ese instante. Hay que decir que son condiciones muy óptimas ya que se están captando bastantes satélites además de recibirse correcciones DGPS. Se hicieron movimientos: primero dos en dirección sur y luego uno en dirección este y se guardaron los datos. Se puede ver que la medida de velocidad oscila más que la de posición y pueden llegarse a encontrar picos de error de hasta 1m/s. Para la posición es normal ver errores de 2-3 metros aproximadamente.

La figura 5.11 muestra los resultados de otro ensayo que se realizó. Se hizo un desplazamiento ida-vuelta-ida-vuelta en dirección sur de unos 20 metros y se guardaron los datos. La línea verde corresponde con los datos dados por el GPS directamente, mientras que la línea azul son los datos obtenidos una vez aplicados el EKF. Puede verse como mejora el aspecto de la medida de posición. Los picos que se observan en la medida de velocidad dada por el Kalman son debido a las aceleraciones bruscas, ya que la distancia se recorrió andando, no durante ningún

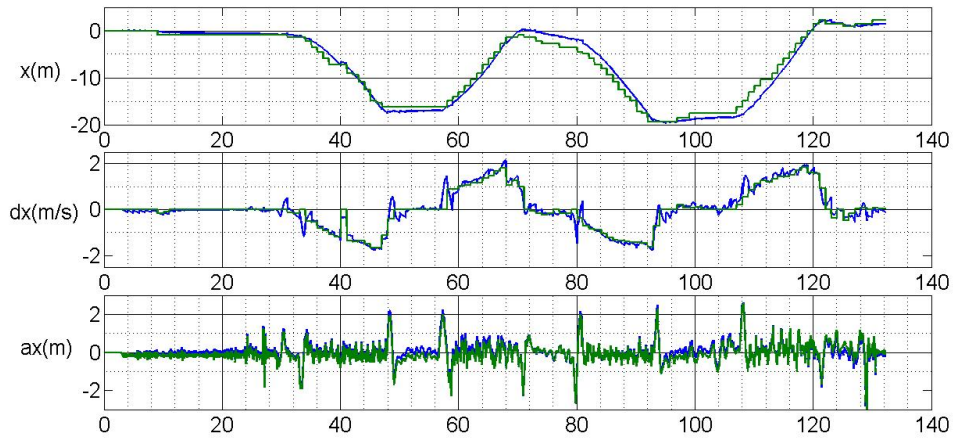


Figura 5.11: Resultados experimentales. Medida de posición tras aplicar el EKF.

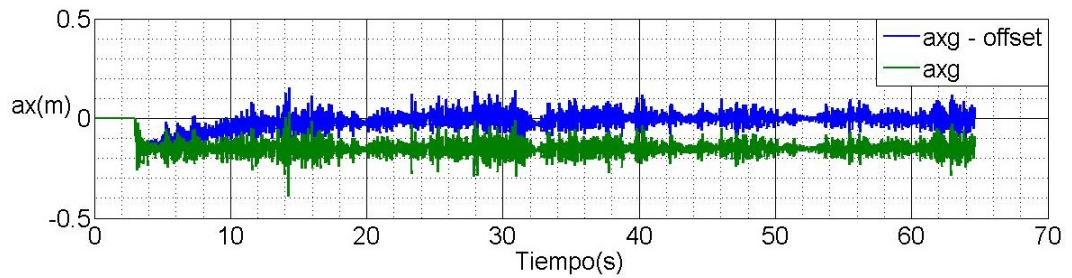


Figura 5.12: Resultados experimentales. Corrección del offset.

vuelo. Es por ello que la aceleración presenta tantas variaciones y éstas se reflejan sobretudo en la velocidad. Hay que mencionar que éstas pequeñas variaciones de velocidad permanecen «ocultas» para el GPS, por lo que son medibles gracias a la incorporación de la IMU.

En la curva de aceleración, la línea verde es la aceleración dada por la IMU directamente. Si a la aceleración de la IMU le restamos el *offset* estimado por el EKF obtenemos la línea azul. La figura 5.12 muestra la corrección del offset en un ensayo en el que se dejó el quadrotor quieto. Puede verse como tras un transitorio queda anulado prácticamente.

5.10. Mejoras y trabajos futuros.

En este proyecto se tenía pensado incluir un predictor de Smith con objeto de intentar eliminar el retardo de 3.5 segundos tanto en posición como en velocidad, no obstante, por falta de tiempo se tuvo que abandonar ése objetivo.

Como trabajo futuro se debería implementar para ver cómo funciona en este caso.

Por otro lado, ya que la precisión final depende básicamente del sensor GPS, se podría ensayar con otros sensores GPS mejores. Por ejemplo alguno que capte satélites tanto GPS, como GLONASS o Beidou, o incluso intentar ver la posibilidad de implementar un sistema GPS diferencial con dos sensores, uno en tierra y otro en el quadrotor.

Por último, este algoritmo no está limitado únicamente a señales GPS, puede adaptarse fácilmente a cualquier otro tipo de señal de posición a cualquier frecuencia. Se puede tratar de implementar con sistemas de posicionamiento de flujo óptico, muy usado en quadrotors, o otros sistemas más avanzados (y caros) como sistemas de posicionamiento por radiofrecuencia.

6. Desarrollo y resultados de la ley de control x-y.

En esta sección se describe el algoritmo de control de posición que se ha desarrollado con el fin de que pueda ser aplicado en la plataforma.

Se ha optado por una estrategia a base de PID's tanto por su sencillez como su robustez. Aunque existen muchas otras estrategias aplicables a quadrotors, como estrategias de control óptimo o seguimiento de trayectorias basadas en control H_∞ .

Es completamente necesario simular antes de implementar el control en el quadrotor y lanzarse a volar, para ello se ha diseñado un pequeño simulador en MatLab al cual le hemos incorporado el modelo estimado del quadrotor. Ahí se puede programar el controlador, simular y visualizar los resultados tanto en forma de gráficas como de forma animada. Se incluye en los anexos las características del simulador, así como las características del modelo empleado.

Para controlar la posición de un quadrotor se actúa a través de las referencias de θ y ϕ . De acuerdo con el modelo simplificado del quadrotor (mostrado en la figura 2.8), ante un $\theta \neq 0^\circ$ ó $\phi \neq 0^\circ$ se genera una aceleración en uno de los dos ejes. De modo que la forma de controlar la posición es controlar éstas aceleraciones que, al fin y al cabo, es controlar los ángulos.

Si queremos controlar la posición global (la referida al marco de referencia de tierra) el problema se complica ya que la dirección en la que se va a dar la aceleración cuando variemos θ o ϕ depende del ángulo yaw, ψ . Es decir, que si $\psi = 0^\circ$ y hacemos $\theta < 0^\circ$ entonces tendremos una aceleración positiva en el eje x. Pero si $\psi = 90^\circ$ y hacemos $\theta < 0^\circ$ entonces tendremos una aceleración en el eje y. Esto se tendrá que tener en cuenta puesto que sino los PID no funcionarán correctamente.

Por último hay que destacar que se han desarrollado dos leyes de control: una sin control de velocidad y otra con control de velocidad. La primera asegurará que el quadrotor va a desplazarse de un punto a otro con un vuelo estable, mientras que la segunda además asegurará que ese desplazamiento se hará a una velocidad dada.

Esta sección de distribuye de la siguiente forma: en el punto 6.1 se analiza el modelo de control de posición, en el 6.2 se asumen dos hipótesis que nos permitirán obtener un modelo apto para desarrollar una estrategia PID. En 6.3 se describe la primera ley, y se comentan tanto sus características como sus desventajas. En 6.4 se describe la segunda ley con control de velocidad. Se verá que si tratamos los ejes x e y como independientes a la hora de hacer control, el quadrotor no se va a mover en línea recta desde un punto (x_0, y_0) a otro (x_1, y_1) a menos que $|x_0 - x_1| = |y_0 - y_1|$. Esto se soluciona en 6.5 estableciendo una pequeña dependencia entre los ejes x e y. En 6.6 y 6.7 se rompe con las dos hipótesis tomadas en 6.2 y de esta forma se tiene ya un control íntegro e «independiente» del resto

de variables del quadrotor. En 6.8 se comenta cómo funciona el algoritmo de gestión de WayPoints 6.8 y, en 6.9 se describe la ley de control x-y final teniendo en cuenta todos los puntos anteriores. Finalmente en 6.10 se muestran los resultados en simulación y se da un ejemplo de la capacidad para seguir trayectorias de este controlador.

6.1. Modelo de control de posición.

El modelo que describe el comportamiento espacial del quadrotor es:

$$\begin{aligned}\ddot{x} &= (-\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi)\frac{F}{m} \\ \ddot{y} &= (-\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)\frac{F}{m} \\ \ddot{z} &= g - \cos\phi\cos\theta\frac{F}{m}\end{aligned}\tag{6.1}$$

Donde se ve que aparecen los ángulos de orientación y donde F es la fuerza resultante dada por los motores (ec. 2.2) y m es la masa del quadrotor.

Este modelo a priori no es muy útil a la hora de aplicarlo, ya que para ello necesitaríamos saber cuánto vale la fuerza F , parámetro desconocido en un principio. Sin embargo se pueden asumir tres hipótesis que nos simplifican notablemente el modelo para transformarlo en un modelo apto para el desarrollo de estrategias de control.

6.2. Modelo de control simplificado.

Las tres hipótesis que se han tomado son:

1. Orientación fija: El quadrotor tiene un control sobre el ángulo ψ que hace que siempre esté mirando al norte. De esta forma $\psi = 0$.
2. Altura constante: El quadrotor tiene un control sobre la altura z que hace que se mantenga a una altura constante. De esta forma $\ddot{z} = 0$ en todo momento.
3. Ángulos pequeños: El quadrotor no se inclinará nunca más de 20° .

Si se cumple la hipótesis 2 la resultante de las fuerzas queda determinada, ya que:

$$0 = g - \cos\phi\cos\theta\frac{F}{m}$$

Y entonces:

$$\frac{F}{m} = \frac{g}{\cos\phi\cos\theta} \quad (6.2)$$

Esto es porque al hacer un control de altura lo que estamos haciendo implícitamente es fijar el valor de F de tal forma que sea justamente la necesaria para sustentar al quadrotor (a pesar de que F no sea medible directamente).

Junto con la hipótesis 1, y tras algunas operaciones de cálculo se tiene que:

$$\begin{aligned} \ddot{x} &= -g \cdot \operatorname{tg}\theta \\ \ddot{y} &= g \cdot \frac{\operatorname{tg}\phi}{\cos\theta} \end{aligned}$$

Si se cumple la hipótesis 3, entonces $1/\cos\theta \simeq 1$, por lo que:¹⁹

$$\begin{aligned} \ddot{x} &= -g \cdot \operatorname{tg}\theta \\ \ddot{y} &= g \cdot \operatorname{tg}\phi \end{aligned} \quad (6.3)$$

Este modelo es el que se ha usado para las simulaciones MatLab y el que, en teoría, marcaría la cinemática del quadrotor siempre y cuando se cumplan las tres hipótesis.

6.3. Ley de control.

La primera ley que se ha implementado es la mostrada en la figura 6.1. Para el eje y es exactamente igual, con las mismas constantes incluso, solo que la salida será la referencia del roll, ϕ_{ref} , y no va multiplicada por -1. Expresada en ecuaciones toman la siguiente forma:

$$\begin{aligned} \theta_{ref}(t) &= - \left(k_p (x_d(t) - x(t)) - k_d \cdot \dot{x}(t) + k_i \cdot \int_0^t (x_d(\tau) - x(\tau)) d\tau \right) \\ \phi_{ref}(t) &= k_p (y_d(t) - y(t)) - k_d \cdot \dot{y}(t) + k_i \cdot \int_0^t (y_d(\tau) - y(\tau)) d\tau \end{aligned} \quad (6.4)$$

¹⁹Efectivamente si $-20^\circ < \theta < 20^\circ$, entonces $1/\cos\theta < 1,065$.

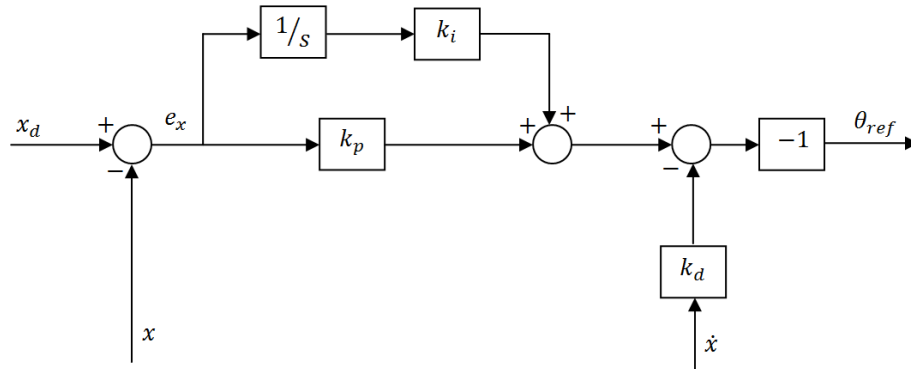


Figura 6.1: Ley de control de posición sobre el eje x.

Se puede observar que no es más que una estrategia PID simple, donde la acción proporcional actúa sobre el error de posición y la constante integral actúa sobre la integral del error. La constante derivativa actúa sobre la derivada de la posición y no sobre la derivada del error. Esto se hace por dos razones:

1. La primera es que la velocidad es una medida que se tiene disponible en la plataforma. Es una medida real que se obtiene de manera «independiente» a la posición (ver sección 5).
2. La segunda es que la posición deseada de x (x_d) va a cambiar de manera escalonada y la derivada de un escalón en el momento del cambio es infinito, por lo que saturaría. Además salvo esos saltos va a ser constante, por lo que:

$$\frac{de_x(t)}{dt} = \frac{dx_d(t)}{dt} - \frac{dx(t)}{dt} = -\frac{dx(t)}{dt} = -\dot{x}$$

Las salidas, θ_{ref} y ϕ_{ref} , antes de ir al algoritmo de control de orientación pasan por una función de saturación, de modo que no se deja que ninguna de ellas supere los $\pm 15^\circ$.

6.4. Ley de control con control de velocidad.

Los motivos que impulsan a plantearse una ley que lleve a cabo también un control sobre la velocidad son fundamentalmente dos: Por un lado cuando nos planteamos que un RPAS se desplace desde un punto a otro, lo que surge de manera natural, es que no únicamente se limite a desplazarse, sino que efectúe el desplazamiento a una velocidad dada. Además, por otro lado, debido al aspecto de la anterior ley y por cómo es el modelo del quadrotor, ante desplazamientos

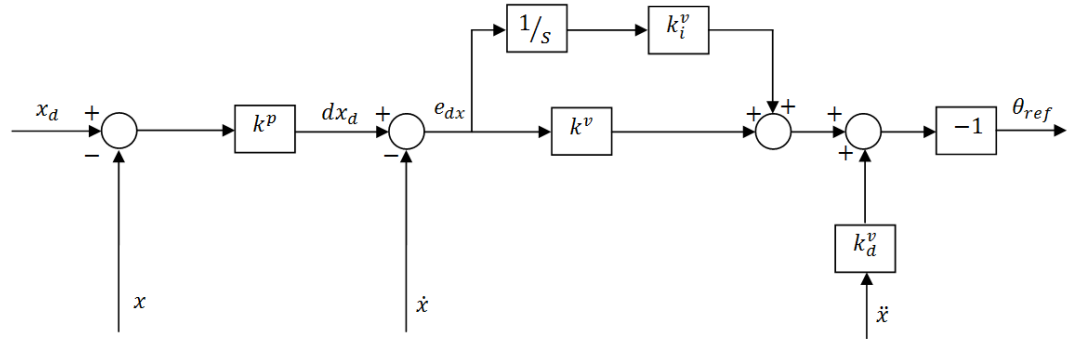


Figura 6.2: Ley de control de posición con control de velocidad sobre el eje x.

largos podría acelerarse más de lo deseado²⁰. De modo que se plantea una realimentación en cascada para llevar a cabo el control de velocidad.

La figura 6.2 muestra el diagrama de bloques de esta segunda ley para el eje x. Al igual que antes, para el eje y es exactamente igual solo que la salida será la referencia de roll y, además, no va multiplicada por -1. Expresada en ecuaciones queda de la siguiente forma.

$$\begin{aligned}
 dx_d(t) &= k^p \cdot (x_d(t) - x(t)) \\
 \theta_{ref}(t) &= - \left(k^v (dx_d(t) - \dot{x}(t)) - k_d^v \cdot \ddot{x}(t) + k_i^v \cdot \int_0^t (dx_d(\tau) - \dot{x}(\tau)) d\tau \right) \\
 dy_d(t) &= k^p \cdot (y_d(t) - y(t)) \\
 \phi_{ref}(t) &= k^v (dy_d(t) - \dot{y}(t)) - k_d^v \cdot \ddot{y}(t) + k_i^v \cdot \int_0^t (dy_d(\tau) - \dot{y}(\tau)) d\tau \quad (6.5)
 \end{aligned}$$

La razón por la que aquí la constante derivativa actúa sobre la derivada de la magnitud y no sobre la derivada del error es porque la medida de aceleración se tiene disponible pero lo más importante es que no se encuentra retrasada en el tiempo como la de velocidad o posición (ver sección 5). Por lo que esta ley, si se aplica con el sensor que tenemos, tiene la ventaja adicional de que rechazará de manera instantáneamente cambios de velocidad.

La ventaja fundamental de hacer este control en cascada es que podemos saturar la velocidad deseada (dx_d) al valor que queramos, al cual llamamos dx_{sat} . De esta forma cuando las distancias sean grandes, la velocidad de referencia quedará

²⁰Ante un error de posición alto, la ley anterior hará que saturé la referencia de roll o pitch, haciendo que se fije y se mantenga en 15° durante gran parte del desplazamiento. Un ángulo de inclinación constante quiere decir una aceleración constante que, dependiendo del aspecto aerodinámico del quadrotor, hará que la velocidad terminal sea mayor o menor.

fijada en dx_{sat} (el control sobre la velocidad ya se encargará de que el quadrotor la siga), pero conforme nos acerquemos al destino ésta referencia de velocidad disminuirá de manera proporcional al error de posición hasta quedarse en cero. La constante k^p marca la dinámica de la velocidad y el control PID actúa sobre las referencias de ángulos de tal forma que la dinámica de velocidad sea la deseada.

6.5. Generalización de la ley de control para los dos ejes.

Las leyes de control descritas arriba actúan sobre cada eje de manera independiente, pero hay que plantearse cómo será su comportamiento cuando actúen sobre los dos ejes simultáneamente.

Con la primera ley no hay ningún problema ya que se actúa directamente sobre los ángulos. Además, como las acciones son proporcionales la dinámica del error en cada eje, el quadrotor se moverá de un punto (x_0, y_0) a otro (x_1, y_1) en línea recta, siempre y cuando no sature ninguno de ellos. Pero lo que importa es que la posición del quadrotor puede ser controlada simplemente controlando cada uno de los ejes de manera independiente.

Para la segunda ley esto no es posible ya que lo que queremos controlar es la velocidad, y no podemos controlar la velocidad del quadrotor si tratamos a cada uno de los ejes de manera independiente. Esto resulta evidente si entendemos que lo que interesa controlar es el módulo de la velocidad del quadrotor y no cada una de sus proyecciones.

Para solucionar el problema simplemente hay que establecer una pequeña dependencia entre los ejes de forma que:

$$\sqrt{dx_{sat}^2 + dy_{sat}^2} = V \quad (6.6)$$

Siendo V el módulo de velocidad deseado.

Se tienen dos incógnitas, pero falta establecer una condición más, que es que también queremos que se desplace en línea recta. Otra forma de decir que el quadrotor va a ir en línea recta, es decir que va a tardar el mismo tiempo en recorrer la distancia $x_d - x$ y la distancia $y_d - y$. Es decir:

$$t_{desplazamiento} = \frac{x_d - x}{dx_{sat}} = \frac{y_d - y}{dy_{sat}} \quad (6.7)$$

De esta forma operando con las ecuaciones 6.6 y 6.7 se llega a que:

$$\begin{aligned}
dx_{sat} &= V \cdot \sqrt{\frac{1}{1 + \left(\frac{y_d - y}{x_d - x}\right)^2}} \\
dy_{sat} &= V \cdot \sqrt{\frac{1}{1 + \left(\frac{x_d - x}{y_d - y}\right)^2}}
\end{aligned} \tag{6.8}$$

Por lo que antes de aplicar la ley de control habrá que definir, a cada iteración de bucle, las saturaciones dx_{sat} y dy_{sat} de ésta forma.

Hay que destacar que en las ecuaciones 6.8 existe una singularidad cuando nos acercamos al destino ya que tanto $x_d - x$ como $y_d - y$ tienden a cero. Podrían existir problemas entonces cuando estemos en las proximidades del destino ya que dx_{sat} o dy_{sat} podrían irse a V o 0 de manera rápida, y el hecho de que cambie de golpe a cero o a V podría afectar al correcto funcionamiento del algoritmo de control. La solución es simplemente no aplicar esas expresiones cuando estemos llegando al punto. En ese caso la velocidad de referencia no saturará, porque si $x_d - x$ y $y_d - y$ son pequeños, entonces:

$$\begin{aligned}
k^p \cdot (x_d - x) &< dx_{sat} \\
k^p \cdot (y_d - y) &< dy_{sat}
\end{aligned} \tag{6.9}$$

Por lo tanto si no saturan, no tiene ningún sentido definir los valores dx_{sat} y dy_{sat} de la manera anterior.

La pregunta que viene ahora es: ¿Y qué distancia se considera como suficientemente cerca del destino para decidir si calculamos, o no, dx_{sat} o dy_{sat} ? La respuesta es que depende tanto de k^p como de V , la forma estrictamente correcta de responder sería sustituir las ecuaciones 6.8 en 6.9 y resolverlas. De esta forma obtendríamos los valores $(x_d - x)$ y $(y_d - y)$ que marcan el límite de cuándo dx_{ref} y dy_{ref} llegan a saturar o no.

En este caso, por simplicidad y como solución inicial, se han fijado 0.5m como límite tanto para x como para y. Aunque como trabajo futuro se plantea el resolver esas dos ecuaciones y programarse en el control.

Hay que destacar que si ningún eje satura, la referencia de velocidad será proporcional al error de posición, por lo que seguirá desplazándose en línea recta²¹,

²¹Esto, realmente, solo es si se cumplen las dos inecuaciones a la vez, sino la trayectoria no será recta.

pero, como es razonable, la velocidad total será menor a V .

6.6. Generalización a tres ejes. ¿Se puede considerar el control de altura independiente?.

Si se quiere tener un modelo un poco más exacto que contemple el efecto de variaciones en altura se puede calcular el término $\frac{F}{m}$ como:

$$\frac{F}{m} = \frac{g - \ddot{z}}{\cos\theta\cos\phi}$$

Donde \ddot{z} es una medida accesible que da la IMU.

En este caso el término $\frac{F}{m}$ aumenta si hay una aceleración hacia arriba. Si aumenta la resultante F y el quadrotor está inclinado aumentarán también las proyecciones sobre los ejes, lo que ocasiona que acelere más ante una misma inclinación.

Sin embargo como se ha dicho antes, los controladores son suficientemente robustos como para rechazar esas variaciones, además que, puestos a ser estrictos, también habría que incluir en el modelo efectos aerodinámicos o de viento que en ningún momento se ha hablado de ellos y son de mayor relevancia que las variaciones de altura. Sin embargo esto no entra dentro del propósito de este proyecto, que es diseñar un controlador para el quadrotor y desarrollar un simulador MatLab con el modelo teórico donde se pueda ver y estudiar la respuesta. Vientos y demás entran al sistema como perturbaciones y se verá la capacidad del control para rechazarlas.

6.7. Rotación de coordenadas para hacer el control independiente del yaw.

Este es sin duda el punto de mayor relevancia, y quizás el más complejo. En el punto 6.6 hemos dicho que la hipótesis 2 se puede asumir cierta siempre, pero ¿qué ocurre con la 1?.

Efectivamente tenemos un control sobre el yaw, de forma que podemos hacer que durante todo el desplazamiento se encuentre en 0 grados, de modo que no hay problema. Pero esto supone añadir una restricción bastante importante al sistema. Una importante aplicación de los quadrotors es la grabación de vídeo, de modo que, ¿qué ocurre si se quiere grabar a 90° a medida que se avanza?, ¿o si queremos que el quadrotor efectúe un círculo mirando siempre al centro?. Se hace necesario hacer independiente el control de ψ del de posición. De esta forma se tendrían completamente separados el control en x, y, z y ψ , aprovechando toda la maniobrabilidad que puede dar un quadrotor.

6.7.1. ¿Porqué únicamente funciona el control para $\psi=0$?

La respuesta es sencilla. Si miramos la ley de control ante un error de posición, por ejemplo, 5m en x al final generará una $\theta_{ref} < 0$. Mirando ahora el modelo 6.3 una $\theta_{ref} < 0$ ocasiona una $\ddot{x} > 0$ que efectivamente contribuye a disminuir el error. Lo mismo pasa con un error positivo en y , que en este caso generará una $\phi_{ref} > 0$ que mirando el modelo ocasiona una $\ddot{y} > 0$ que contribuye también a disminuir el error. Esto se cumple para las dos leyes.

Pero el problema es que el comportamiento del quadrotor solo responde al modelo 6.3 bajo el caso particular de que $\psi = 0$. El comportamiento general viene determinado por 6.1.

Vemos el mismo caso que antes solo que ahora $\psi = 90^\circ$. Ante un error de 5m positivo, al final se genera $\theta_{ref} < 0$, y viendo ahora el modelo 6.1 se ve que ocasiona una $\ddot{y} > 0$. Por tanto el error en x no disminuye y la acción integral ocasionará se aumente todavía más la acción de control θ_{ref} .

6.7.2. Solución al problema.

Tras darle repetidas vueltas al problema se encontró la solución. Explicamos aquí todo el proceso deductivo y finalmente se da el algoritmo de control final en el punto 6.9.

Si asumimos solo las hipótesis 2 y 3 la dinámica de la posición viene dada por el siguiente par de ecuaciones:

$$\ddot{x} = (-tg\theta\cos\psi - \tan\phi\sen\psi) \cdot g$$

$$\ddot{y} = (-tg\theta\sen\psi + \tan\phi\cos\psi) \cdot g$$

Que se puede expresar en forma matricial de esta forma:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sen\psi \\ \sen\psi & \cos\psi \end{pmatrix} \cdot \begin{pmatrix} -tg\theta \cdot g \\ tg\phi \cdot g \end{pmatrix} \quad (6.10)$$

La matriz $\begin{pmatrix} -tg\theta \cdot g \\ tg\phi \cdot g \end{pmatrix}$ no es más que el modelo 6.3 y para el cual son válidas las estrategias de control, a este lo vamos a llamar modelo de posición principal.

La matriz $\begin{pmatrix} \cos\psi & -\sen\psi \\ \sen\psi & \cos\psi \end{pmatrix}$ no es más que una matriz de rotación que transforma la respuesta del modelo principal. Es decir, que ante unas mismas acciones

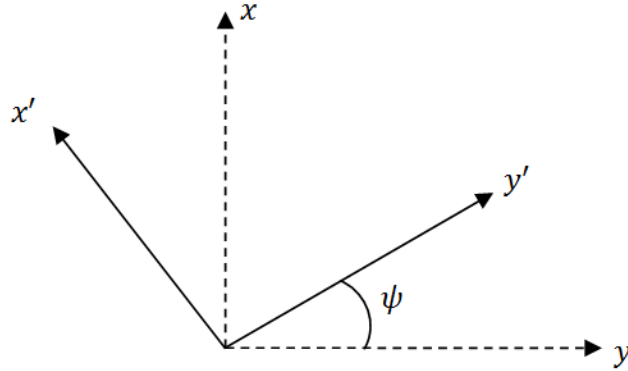


Figura 6.3: Rotación del sistema de referencia de tierra.

de control (θ, ϕ) la respuesta en x o y será diferente debido a la acción de esta matriz. Con lo que vamos a ver si existe algún marco de referencia sobre el que podamos referir las medidas de tal forma que se anule la acción de esta matriz.

Si $R = \begin{pmatrix} \cos\psi & -\text{sen}\psi \\ \text{sen}\psi & \cos\psi \end{pmatrix}$ entonces:

$$R^{-1} \cdot \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} -tg\theta \cdot g \\ tg\phi \cdot g \end{pmatrix}$$

Al ser R una matriz de rotación se cumple que $R^{-1} = R^T$, por lo que se tiene que:

$$\begin{pmatrix} \ddot{x}' \\ \ddot{y}' \end{pmatrix} = \begin{pmatrix} \cos\psi & \text{sen}\psi \\ -\text{sen}\psi & \cos\psi \end{pmatrix} \cdot \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} -tg\theta \cdot g \\ tg\phi \cdot g \end{pmatrix}$$

La matriz $\begin{pmatrix} \cos\psi & \text{sen}\psi \\ -\text{sen}\psi & \cos\psi \end{pmatrix}$ no es más que una matriz que expresa las aceleraciones que sufre el quadrotor en otro sistema de referencia. Y este sistema de referencia nuevo no es más que el sistema de referencia rotado en yaw (Ver punto 2.2). Efectivamente, si miramos la figura 6.3, vemos que la matriz que transforma un vector expresado en el sistema (x, y) al sistema (x', y') es R^T .

Esto quiere decir que en un marco de referencia fijo en tierra pero que rote libremente alrededor de z un ángulo ψ , la respuesta del quadrotor no se encuentra modificada por la matriz R y por lo tanto si expresamos todas las medidas de posición, de velocidad, de aceleración así como las coordenadas de referencia en este marco se podrá aplicar el controlador anterior sin ningún problema.

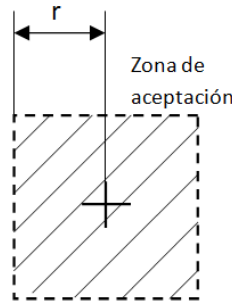


Figura 6.4: Definición de la zona de llegada al WayPoint.

Por ello, antes de ejecutar la ley de control será necesario multiplicar por R^T las coordenadas (x_d, y_d) así como las medidas (x, y) , (\dot{x}, \dot{y}) y (\ddot{x}, \ddot{y}) .

6.8. Gestión de WayPoints.

Una vez se tiene el control de posición completo resulta interesante que el algoritmo no solo sea capaz de ir a unas referencias (x_{ref}, y_{ref}) sino que se le puedan mandar una secuencia ordenada de puntos y que el quadrotor vaya de un punto a otro, siguiendo una trayectoria basada en WayPoints.

Es necesario definir una área de tal forma que se acepte que se ha alcanzado el WayPoint cuando el quadrotor entre a en ella. Esa área se ha definido de forma cuadrada y de lado $2r$, siendo r un parámetro que entra junto a las coordenadas del WayPoint.

Además si estamos empleando la ley con control de velocidad también junto con las coordenadas y r hay que mandar la velocidad V con la que queremos que se desplace al WayPoint, y también se le puede especificar la altura deseada, h , a la que está y el ψ con el que se desplazará, así como el tiempo de permanencia en cada WayPoint.

De esta forma al algoritmo de gestión de WayPoints se le debe pasar un vector de estructuras en la que cada posición del vector incluya una toda la información del WayPoint, que es:

- Coordenada x .
- Coordenada y .
- Velocidad de desplazamiento V (si el controlador controla la velocidad).
- Ángulo ψ con el que se efectuará el desplazamiento.

- Tiempo de permanencia en el WayPoint.
- Radio de aceptación r .

De momento, y por falta de tiempo, únicamente se ha podido desarrollar una versión inicial de éste algoritmo que incluye todos esos parámetros menos el tiempo de permanencia. El quadrotor se limitaría a recorrer de uno en uno todos los WayPoints hasta llegar al último.

Como trabajo futuro se deja el completarlo, ello incluiría la posibilidad de poder cambiar de WayPoint en medio del recorrido (es decir, que cuando esté dirigiéndose al tercero poder decirle que cambie al cuarto, por ejemplo), la posibilidad de anular el recorrido completo, la posibilidad de añadir WayPoints nuevos durante el recorrido, la posibilidad de borrar algunos, etc.

6.9. Algoritmo de control x-y final.

El algoritmo final del control x-y con gestión de WayPoints consiste en un bucle que realiza secuencialmente las siguientes tareas:

En primer lugar lee las medidas actuales de posición, velocidad y aceleración, así como toda el vector con la información de los WayPoints.

Luego, si $X_d = (x_d, y_d)^T$ son las coordenadas del WayPoint al que nos dirigimos y $X = (x, y)^T, \dot{X} = (\dot{x}, \dot{y})^T, \ddot{X} = (\ddot{x}, \ddot{y})^T$ son las medidas actuales de posición, velocidad y aceleración, y $R = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix}$ es la matriz de rotación de coordenadas se hace la siguiente operación:

$$\begin{pmatrix} X'_d \\ X' \\ \dot{X}' \\ \ddot{X}'' \end{pmatrix} = \begin{pmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{pmatrix} \cdot \begin{pmatrix} X_d \\ X \\ \dot{X} \\ \ddot{X} \end{pmatrix} \quad (6.11)$$

Así se expresan todas las medidas en el marco de referencia rotado en yaw.

Tras esta operación se obtienen las saturaciones empleando la expresión 6.8. Ahí V será la definida para ese WayPoint. Además las coordenadas deseadas así como las de posición son las que se encuentran ya rotadas por 6.11.

Después se comprueba si se ha llegado al WayPoint, en ese caso se cambia el destino al siguiente. Si ya se ha llegado al último se marca como siguiente el origen.

Algoritmo 4 Control de trayectorias basadas en WayPoints.

- Lee medidas de posición, velocidad, aceleración y el vector con toda la información de WayPoints.

- Computa:
$$\begin{pmatrix} X'_d \\ X' \\ \dot{X}' \\ \ddot{X}'' \end{pmatrix} = \begin{pmatrix} R & 0 & 0 & 0 \\ 0 & R & 0 & 0 \\ 0 & 0 & R & 0 \\ 0 & 0 & 0 & R \end{pmatrix} \cdot \begin{pmatrix} X_d \\ X \\ \dot{X} \\ \ddot{X} \end{pmatrix}$$

- if (Estamos suficientemente lejos del destino) {
 - Computa: $dx_{sat} = V \cdot \sqrt{\frac{1}{1 + \left(\frac{y_d - y}{x_d - x}\right)^2}}$; $dy_{sat} = V \cdot \sqrt{\frac{1}{1 + \left(\frac{x_d - x}{y_d - y}\right)^2}}$
 - }
 - else {
 - $dx_{sat} = V$; $dy_{sat} = V$
 - }
 - if (Posición X' está dentro del área definida por X'_d y r y además este WayPoint no es el último) {
 - Se actualiza WayPoint.
 - }
 - if (Posición X' está dentro del área definida por X'_d y r y además este WayPoint si que es el último) {
 - Actualiza WayPoint siguiente como el (0,0).
 - }
 - Aplica ley de control.
 - Envía referencias al modulo de estabilización.
-

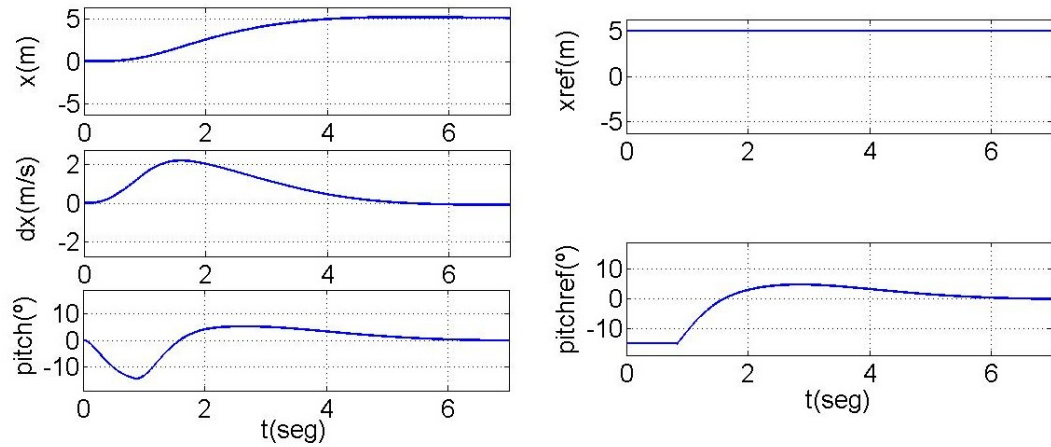


Figura 6.5: Resultados Simulación. Desplazamiento 5m en el eje x sin control de velocidad.

Luego aplica la ley de control que se desee, donde la integral se aproxima por rectángulos conociendo que el hilo se ejecuta en tiempo real a 300Hz.

Finalmente manda las acciones al módulo de estabilización que se encargará de actuar sobre los motores para que se sigan las referencias de roll y pitch.

6.10. Resultados y simulaciones.

Se presentan aquí los resultados obtenidos con el simulador programado en MatLab y que se ha desarrollado específicamente para este proyecto, del cual se detallan sus características en Anexos.

6.10.1. Control de posición de un desplazamiento recto.

La figura 6.5 muestra las gráficas con la respuesta del modelo del quadrotor sobre el que ha actuado la primera ley, que no tiene control de velocidad. Se ve la evolución de la posición, x , así como la velocidad, dx , y el pitch, θ . Se muestra también la referencia de pitch generada por el algoritmo. Puede comprobarse que llega a saturar. También se ve la rápida respuesta, ya que llega a alcanzar una velocidad de 2m/s en una distancia de tan solo 5m. Esta es una de las desventajas de no hacer control sobre la velocidad.

La figura 6.6 muestra la respuesta cuando actúa la segunda ley, y a la que se ha fijado una velocidad de saturación de 2m/s. Se le ha establecido una referencia en x de 20m. Puede verse como el quadrotor se estabiliza en 2m/s.

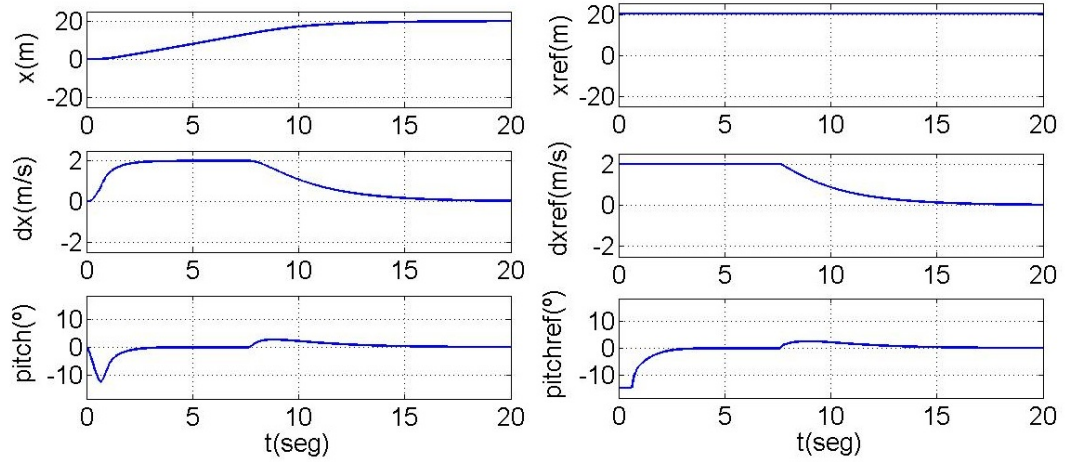


Figura 6.6: Resultados Simulación. Desplazamiento 20m en el eje x con control de velocidad donde se ha fijado en 2m/s.

6.10.2. Control de posición de un desplazamiento recto cuando $\psi \neq 0$.

Las figuras 6.7 y 6.8 muestran el correcto desempeño de la solución planteada para hacer el control independiente del yaw. Pasamos a analizar cada una de ellas.

En la figura 6.7 puede verse la respuesta del quadrotor cuando al encontrarse inicialmente con un yaw de 30 grados se le pide que efectúe un desplazamiento de 20m en x con control de velocidad fijado en 2m/s. Efectivamente satisface las demandas como puede verse en las gráficas $x(m)-t(s)$, $y(m)-t(s)$, $dx(m/s)-t(s)$, $dy(m/s)-t(s)$. Lo interesante, sin embargo, es ver como «se engaña» al algoritmo de control. Se puede observar que, si efectuamos la transformación 6.11 antes de ejecutar el control, se cambian las coordenadas que le introducimos. De esta forma al controlador no le llega $x_{ref} = 20$, sino que le llega $x_{ref} = 20 \cdot \cos(30)$. De la misma forma tampoco le llega $y_{ref} = 0$ sino que le llega $y_{ref} = -20 \cdot \sin(30)$. La velocidad de saturación dx_{sat} y dy_{sat} también cambian de la misma forma que las posiciones de referencia.

Consecuentemente el controlador genera una salida $\theta_{ref}(t)$ y $\phi_{ref}(t)$ que hacen que realmente el quadrotor se desplace en línea recta. Por ello es necesario también transformar las medidas de posición, velocidad y aceleración que le llegan al controlador de tal forma que él «crea» que está ejecutando efectivamente un movimiento tanto en x como en y , y por lo tanto vea que, efectivamente, con esas acciones de control se está posicionando en las referencias que le llegan.

En la figura 6.8 se ha ido un poco más allá. Se ha forzado que el en un instante determinado el quadrotor comience a seguir una ley de yaw $\psi = f(t)$ concreta. En este caso el yaw va girando de forma senoidal entre -90° hasta $+90^\circ$. Es como si el

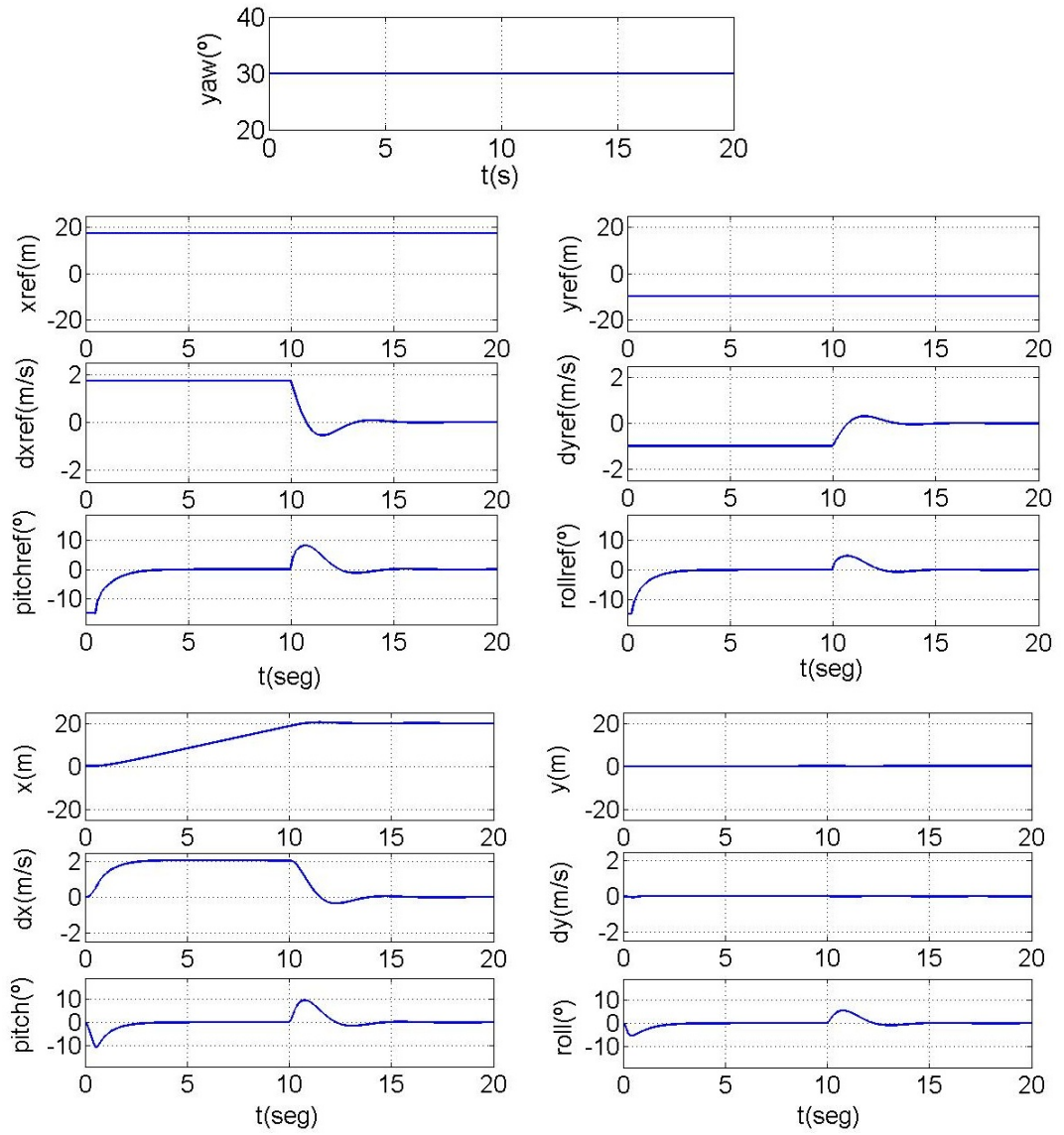


Figura 6.7: Resultados Simulación. Desplazamiento 20m en el eje x con $\psi \neq 0$.

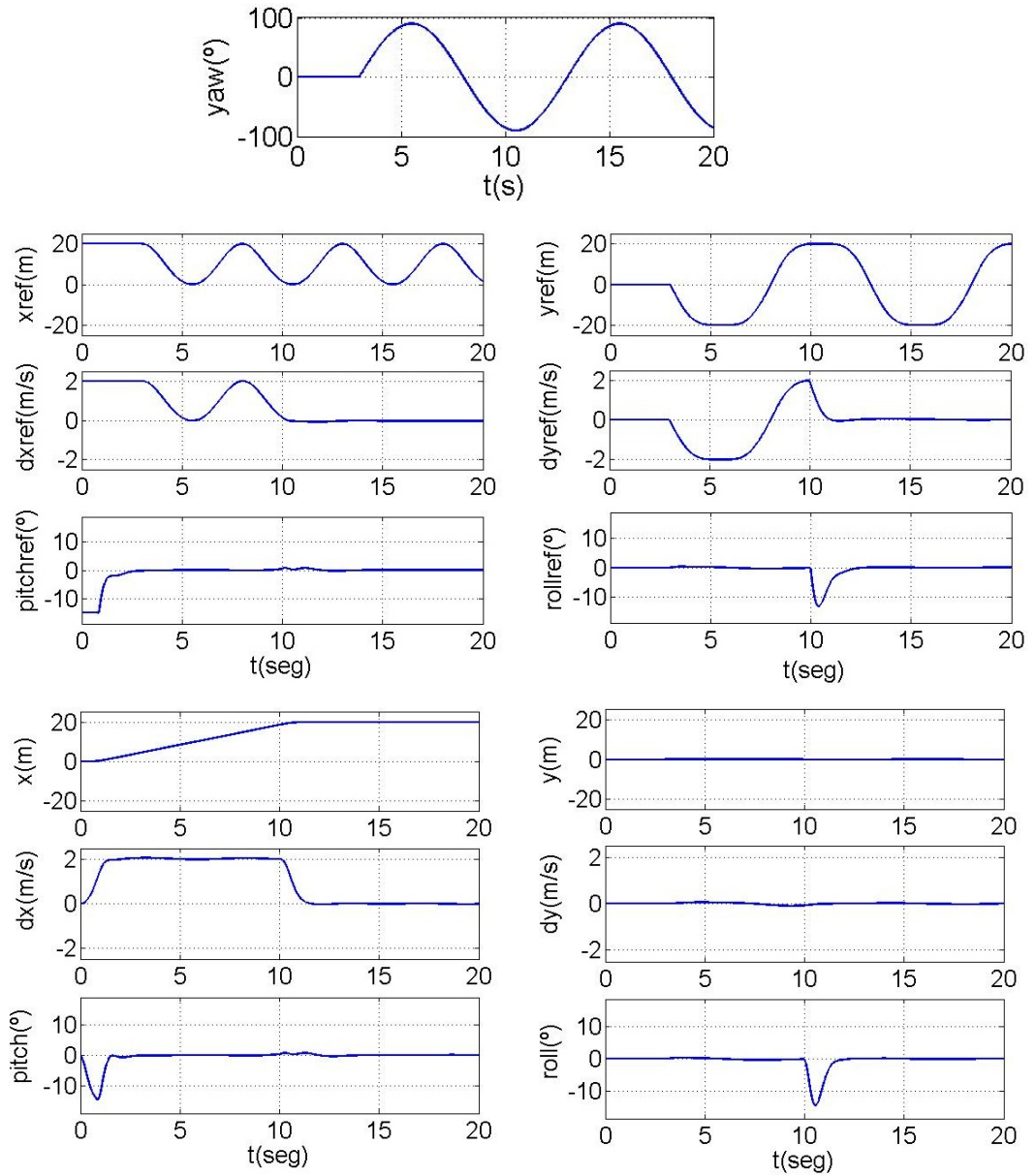


Figura 6.8: Resultados Simulación. Desplazamiento 20m en el eje x con $\psi = f(t)$.

quadrotoz tuviese una cámara a bordo y a medida que avanzase quisiese mirar de lado a lado abarcando 180° y sin interrumpir el desplazamiento. De igual forma puede verse como las referencias de acoplan en cada instante y el resultado final sigue siendo un desplazamiento recto y con la velocidad controlada.

6.10.3. Control de posición de un desplazamiento en diagonal.

En este apartado se justifica el empleo de las ecuaciones 6.8 en la ley con control de velocidad.

En la figura 6.9 se ha simulado un desplazamiento desde el origen al punto (20,5)m. Se ha empleado la ley con control de velocidad pero no se han aplicado las ecuaciones 6.8. En su lugar se ha definido para cada eje la misma velocidad de saturación, que en este caso se ha fijado en 2m/s. Puede observarse cuál será la respuesta del quadrotoz en este caso. Como la distancia en y es más corta que la distancia en x, llega antes a y_{ref} y luego completa el recorrido en línea recta siguiendo al eje x. Puede verse que durante el primer instante el módulo de la velocidad supera los 2m/s, y efectivamente si en cada eje se alcanzan los 2m/s que se ha fijado el módulo puede llegar a 2.83 m/s.

Las líneas verdes en las gráficas $dx_{ref}(m/s)-t(s)$ y $dy_{ref}(m/s)-t(s)$ son (y serán de ahora en adelante) las velocidades de saturación.

En la figura 6.10 se muestran los resultados tras efectuar el mismo desplazamiento, solo que ahora si que se han aplicado las ecuaciones 6.8. Puede comprobarse como cuando está a menos de 0.5m del origen en cada eje dejan de aplicarse las ecuaciones 6.8 para evitar la singularidad existente. No obstante a esas distancias, debido a la proximidad, la velocidad ya no se encuentra saturada, por lo que el control no se ve afectado. Puede comprobarse como el módulo de la velocidad se establece en 2m/s y el trayecto se efectúa además en línea recta.

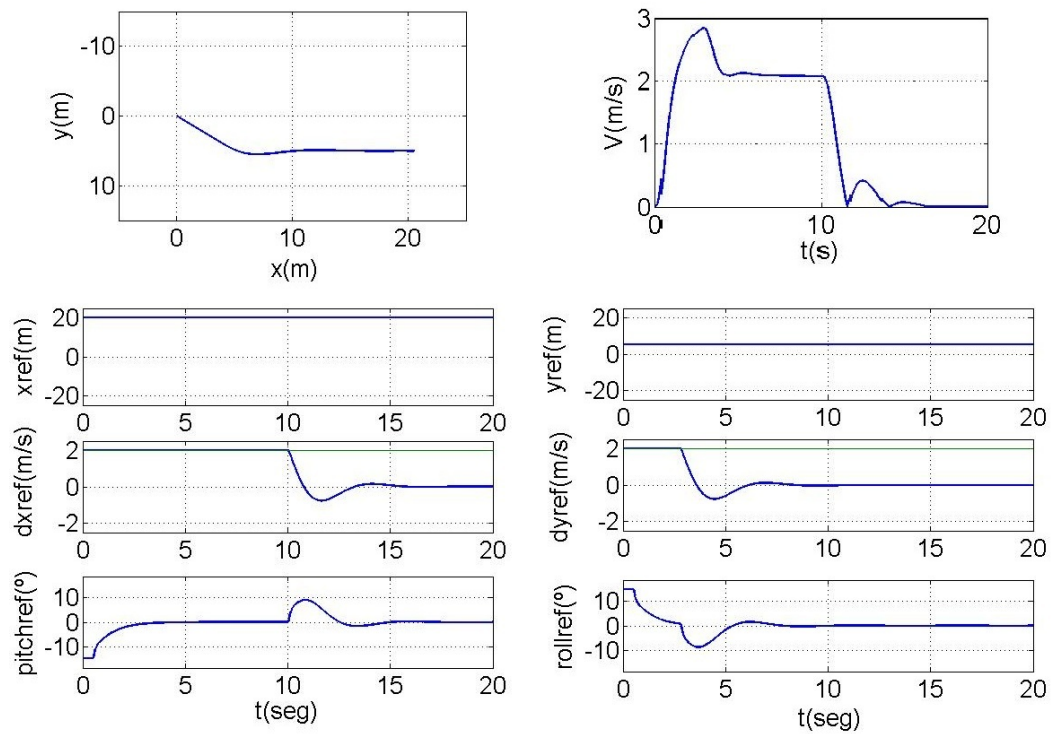


Figura 6.9: Resultados Simulación. Desplazamiento diagonal donde no se han aplicado las ecuaciones que definen las saturaciones.

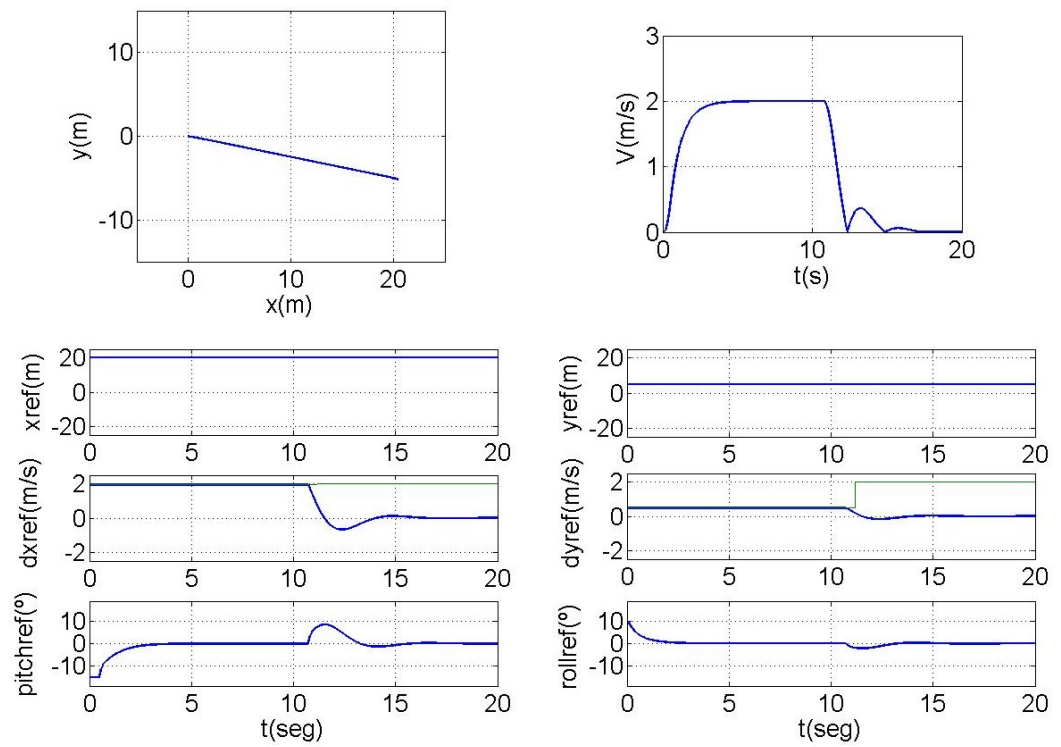


Figura 6.10: Resultados Simulación. Desplazamiento diagonal donde sí que se han aplicado las ecuaciones que definen las saturaciones.

6.10.4. Rechazo de Perturbaciones. Cambio brusco de velocidad durante el desplazamiento.

Es interesante ver cómo va a responder el sistema ante una perturbación. La respuesta ante un cambio brusco en la medida de posición va a ser igual que ante un cambio de referencia, puede verse la evolución dinámica en cualquiera de las figuras de los puntos anteriores.

Nos centramos aquí en una perturbación que consiste en un cambio muy brusco de velocidad en un instante. Además la perturbación se dará en el eje contrario al desplazamiento, es decir, que el quadrotor se mueve en el eje x pero en un instante se da un cambio de velocidad en el eje y.

Se puede comprobar que una vez se rechaza la perturbación el quadrotor no trata de retomar su trayectoria inicial, sino que una vez queda rechazada, desde su nueva posición se dirige al objetivo en línea recta. Se puede observar también como las saturaciones dx_{sat} y dy_{sat} van cambiando y se amoldan a la nueva posición del quadrotor. Todo esto se traduce a que, tras el rechazo de la perturbación, la velocidad en cada eje habrá cambiado, pero el módulo seguirá fijo en el valor deseado.

En este caso se ha establecido la velocidad de referencia $V = 2\text{ m/s}$ y se da el cambio de velocidad en el instante $t = 6\text{ s}$. Se han dibujado de rojo las líneas a las que el quadrotor va a seguir en su recorrido.

6.10.5. Rechazo de Perturbaciones. Viento.

Uno de los aspectos más críticos en quadrotors es la capacidad del controlador para rechazar el viento. Como puede verse, el modelo no incluye esta fuerza que, al ser de carácter externo, se considerará como una perturbación.

En el modelo se ha asumido una velocidad de viento constante e independiente de la velocidad del quadrotor. No se ha hecho para obtener una respuesta cuantitativa (un tema bastante complejo de tratar y en el que intervienen muchas variables) sino que se ha hecho únicamente para obtener una respuesta cualitativa de la capacidad del controlador para rechazar este tipo de perturbaciones. Por ello no se ha incluido el término $k\dot{x}^2$ ²². Únicamente se ha supuesto una fuerza de viento F constante actuando sobre la estructura en una dirección determinada.

En la figura 6.12 muestra los resultados. El quadrotor sigue efectuando el desplazamiento deseado de 20m en dirección x y a la velocidad deseada. Pero hay que ver como la referencia de pitch no se establece en cero en ningún momento, ni siquiera cuando ha llegado al destino y está parado. Siempre se queda inclinado

²²donde \dot{x} es la velocidad del quadrotor y k una constante que depende de numerosos parámetros geométricos y superficiales.

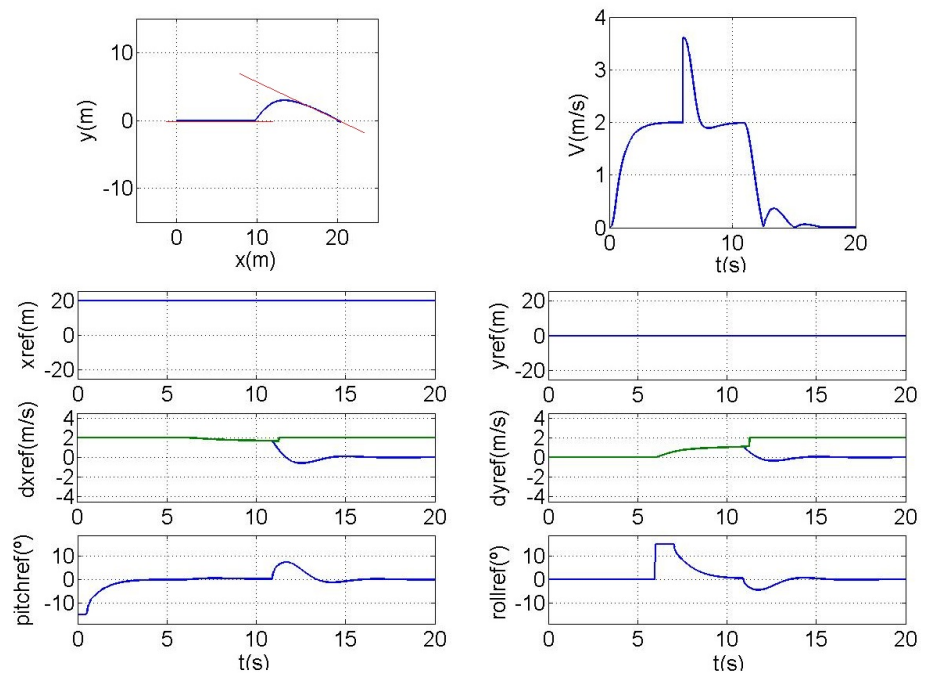


Figura 6.11: Resultados Simulación. Rechazo de un cambio muy brusco de velocidad.

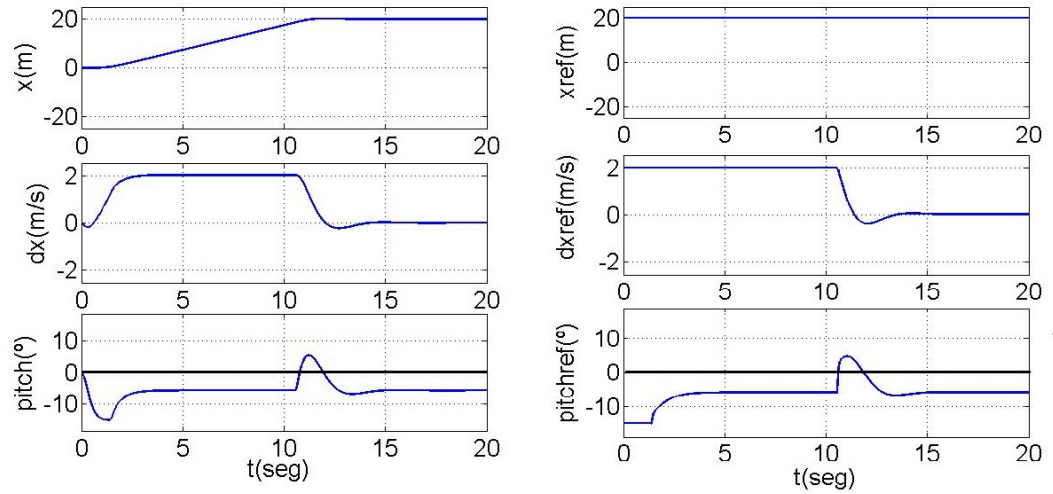


Figura 6.12: Resultados Simulación. Rechazo del viento.

con el ángulo justo que ocasiona una fuerza de avance igual a la del viento.

6.10.6. Ejemplo de control de trayectorias basadas en el algoritmo de seguimiento de WayPoints. Avance recto ejecutando diagonales.

Una vez llegados a este punto, y con el controlador programado y ajustado, hacer que el quadrotor siga trayectorias basadas en WayPoints es algo trivial. Como se ha comentado antes por falta de tiempo no se pudo pulir el algoritmo de control de WayPoints todo lo que se quiso y nos hemos visto forzados a dejar implementada una versión inicial en el que el quadrotor únicamente se limita a ir de un WayPoint a otro hasta completarlos todos, y al final vuelve al origen. Pero sería interesante dotar al programa de cierta maniobrabilidad e incluir opciones como que se pudiese abortar la trayectoria, o que se pudiesen añadir o borrar WayPoints durante el recorrido.

En la figura 6.13 se le han insertado 4 WayPoints al algoritmo de control con un margen de aceptación de 2m de tal forma que el quadrotor avance ejecutando diagonales. Puede verse como van cambiando las referencias que entran al controlador conforme va llegando a los destinos.

6.10.7. Capacidad para aproximar trayectorias más complejas. Seno + semicírculo.

Teniendo disponible el algoritmo de seguimiento de WayPoints puede irse un paso más allá, que consiste en crearse un programa que tome una función ma-

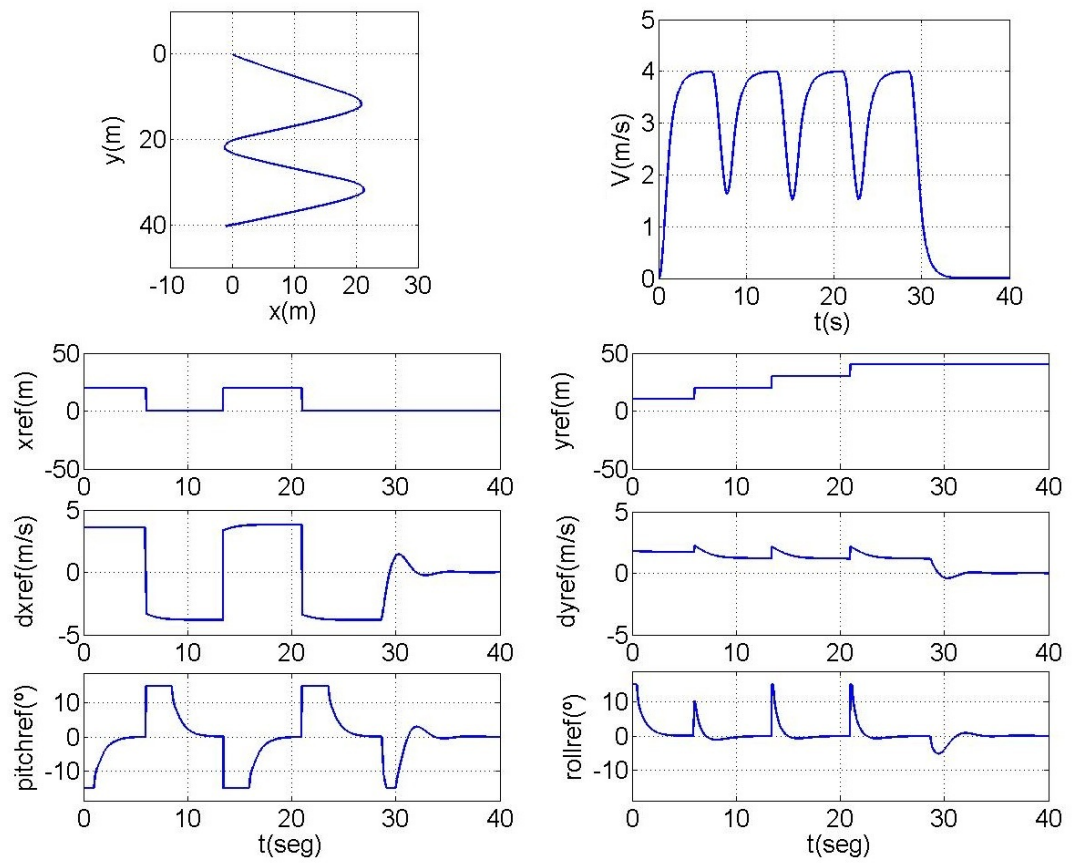


Figura 6.13: Resultados Simulación. Trayectoria, el quadrotor avanza siguiendo diagonales.

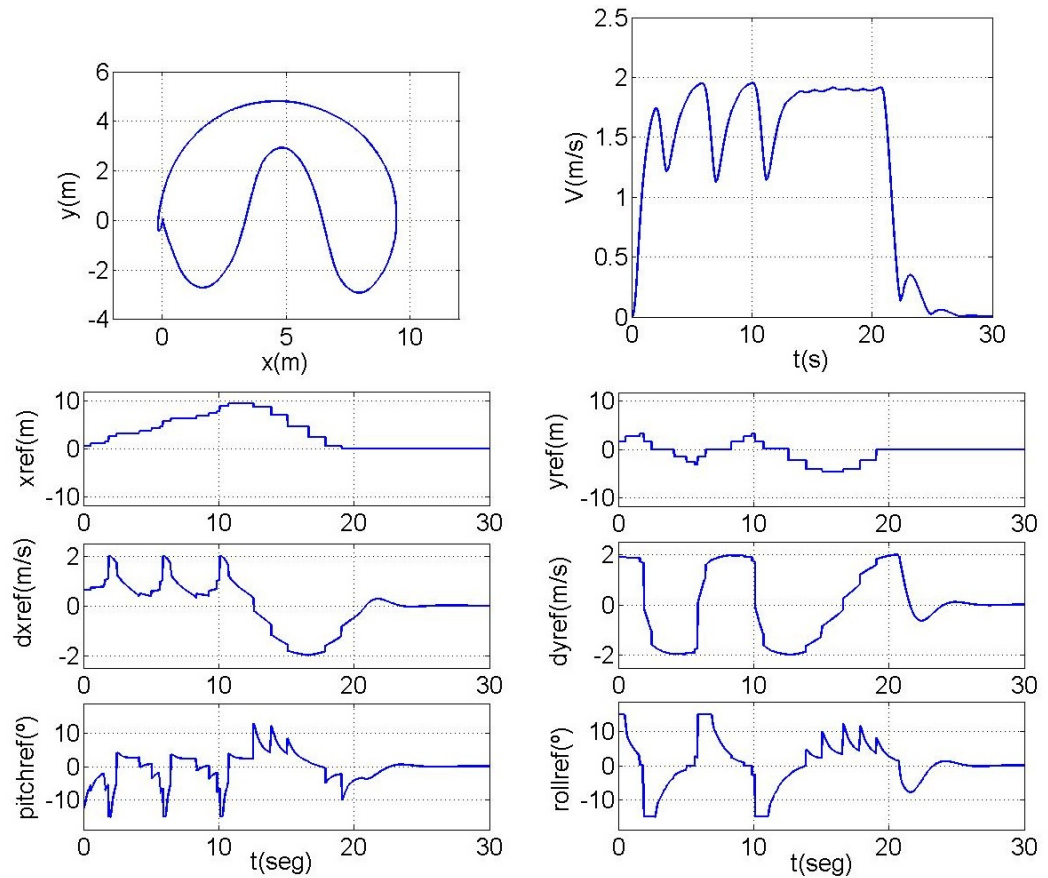


Figura 6.14: Resultados Simulación. Trayectoria compleja, el quadrotor sigue una trayectoria aproximada de un seno más un semicírculo.

temática continua (que representa la trayectoria deseada que queremos que el quadrotor ejecute) y extraiga de ella las coordenadas (x,y) de un número discreto de puntos. De esta forma el quadrotor seguirá la trayectoria de manera aproximada.

En este ejemplo se han extraído 18 WayPoints uniformemente repartidos de una trayectoria continua en forma de seno y después se ha prorrogado con 6 WayPoints más que se han extraído de la ecuación de una trayectoria en forma de semicírculo que le haría volver al origen. El resultado puede verse en la figura 6.14.

En estos casos ya resulta muy complejo ver cuál es la lógica con la que se generan las velocidades y ángulos de referencia, pero ése es el objetivo principal de cualquier ingeniero, que es hacer sencillo algo que de fondo no lo es. El usuario en este caso simplemente se preocupará de cargar los WayPoints y el control hará to-

do el resto. Además esta trayectoria podría efectuarse de igual forma con cualquier ψ , incluso con un ψ que varíe durante el recorrido. Se ve la enorme maniobrabilidad que se le puede extraer a un simple controlador en cascada formado por PIDs.

Durante todo el recorrido la velocidad de referencia ha sido 2m/s. Se comprueba que a pesar de que no se haya podido alcanzar por ser un recorrido que se lleva a cabo en un espacio reducido, nunca se sobrepasa, lo cual es también muy importante puesto que nos dice que a pesar de estar cambiando de WayPoints constantemente siempre se respeta el límite. Alcanzarlo, o no, dependerá del espacio disponible para acelerar.

6.11. Resultados experimentales. Simulaciones en bucle abierto.

Una vez se vio que la ley funcionaba bien en simulación, se procedió a programarla en la plataforma y ensayar la respuesta en bucle abierto.

El comportamiento en bucle abierto es difícil de evaluar directamente ya que el quadrotor no está aplicando las acciones que le dicta el control, sino que dichas acciones simplemente se guardan en un fichero para después observar que están dentro de lo razonable y que, por tanto, el control funciona bien.

Generalmente, antes de cerrar el bucle y como medida de seguridad, siempre es recomendable hacer unos cuantos ensayos en bucle abierto para ver que todo responde como es debido. Después se procede a cerrar el bucle.

A pesar de que, tras los ensayos, la respuesta en bucle abierto fue correcta, no se llegó a cerrar el bucle por motivos de seguridad. El problema fundamental que comprometía la seguridad del quadrotor era que los motores estaban algo infradimensionados y no daban suficiente potencia como para llegar a rechazar el viento. Eso provocaba que supusiera un riesgo lanzar el piloto automático. No obstante, tras los resultados en simulación y los resultados en bucle abierto casi se puede afirmar que esta ley funcionará correctamente.

Se muestran aquí dos de los ensayos hechos en bucle abierto.

6.11.1. Ensayo bucle abierto 1: Con el quadrotor en parado.

El primer ensayo que se hizo fue con el quadrotor en parado en la posición de referencia. Efectivamente si el quadrotor se encuentra en esta posición las referencias deberían ser nulas.

En la figura 6.15 se muestran los datos recogidos para uno de los dos ejes ya que el comportamiento en el otro eje es idéntico. Se muestra en la primera gráfica la posición de referencia junto con la posición real. El error de posición genera que la velocidad de referencia sea diferente de cero. En la segunda gráfica se muestra

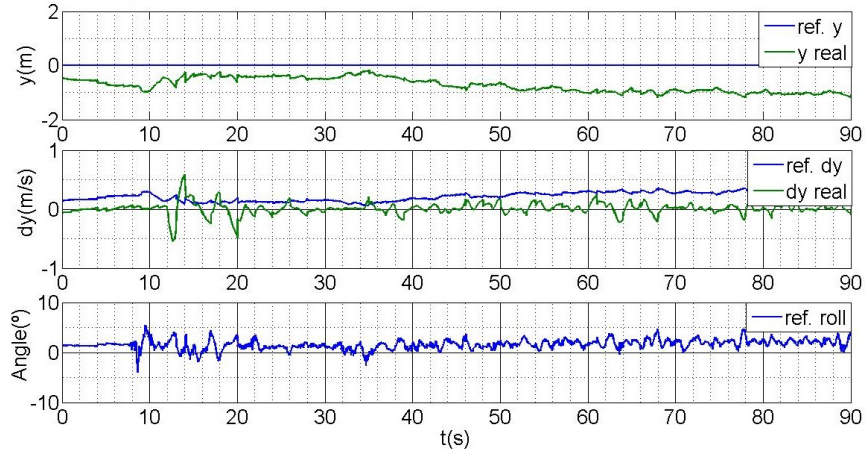


Figura 6.15: Resultados experimentales. Simulación ley de control x-y en bucle abierto.

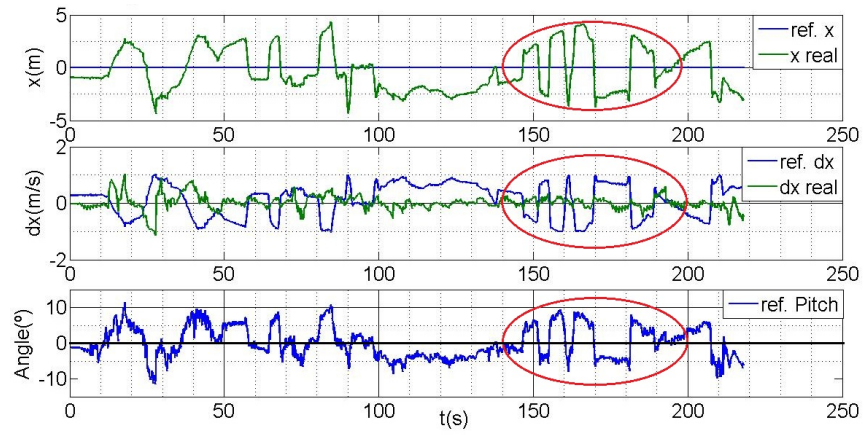


Figura 6.16: Resultados experimentales. Simulación de la ley x-y en bucle abierto. Cambios en yaw.

la velocidad real y la velocidad de referencia. En la tercera gráfica se muestra el ángulo de referencia que se le mandaría al algoritmo de control.

Este ensayo sirvió tanto para verificar que los signos estaban como tocan, como para comprobar la estabilidad de la ley.

6.11.2. Ensayo bucle abierto 2: Oscilando en torno a la posición de referencia y cambiando el yaw.

En este segundo ensayo se quiso comprobar las rotaciones de coordenadas para que el control fuese independiente del yaw, y de paso, ver que ante errores de posición las acciones generadas eran acordes con la ley.

En la figura 6.16 se ven éstos resultados. En las zonas señaladas se fue girando el quadrotor 180° . Se puede ver como las medida x así como las referencias de velocidad y de pitch cambian de signo. Efectivamente si giramos el quadrotor 180° la referencia de pitch deberá cambiar de signo para que el desplazamiento siga ejecutándose en la misma dirección.

6.12. Conclusiones y trabajos futuros.

Como resumen, decir solo que se han implementado dos leyes de control basadas en PID's, una sin control de velocidad y otra con control de velocidad. Se ha visto que la ley que no tiene control de velocidad presenta mejor respuesta en distancias cortas ($\pm 5 m$) mientras la otra funciona mejor en distancias más largas.

La ley que no tiene control de velocidad es una ley de control x-y común y muy citada en la literatura, sin embargo la ley con control de velocidad se ha diseñado exclusivamente para este proyecto. La forma en la que se definen las velocidades de saturación, así como la rotación de medidas para hacer el control independiente del yaw también se ha desarrollado para este proyecto.

Como trabajo futuro se propone acabar el algoritmo de WayPoints para que cumpla todas las características que se mencionaron en 6.8. También es necesario seguir trabajando en obtener una medida de posición muy exacta y fiable de tal forma que pueda ensayarse éste controlador en una plataforma real sin correr ningún riesgo. Para ello el siguiente paso es prescindir de los sensores GPS comerciales que se limitan a dar los datos de latitud y longitud con el protocolo NMEA para pasar a manejar sensores que den los datos directos de los satélites, sin procesar. Y así poder tratar la señal bruta y poder construirse un sistema GNSS/INS con total libertad.

Se propone también poner unos motores más potentes, que sean capaces de sustentar el quadrotor al 50-60 % de su capacidad de empuje. De esta forma ya se estaría en condiciones de cerrar el bucle.

Por otro lado hay que destacar que esta ley tampoco tiene porque funcionar solo con GPS, podría emplearse cualquier otro sistema de posicionamiento como flujo óptico o posicionamiento por radio. Lo único que hay que tener en cuenta es que las posiciones x-y se tienen que referir a un marco de referencia global (que no necesariamente tiene que ser x-norte, y-este).

Referencias

- [1] (Página web) Comisión Europea, sector de los UAS, <<http://ec.europa.eu/enterprise/sectors/aerospace/uas/>>.
- [2] European RPAS Steering Group, *Roadmap for the integration of civil Remotely-Piloted Aircraft Systems into the European Aviation System*, Junio 2013.
- [3] Pablo Albiol, *Desarrollo integral de un quadrotor: Diseño e implementación de los sistemas de telemetría e interfaz gráfica (HMI) para la monitorización y control mediante entorno Qt*, Universidad Politécnica de Valencia, 2014.
- [4] Daniel Verdú, *Desarrollo integral de un quadrotor: Control de orientación basado en una IMU de bajo coste y control de altura en un sensor barométrico*, Universidad Politécnica de Valencia, 2014.
- [5] Randal W. Beard, *quadrotor Dynamics and Control*. Brigham Young University, 2008.
- [6] Guilherme Vianna Raffo, *Modelado y Control de un Helicóptero quadrotor*. Universidad de Sevilla, 2007.
- [7] Luís Ródenas Lorda, *Plataforma de desarrollo para el control de estabilidad en tiempo real de un vehículo aéreo tipo quadrotor*, Universidad Politécnica de Valencia, 2013.
- [8] (Página web) Especificaciones IMU 10DOF, <<http://www.drotek.fr/shop/en/home/62-imu-10dof-mpu6050-hmc5883-ms5611.html>>.
- [9] (Página web) Especificaciones IMU MPU Inven Sense, <<http://www.invensense.com/mems/gyro/mpu6050.html>>.
- [10] (Página web) Características Ultrasonic Distance Sensor de Parallax, <<http://www.parallax.com/product/28015>>.
- [11] (Página web) Sitio oficial de Arduino, <<http://arduino.cc/>>
- [12] (Página web) Documentación oficial GPS Adafruit, <<http://www.adafruit.com/products/746>>.
- [13] (Página web). Documentación oficial placa Igep, <<https://www.isee.biz/products/igep-processor-boards/igepv2-dm3730>>.
- [14] Geoffrey Blewitt, *Basics of the GPS technique: Observation Equations*, Department of Geomatics, University of Newcastle.
- [15] Klaus Betke, *The NMEA 0183 Protocol*, mayo 2000
- [16] D.Alazard, *Introduction to Kalman Filtering*, Septiembre 2005.

- [17] (Página web) Características Spatial FOG,
<<http://www.advancednavigation.com.au/product/spatial-fog>>.
- [18] Karl J. Amström y Björn Wittenmark, *Computer Controlled Systems: Theory and design*.

Plataforma para la implementación y validación de algoritmos de control de tiempo real en mini-helicópteros de varios rotores

Luis Ródenas
luidelo@aii.upv.es

Ricardo Sanz
rsanz@aii.upv.es

Pablo Albiol
pabalgr@etsii.upv.es

Alberto Castillo
alcastfra@etsii.upv.es

Daniel Verdú
davertor@etsii.upv.es

Pedro García
pggil@isa.upv.es

Instituto de Automática e Informática Industrial. Universidad Politécnica de Valencia

Resumen

En este trabajo se presenta una plataforma para la implementación y validación de algoritmos de control en mini-helicópteros de varios rotores. La plataforma, se compone de una unidad de control terrestre (UCT) y un quadrotor con un sistema empotrado de control. La configuración del sistema empotrado, permite la ejecución de las tareas de control más críticas sobre un Sistema Operativo de Tiempo Real (SOTR). Para validar la plataforma, se han implementado varios algoritmos de control. Los resultados experimentales muestran el buen desempeño del sistema propuesto así como una fácil implementación y ajuste de los algoritmos de control considerados.

Palabras clave: Plataforma de desarrollo para quadrotors, Sistemas de tiempo real, PIDs y control no lineal por saturaciones.

1. Introducción

Un vehículo aéreo remotamente pilotado, “Remotely Piloted Aircraft Systems” (RPAS) es un vehículo aéreo capaz de realizar misiones en modo (semi) autónomo. Los mini-helicópteros con varios motores, son una clase de *RPAS* que han ocupando la atención de los investigadores durante los últimos años [4, 7].

Recientemente, ha empezando a plantearse su uso en múltiples aplicaciones civiles, si bien, cabe destacar que, en España, su uso con fines comerciales está explícitamente prohibido por el Ministerio de Fomento. En cualquier caso, la proyección que se prevé para este sector, y la necesidad de desarrollar nuevas estrategias de control que puedan adaptarse fácil y rápidamente a la futura normativa, justifica el desarrollo de nuevas plataformas para la implementación y validación de nuevos algoritmos de control. La Unión Europea, también considera, en su programa HORIZONTE 2020, los

RPAS, y en particular los helicópteros de varios rotores, un sector estratégico de cara a su uso en aplicaciones civiles.

El desarrollo de un sistema de control para un RPAS no es trivial, debido principalmente a la compleja dinámica inherente a los sistemas aerodinámicos, los cuales son multivariables, normalmente subactuados y no lineales [20, 21]. Equipado, normalmente, con un sistema empotrado de control, este vehículo puede emplear diferentes sensores, como cámaras de video, GPS, láser, detectores infra-rojos, detectores ultra-sonidos, etc., para poder navegar en un ambiente desconocido (ver por ejemplo, [6] [24]). El desarrollo del sistema de control para este tipo de vehículos, no sólo implica un adecuado ajuste de las leyes de control, sino, la ejecución de las mismas de forma segura, y en el menor tiempo de muestreo posible [19]. Igual de importante, es que la tarea de control que implemente el algoritmo de control presente el menor retardo (jitter) posible [3]. Como es sabido, este objetivo se puede cumplir fácilmente realizando la implementación sobre un micro-controlador y/o en un mini-PC con un Sistema Operativo de Tiempo Real (SOTR).

Un problema crucial, cuando se trabaja con este tipo de sistemas, es la necesidad de disponer de una herramienta que nos permita interactuar con el sistema empotrado en la fase de ajuste on-line de los parámetros de control [1, 21, 5].

El trabajo presentado, consiste en el desarrollo de una plataforma genérica para la implementación y validación de un sistema empotrado de control en un quadrotor. La plataforma, esta concebida para permitir una fácil implementación y comparación entre diferentes algoritmos de control.

El resto del artículo está estructurado en las siguientes secciones: en la sección 2 se describe la herramienta desarrollada para la solución adoptada mientras que en la sección 3 se plantea tanto el modelado matemático como las leyes de control básicas del quadrotor. En la sección 4 se presentan

los resultados experimentales, y en la sección 5 las conclusiones y futuros trabajos.

2. Descripción de la plataforma

En esta sección se realiza una descripción de la plataforma desarrollada. Ésta consta de una Unidad de Control Terrestre (UCT) que se utilizará como interfaz (Human Machine Interface, HMI), entre el usuario y el sistema empotrado en el quadrotor. En el UCT se ha desarrollado una herramienta que permite un ajuste on-line de los parámetros de control y facilita el hecho de implementar y comparar diversos algoritmos de control.

Con respecto al sistema empotrado, se ha desarrollado una arquitectura en dos capas. El objetivo es que los algoritmos de control más críticos (orientación) se ejecuten sobre un μC (emulando una tarea crítica de un SOTR), dejando las tareas de mayor carga computacional (control de posición, flujo óptico, etc.) y comunicación con la UCT a un mini-PC.

2.1. Arquitectura del sistema empotrado

Los prototipos construidos se componen de tres elementos, que denominaremos: unidades de control, actuadores y elementos de sensorización (figura 1). Se han realizado varias iteraciones en el diseño de la plataforma (dispositivos HW y SO) hasta conseguir una arquitectura lo más simple y sencilla de configurar pero que, a la vez, ofrezca las mejores prestaciones posibles para realizar el control integral de un quadrotor tanto en interiores como en exteriores.

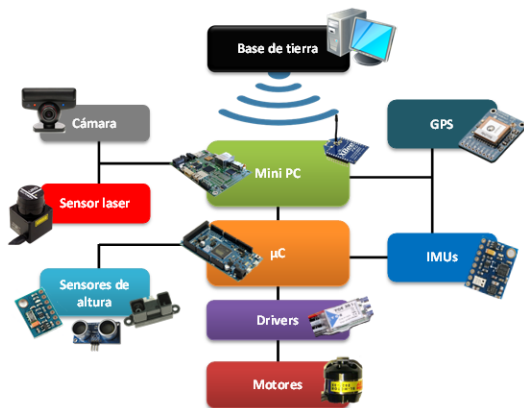


Figura 1: *Sistema empotrado*

Unidades de control

Las unidades de control del sistema empotrado son dos, un mini-PC y un μC . En la versión actual de la plataforma, el mini-PC utilizado es una placa

IGEP V2 [10], mientras que el μC es un Arduino Due [9].

La IGEP V2 está provista de un procesador ARM Cortex A8 a 1 GHz, 512MB SDRAM, 512MB NAND Flash, lector de tarjetas micro-SD, USB 2.0, mini USB OTG 2.0, HDMI, Ethernet 10/100, Wi-Fi, RS-232, RS-485, I2C y SPI. Pero lo más interesante, tal vez, es que cuenta con un DSP (digital signal processing), el cual resulta muy útil a la hora de realizar el tratamiento de imágenes para la implementación del control de posición x-y mediante flujo óptico, por ejemplo, dado que descarga al procesador de todo el algoritmo de tratamiento de imágenes.

Como Sistema Operativo se ha instalado Linux + Xenomai [12]. El Arduino Due cuenta con un microcontrolador de 32 bits ARM Cortex M3 a 84 MHz, acompañado de 54 pines de entrada/salida digitales, 12 pines de entrada analógicos, 2 pines de salida analógicos, así como periféricos que se encargan de las comunicaciones serie, USB OTG, I2C, SPI e incluso bus CAN.

La figura 2 muestra los diferentes protocolos de comunicación utilizados para conectar los componentes del sistema empotrado entre ellos y con la UCT.

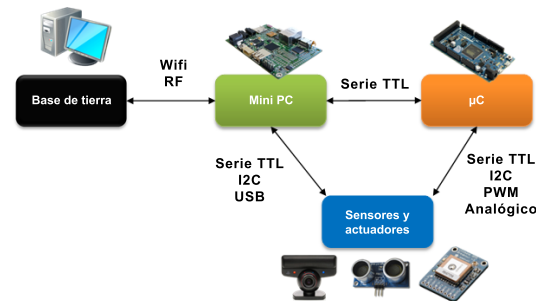


Figura 2: *Arquitectura de las unidades de control*

Actuadores

La plataforma utiliza motores brushless (sin escobillas) de la marca Robbe Roxxy, modelo 2827-35, especialmente pensados para quadrotors. Junto a unas helices 1045 flexibles, cada motor tiene un empuje máximo de unos 600 gramos (5.9 N) y un consumo máximo de 10 amperios.

Para variar la frecuencia de giro de los motores, y en consecuencia la fuerza de empuje que genera cada uno de ellos, se utilizan drivers o ESC (electronic speed controller). Los drivers utilizados en la plataforma son de la marca YGE [11], modelo 25i.

El diseño actual de la plataforma utiliza señales

PWM (pulse-width modulation) para controlar los drivers. Se utiliza un pulso alto entre 1 y 2 ms de ancho, a una frecuencia de entre 200 y 500 Hz, actualmente fijada en 300 Hz. Por lo que esta es la frecuencia máxima de operación de los motores.

Sensorización.

Para poder obtener las distintas variables de estado, se utilizarán diferentes tipos de sensores. Para medir la orientación se utilizan IMUs, equipadas con acelerómetros, giroscopios y magnetómetros, la IMU hace el tratamiento de los datos, proporcionando directamente la medida de los ángulos de Euler (ϕ, θ, ψ) y sus derivadas. Para este proyecto se ha utilizado la IMU *MPU 6050* de InvenSense Inc [13]. El quadrotor incorpora dos IMUs puesto que un fallo en la medida de los ángulos sería una situación crítica y puede llegar a provocar un fallo en la ley de control. En una de ellas los ángulos de Euler los proporciona directamente el procesador interno de la IMU. En la otra, se ha implementado un código que lee directamente las medidas en bruto de los sensores y aplica un filtro de Kalman con las medidas de giroscopios y acelerómetros para obtener el pitch y el roll y otro filtro de Kalman con las medidas del giroscopo y del magnetómetro [17] para obtener el yaw.

Para el control de posición, es necesario obtener una medida rápida y fiable tanto de la posición en el plano xy como de las respectivas velocidades. El GPS presenta el problema que en interiores no es funcional, ya que no hay visibilidad de los satélites. En el caso de interiores se puede utilizar distintos sistemas de posicionamiento, la mayoría se basan en el principio de triangularización (Vicon, kinematics, etc.), si bien también es posible realizar un control de posición a partir de dispositivos de barrido laser (SLAM, [2]), o a partir de algoritmos de visión, como es el caso del control x-y por flujo óptico [22], a partir del cual es posible obtener las variables \dot{x}, \dot{y} , y a partir de ellas, estimar la posición relativa x-y.

Se ha dotado a la plataforma con un sistema de posicionamiento GNSS/INS. Dicho sistema combina las ventajas del *Sistema Global de Navegación por Satélite* (Global Navigation Satellite System: GNSS) con las ventajas de los *Sistemas de Navegación Inerciales* (Inertial Navigation Systems: INS).

Por un lado los sistemas GNSS dan una posición absoluta que no está sometida a deriva, sin embargo son lentos en cuanto a actualizaciones. Por otro lado los sistemas INS detectan muy bien las variaciones rápidas de posición o velocidad, pero su mayor inconveniente es que están sometidos a una deriva temporal ya que obtienen la posición y

velocidad por integración de la aceleración.

Actualmente el sistema GNSS más utilizado es el Sistema de Posicionamiento Global (GPS) de los Estados Unidos. El sensor GPS que se ha incorporado es el *Ultimate GPS Breakout v3* de Adafruit [14]. Se ha comprobado experimentalmente que el error de medida de dicho sensor, en un día despejado, con visibilidad directa al cielo y recibiendo correcciones DGPS está en torno a los 2 m para la posición y a los 0.1 m/s para la velocidad.

El GPS emite la información de posición en formato NMEA, por lo que da medidas de latitud, longitud y de la velocidad en módulo. Las señales de latitud y longitud se han transformado a medidas de posición local x, y (m). También se ha desarrollado un algoritmo que observa las variaciones de x, y y obtiene un coeficiente que pondera la medida del módulo de la velocidad. De esta forma se obtienen también las medidas v_x, v_y . Con esto logramos que el GPS proporcione las 4 coordenadas (x, y, \dot{x}, \dot{y}), necesarias para el correcto desarrollo de cualquier algoritmo de control x-y.

Para completar el sistema GNSS/INS se ha implementado un filtro de Kalman, en el cual se estiman las medidas del GPS (x, \dot{x}, y, \dot{y}) integrando las aceleraciones $\ddot{x}, \ddot{y}, \ddot{z}$ dadas por la IMU, y se corrigen cuando llega una señal GPS útil. El algoritmo de Kalman, como resultado, nos ofrece una estimación de los bias ($\delta a_x, \delta a_y$) de los acelerómetros ya que han sido debidamente incluidos en el modelo.

Para la medición de la altura “z”, los tipos de sensores más usuales son los infrarrojos, ultrasonidos o sensores barométricos. Los primeros se usan para hacer medidas menores de 1,5 metros. Los ultrasonidos proporcionan medidas hasta los 3 o 6 metros dependiendo del tipo del sensor. Su desventaja es su reducido alcance y que dependiendo del tipo de suelo que haya debajo (cesped, una alfombra, etc.) se pueden obtener medidas erróneas.

Por último los sensores barométricos, pueden medir casi cualquier altura, ya que utilizan el cambio de presión en la atmósfera para calcular la altura, este tipo de sensores es apto para exteriores, el inconveniente es que no puede detectar los obstáculos que hay debajo del vehículo y que cualquier ráfaga de viento puede generar una sobrepresión o una depresión que genere una medida errónea de altura. Es por esto que la mejor medida consiste en la integración sensorial de distintos dispositivos.

En este proyecto se ha optado por dotar a la plataforma con un sensor barométrico [16] junto a un sensor de ultrasonidos [15] y fusionar las medidas. Se podría haber realizado el control de altura de la plataforma solo con el sensor barométrico pero se optó por incorporar también un sensor de ultra-

sonidos por dar una medida más precisa y fiable que el barómetro, a cotas bajas, y así facilitar las maniobras de despegue y aterrizaje.

Como sensor de ultrasonidos se utiliza el *Ultrasonic Distance Sensor* de Parallax. El rango de funcionamiento de este sensor es de entre 2 cm y 3 m.

Para el sensor barométrico se ha optado por el modelo *MS5611-01BA03* de Measurements Specialities. Una de las ventajas que presenta este sensor es que incorpora un sensor de temperatura interno para corregir la medida de presión. Este sensor funciona para rangos de presión entre 10 y 1200 mbar y rangos de temperatura entre -45 y 80 °C. es la solución más empleada en quadrotors concebidos para volar en exteriores.

2.2. Comunicación mini PC - UCT e interfaz (HMI)

Además de las unidades de control y sensores presentados anteriormente, resulta imprescindible disponer de un sistema de telemetría entre el mini PC de a bordo y un PC en la unidad de control terrestre (UCT). Este enlace se utiliza tanto para la monitorización de las distintas variables e información proporcionada por los sensores del vehículo, como para el control del mismo.

La plataforma puede comunicarse con la UCT mediante WiFi y radiofrecuencia. Para el enlace WiFi se aprovecha la IGEP V2 que dispone de un chipset dedicado a ello. La transmisión de información se lleva a cabo mediante sockets a través del protocolo TCP. El principal problema es que el alcance está limitado a 10-15 metros y por tanto no sirve para vuelos en exteriores. Para la comunicación por radio se han utilizado dos *Xbee PRO S1* [18] (una conectada por serie al mini PC y la otra directamente conectada a un PC en la UCT). Las Xbee son módulos de radiofrecuencia de bajo coste, bajo consumo y que además permiten crear redes extensibles. La principal ventaja respecto a la comunicación por WiFi es que el alcance puede llegar a 1,5 kilómetros.

La visualización de todas las variables y constantes de interés, además del control del UAV se realiza por medio de una interfaz hombre máquina (HMI). Este software ha sido desarrollado en Qt, una biblioteca multiplataforma para desarrollar aplicaciones en C++ con interfaz gráfica de usuario o sin ella y que además cuenta con IDE propio.

Se trata de una interfaz sencilla que cuenta con una ventana principal donde se pueden visualizar las variables más importantes del vehículo y varios bloques para realizar funciones adicionales. A

grandes rasgos, la herramienta dispone de las siguientes funcionalidades:

Sección central con mapa de Google Maps, instrumentos de vuelo en la parte inferior (horizonte artificial, altímetro, brújula, indicador de velocidad en el plano horizontal e indicador de velocidad vertical). Asimismo, en el bloque Mission Information es posible exportar y visualizar estos parámetros en forma de gráficos. En la parte derecha hay un panel para enviar referencias al vehículo, así como cambiar su estado de vuelo. La interfaz también está preparada para recibir información sobre el estado de vuelo o de la cobertura GPS.

Mención aparte merece el bloque dedicado al ajuste online de las constantes de control (proporcional, derivativa e integral). Notar que, dado que los parámetros físicos del quadrotor pueden variar para diferentes configuraciones, el ajuste de los parámetros de control se debe realizar a partir de la experiencia del ingeniero de control realizando pequeños ensayos hasta alcanzar las prestaciones de control deseadas.

Por último, la interfaz también permite al usuario el control del vehículo a través del teclado o de un joystick. En particular, el offset de velocidad de los motores podrá ser controlado mediante teclado, así como las referencias sobre el roll y pitch.

3. Modelado dinámico y algoritmo de control

3.1. Modelo dinámico

Para el cálculo del modelo matemático se utiliza la aproximación de las ecuaciones de Euler-Lagrange [8]. Las coordenadas generalizadas del helicóptero se presentan como:

$$q = (x, y, z, \psi, \theta, \phi) \quad R^6 \quad (1)$$

donde (x, y, z) denotan la posición del centro de masas del helicóptero, y (ψ, θ, ϕ) describen los tres ángulos de Euler, que representan la orientación del helicóptero. Después de una serie de cálculos y simplificaciones, el modelo matemático se puede representar como [7]:

$$m\ddot{x} = -u \quad \theta \quad (2a)$$

$$m\ddot{y} = u \quad \theta \quad \phi \quad (2b)$$

$$m\ddot{z} = u \quad \theta \quad \phi - mg \quad (2c)$$

$$\ddot{\psi} = u_\psi \quad (2d)$$

$$\ddot{\theta} = u_\theta \quad (2e)$$

$$\ddot{\phi} = u_\phi \quad (2f)$$

donde m es la masa del helicóptero, g es la aceleración gravitacional, y u el empuje total. Las va-



Figura 3: Interface (HMI) para la configuración del control en vuelo libre de un quadrotor.

variables u_ψ , u_θ y u_ϕ se definen como las estradas de control de los ángulos (ψ , θ , ϕ), tal que:

$$u_\psi = k_\psi(V_1 - V_2 + V_3 - V_4) \quad (3a)$$

$$u_\theta = k_\theta(V_1 + V_2 - V_3 - V_4) \quad (3b)$$

$$u_\phi = k_\phi(V_1 - V_2 - V_3 + V_4) \quad (3c)$$

Donde k_ψ , k_θ , k_ϕ , son los parámetros de momentos de inercia (normalmente de valor desconocido a priori), y V_1, V_2, V_3, V_4 , las tensiones aplicadas a los motores, numerados en orden horario.

3.2. Algoritmo de control

La estrategia de control considerada en este trabajo se ha obtenido como un caso particular de la ley de control propuesta en [23]. La utilización de esta propuesta no lineal, tiene como ventaja que, a parte de su simplicidad, utiliza funciones de saturación para cada estado, lo cual proporciona un fácil ajuste de las ganancias.

De forma general, y para los tres ángulos se propone la siguiente ley de control:

$$\begin{aligned} u_{PD} &= f_s(k_d \dot{\alpha}, 1) - f_s(k_p(\alpha - d), 2) \\ u_I &= f_s(u_I + k_i(\alpha - d), 3) \\ u_\alpha &= u_{PD} + u_I \end{aligned} \quad (4)$$

donde $k_d, k_p, k_i, 1, 2, 3$ son constantes positivas, α es el ángulo considerado, y d es el valor del

ángulo deseado. f_s se define como una función de saturación acotada por los valores ± 1 .

Para asegurar teóricamente la estabilidad global, y para cada una de las variables se tienen que cumplir las siguientes desigualdades $k_d > 1$, $k_d^2 > k_p$ (para más detalles ver [23]).

Para pequeñas variaciones en los ángulos de orientación, el control de altura “z” se puede considerar desacoplado del resto de variables, siendo el modelo equivalente un doble integrador, la ley de control utilizada será la misma que se propone en la ecuación 4. Para el diseño del control de posición, es posible reescribir el modelo (2b)-(2c) obteniéndose un desacoplamiento del desplazamiento lateral “y” con respecto al avance “x”. Se ha diseñado un algoritmo de control para la posición que permita al quadrotor desplazarse de un punto a otro en línea recta y a una velocidad constante que se le envía como parámetro de entrada. Además independientemente de la orientación ψ .

Si $\mathbf{X}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ representa una medida de coordenadas cualquiera, en un sistema de referencia con x apuntando al norte e y apuntando al este; y $\mathbf{R} = \begin{pmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{pmatrix}$, con ψ definido en el eje z

que apunta al centro de la tierra. Entonces:

$$\begin{pmatrix} \mathbf{X}'_d \\ \dot{\mathbf{X}}'_{gps} \\ \ddot{\mathbf{X}}'_{imu} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & 0 & 0 & 0 \\ 0 & \mathbf{R} & 0 & 0 \\ 0 & 0 & \mathbf{R} & 0 \\ 0 & 0 & 0 & \mathbf{R} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}_d \\ \dot{\mathbf{X}}_{gps} \\ \ddot{\mathbf{X}}_{imu} \end{pmatrix} \quad (5)$$

Es la transformación que hace que el control x-y sea independiente del yaw. Por otro lado, se definen los valores de saturación de la velocidad según:

$$v = \begin{pmatrix} V_x \\ V_y \end{pmatrix} = V \cdot \begin{pmatrix} \frac{1}{1 + \left(\frac{y' - y'}{x' - x'}\right)^2} \\ \frac{1}{1 + \left(\frac{x' - x'}{y' - y'}\right)^2} \end{pmatrix} \quad (6)$$

De esta forma se consigue que las referencias de velocidad en cada eje se posicionen en un valor tal que el quadrotor se desplace de (x, y) a (x_d, y_d) en línea recta. Las siguientes ecuaciones muestran la ley de control aplicada:

$$e_v(k) = f_s(k_p \cdot \mathbf{X}'_d(k) - \mathbf{X}'_{gps}(k), v) - \dot{\mathbf{X}}'_{gps}(k)$$

$$u_I(k) = \frac{h}{2} \cdot (e_v(k) + e_v(k-1)) + u_I(k-1)$$

$$u_{xy} = k_{dv} \cdot \ddot{\mathbf{X}}'_{gps}(k) + k_{pv} \cdot e_v(k) + k_{iv} \cdot u_I(k)$$

$$\begin{pmatrix} \theta_d \\ \phi_d \end{pmatrix} = f_s(u_{xy}, \alpha)$$

donde las constantes $k_p, k_{dv}, k_{pv}, k_{iv}, \alpha$ se corresponden con los parámetros de control y $\mathbf{X}_d, \mathbf{X}_{gps}, \dot{\mathbf{X}}_{gps}, \ddot{\mathbf{X}}_{imu}$ son las coordenadas de posición deseada, de posición y velocidad dadas por el gps y de aceleración dada por la IMU, respectivamente.

La tensión aplicada a los motores, será el resultado de la suma de las distintas acciones de control sobre los ángulos (ϕ, θ, ψ) y la acción para el control en z:

$$V_1 = u + u_\psi + u_\phi + u_\theta \quad (7a)$$

$$V_2 = u - u_\psi - u_\phi + u_\theta \quad (7b)$$

$$V_3 = u + u_\psi - u_\phi - u_\theta \quad (7c)$$

$$V_4 = u - u_\psi + u_\phi - u_\theta \quad (7d)$$

4. Resultados experimentales

El objetivo de esta sección es la validación y evaluación experimental de la plataforma desarrollada y sus prestaciones.

Las figuras 4 y 5 muestran la evolución de los ángulos de orientación Pitch y Roll, y sus respectivas velocidades angulares, durante un vuelo libre a punto fijo a una altura aproximada de un metro. En esta ocasión el algoritmo de control utilizado

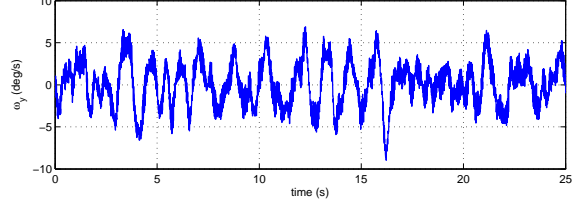
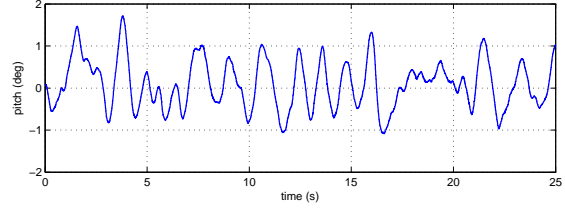


Figura 4: Control del pitch en hover

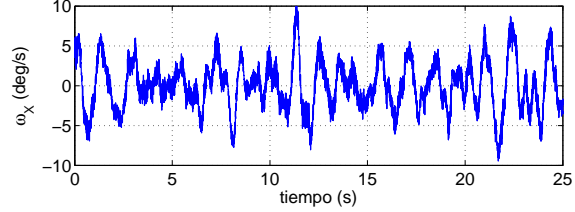
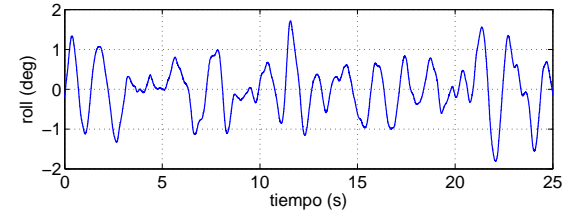


Figura 5: Control del roll en hover

para ambos ángulos es un PID, convenientemente ajustado utilizando la interfaz HMI.

Se observa el buen comportamiento de la plataforma, consiguiendo que ambos ángulos estén durante todo el vuelo dentro de un rango de +/- 2 grados.

Las figuras 6 y 7 muestran la respuesta de los ángulos Pitch y Roll frente a perturbaciones. En este caso el quadrotor está realizando un vuelo libre a punto fijo cuando se le introducen perturbaciones en los ángulos manualmente. El algoritmo de control utilizado para ambos ángulos es un UDE y las figuras destacan su rápida respuesta para volver al vuelo a punto fijo y su gran estabilidad.

La figura 8 muestra la evolución de la medida de altura y el valor de la acción de control durante un vuelo a punto fijo durante el cual se ha seguido un patrón de referencias de altura para evaluar el algoritmo de control, que en este caso es un UDE. Se observa que se siguen las referencias de forma

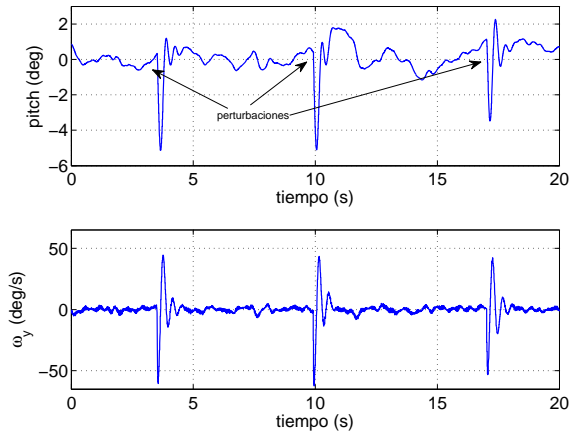


Figura 6: Respuesta en pitch ante perturbaciones

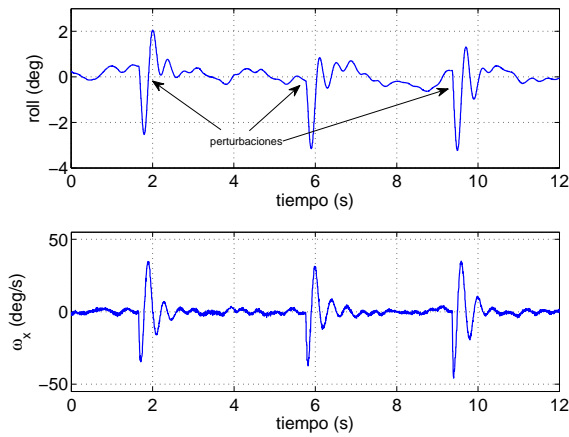


Figura 7: Respuesta en roll ante perturbaciones

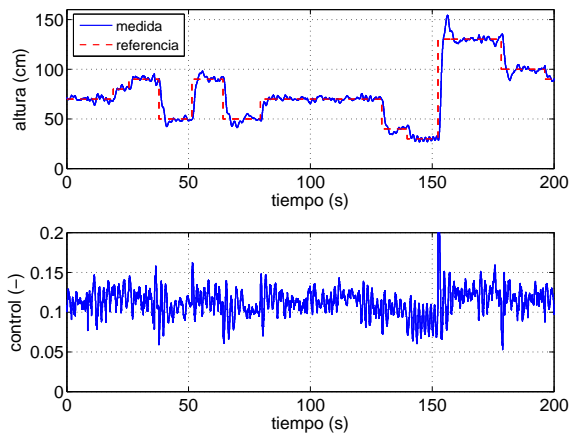


Figura 8: Control de altura siguiendo un patrón de referencias

bastante precisa y rápida.

Las figuras 9 y 10 muestran las medidas obtenidas tras someter al quadrotor a un desplazamiento de

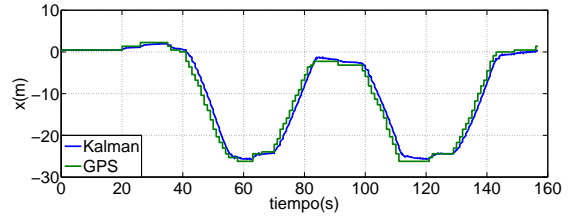


Figura 9: Medida experimental de posición tras efectuar un desplazamiento.

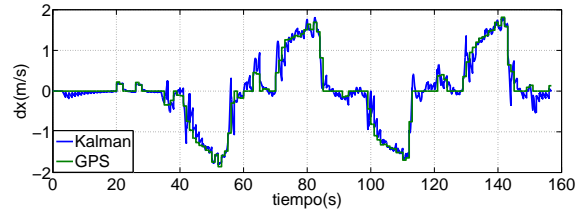


Figura 10: Medida experimental de velocidad tras efectuar un desplazamiento.

unos 25 m en dirección norte-sur.

5. Conclusiones y trabajos futuros

Para validar la plataforma, se ha desarrollado el control de vuelo tanto en interiores como en exteriores de varios quadrotors, los resultados experimentales, muestran el buen desempeño del sistema propuesto, así como una fácil y sistemática sintonización y evaluación de los algoritmos de control.

Agradecimientos

Este proyecto ha sido financiado con fondos pertenecientes al proyecto PROMETEO-2008-088 de la Generalitat Valenciana.

Referencias

- [1] L. Acosta, J. Toledo, G.N. Marichal, S. Hernández, J.N. Rodríguez, and S. Torres. Diseño e implementación de una maqueta de un helicóptero de 4 rotores para un laboratorio de control. In *XXV Jornadas de Automática*, Ciudad Real, España, 2004.
- [2] P. Agarwal and T. Brady. Slam strategy for an autonomous quadrotor.
- [3] P. Balbastre, M. Lluesma, and I. Ripoll. Análisis y compensación de los retardos de planificación en sistemas de control. *RIAI*, 3(2):40–49, 2010.

- [4] M. Béjar and A. Ollero. Modelado y control de helicópteros autónomos. revisión del estado de la técnica. *RIAI*, 5(4):5–16, 2008.
- [5] A. Berna, J. Muñoz, G. Sanahuja, and P. Garcia. Desarrollo de una plataforma para la implementación en tiempo real del control de un mini-helicóptero de 4-rotores. 2010.
- [6] J. Borenstein. The hoverbot: An electrically powered flying robot, "<http://www-personal.umich.edu/~johannb/hoverbot.htm>".
- [7] P. Castillo, R. Lozano, and A. Dzul. *Modelling and Control of Mini-Flying Machines*. Springer-Verlag in Advances in Industrial Control, julio 2005.
- [8] I. Fantoni and R. Lozano. *Nonlinear Control for Underactuated Mechanical Systems*. Springer-Verlag, 2002.
- [9] Site internet de la *Arduino*. "<http://www.arduino.cc>".
- [10] Site internet de la *IGEP*. "<http://www.isee.biz>".
- [11] Site internet de la *YGE*. "<http://www.yge.de/artikel.php?search=yge30>".
- [12] Site internet *Xenomai*. "http://www.xenomai.org/index.php/Main_Page".
- [13] Site internet *ImuMPU6050*. "<http://www.invensense.com/mems/gyro/mpu6050.html>".
- [14] Site internet *Gps*. "<http://www.adafruit.com/products/746>".
- [15] Site internet *Ultrasonidos*. "<http://www.parallax.com/product/28015>".
- [16] Site internet *Barometro*. "<http://www.meas-spec.com/product/pressure/MS5611-01BA03.aspx>".
- [17] Site internet *Magnetometro*. "<http://www.aerospace.honeywell.com/products/sensors/magnetic-sensors/magnetics-products/featured-product-hmc58831>".
- [18] Site internet *Xbee PRO S1*. "<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module>".
- [19] R. Lozano, editor. *Unmanned Aerial Vehicles, Embedded Control*. ISTE-WILEY, mayo 2010.
- [20] G.V. Raffo. *Modelado y Control de un Helicóptero Quad-Rotor*. PhD thesis, Master's thesis, Universidad de Sevilla, Dept. Ing. Sistemas y Automática, Sevilla, Spain, 2007.
- [21] G.V. Raffo, M.G. Ortega, and F.R. Rubio. Plataforma de pruebas para un vehículo aéreo no tripulado utilizando labview. Dept. Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros. Universidad de Sevilla, 2009.
- [22] H. Romero, S. Salazar, J. Escareño, and R. Lozano. Estabilización de un mini helicóptero de cuatro rotores basada en flujo óptico y sensores inerciales. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 7(2):49–56, 2010.
- [23] G. Sanahuja, P. Castillo, and A. Sanchez. Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system. *International Journal of Robust and Nonlinear Control*, 2009. DOI: 10.1002/rnc.1494.
- [24] M. Simons. *Model aircraft aerodynamics*. Model and Allied Publications, 1978.
- [25] H.J. Sussmann, E.D. Sontag, and Y. Yang. A general result on the stabilization of linear systems using bounded controls. *IEEE Transactions on Automatic Control*, 39:2411–2425, 1994.
- [26] A.R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems and Control Letters*, 18:165–171, 1992.
- [27] A.R. Teel. Semi-global stabilization of minimum phase nonlinear systems in special normal forms. *Systems and Control Letters*, 19:187–192, 1992.
- [28] A.R. Teel. Semi-global stabilization of the ball and beam using output feedback. In *American Control Conference*, San Francisco, Estados Unidos, 1993.

Anexo 2: Descripción del simulador MatLab.

El modelo que se ha programado en el simulador MatLab es el modelo siguiente:

$$\begin{aligned}\ddot{x} &= (-tg\theta\cos\psi - \tan\phi\sin\psi) \cdot g \\ \ddot{y} &= (-tg\theta\sin\psi + \tan\phi\cos\psi) \cdot g \\ \ddot{\phi} &= \frac{1}{J_{xx}} \cdot \tau_{\phi} \\ \ddot{\theta} &= \frac{1}{J_{yy}} \cdot \tau_{\theta}\end{aligned}$$

Es, por tanto, únicamente válido para el control de posición x-y, ya que se asume que existe paralelamente un control de altura que se encarga de controlar z, así como un control sobre ψ .

El único parámetro que hay que estimar en este caso son las inercias J_{xx}, J_{yy} . La idea no es obtener una versión exacta del comportamiento del quadrotor. Eso no tendría mucho sentido ya que, como se comentó previamente, éste modelo tampoco es exacto ya que omite parámetros muy importantes como lo son las fuerzas de rozamiento con el aire. La idea es tener una respuesta aproximada a la dinámica del quadrotor para así poder ensayar de manera fácil y rápida posibles realimentaciones de control y comprobar la correcta estabilidad global antes de implementarlo en la plataforma real.

¿Porque en código MatLab?

Una pregunta muy importante es porqué se ha escogido código MatLab como plataforma para programar este simulador y no otras plataformas como SimuLink. La respuesta es que tanto por su sencillez como por su similitud con el código que luego se programará (generalmente en C) en la plataforma.

En MatLab se pueden programar muy sencillamente la solución a las ecuaciones exactas del modelo, además se puede hacer que el controlador actúe sobre el proceso de manera discreta a una frecuencia determinada. Exactamente igual a como se hará después en el ordenador de abordo.

De esta forma se tiene un sistema continuo que es controlado por un controlador que se ejecuta de forma discreta. Se puede, por tanto, ver los efectos sobre el sistema al cambiar la frecuencia de ejecución.

Para el usuario la forma de operar es la siguiente: Una vez diseñada una ley de control, para simular la respuesta simplemente se programan las ecuaciones

del controlador en la parte específica de este simulador (que se encuentra debidamente señalada).

Una vez se compruebe que la ley funciona correctamente en simulación, el hecho de pasarla a código C es un paso casi instantáneo por la similitud que ofrece con el código en MatLab.

Características del simulador.

El simulador se ha programado de tal forma que sea muy sencillo para el usuario, el cual únicamente se tiene que preocupar de programar su ley de control y seleccionar diversos parámetros.

El simulador consta de diversas áreas completamente diferenciadas:

- Variables de control: Al inicio del programa. Permiten seleccionar, entre otras, el tiempo de simulación, los estados iniciales del sistema, si queremos que se de una perturbación, o si queremos incluir el viento en el modelo; también se puede seleccionar la frecuencia de ejecución del controlador y las inercias del quadrotor. Son todas las variables que el usuario puede manipular para ensayar resultados.
- Constantes de control: Donde el usuario pondrá los valores de las constantes que rigen su controlador.
- Inicialización de variables: Se inicializan ciertas variables necesarias para el bucle de simulación.
- Bucle principal de simulación:
 - Controlador: El lugar donde el usuario deberá programar su controlador.
 - Modelo: Incluye la respuesta del modelo del quadrotor.
- Plots: Al finalizar la simulación, se muestran diversas gráficas con los resultados.

Estimación de las inercias

Motores:

Forma: cilíndrica.

$$R = 3 \text{ cm}$$

$$L = 3 \text{ cm}$$

$$m = 57 \text{ g}$$

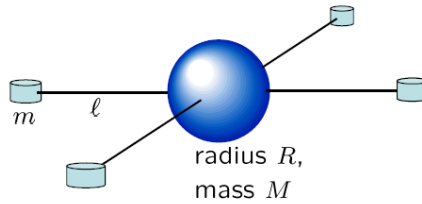


Figura 6.17: Los momentos de inercia del quadrotor se calculan asumiendo la siguiente geometría. Tiene en cuenta la masa y forma de los motores, la masa y longitud de las varillas de sujeción y la masa de toda la electrónica y batería (que se se asume como una esfera situada en el centro).

El momento de inercia de un cilindro a través de cualquier eje que pase por su c.d.g y sea paralelo al propio eje del cilindro es:

$$I = \frac{1}{4}mR^2 + \frac{1}{12}mL^2 = 1,71 \cdot 10^{-5} \text{ kg} \cdot \text{m}^2$$

Aplicando Steiner y teniendo en cuenta que la longitud de cada varilla es $l = 20\text{cm}$, los podemos trasladar al c.d.g del quadrotor:

$$I_{MOT} = 1,71 \cdot 10^{-5} + 0,057 \cdot 0,2^2 = 2,3 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$$

Varillas de sujeción:

Forma: varilla delgada.

$$l = 20 \text{ cm}$$

$$m = 25 \text{ g}$$

El momento de inercia de una varilla delgada en cualquier eje perpendicular a ella y en uno de sus extremos vale:

$$I_{VAR} = \frac{1}{3}ml^2 = 3,333 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$$

Esfera central:

El peso de todos los elementos que no son ni motores ni varillas es de 1563 kg.

Forma: esfera maciza.

$$R = 7,5 \text{ cm}$$

$$m = 1563 \text{ g}$$

El momento de inercia de una esfera maciza a través de cualquier eje que pase por su c.d.g es:

$$I_{ESF} = \frac{2}{3}mR^2 = 5,86 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$$

Por lo tanto se estiman las inercias de la siguiente forma:

$$J_{xx} = J_{yy} = 2 \cdot I_{MOT} + 2 \cdot I_{VAR} + I_{ESF} = 0,01113 \text{ kg} \cdot \text{m}^2$$

Resolución exacta del modelo en ecuaciones discretas.

Si o es el pitch, q es el roll, do y dq son las derivadas de pitch y roll, ux y uy son unas variables auxiliares y x, y, dx, dy son la posición x e y, y sus velocidades respectivamente, entonces, las ecuaciones que nos dicen de manera exacta cual es el estado $i + 1$ del modelo, conociendo el estado i y las acciones de control $px(i)$ y $py(i)$ (que es el par que actúa sobre cada uno de los ejes) son:

$$o(i+1) = o(i) + (h \cdot do(i)) + ((h^2)/(2 \cdot J_x)) \cdot px(i);$$

$$q(i+1) = q(i) + (h \cdot dq(i)) + ((h^2)/(2 \cdot J_y)) \cdot py(i);$$

$$do(i+1) = do(i) + (h/J_x) \cdot px(i);$$

$$dq(i+1) = dq(i) + (h/J_y) \cdot py(i);$$

$$ux(i+1) = 9.80665 \cdot (-\tan(o(i)) \cdot \cos(yaw(i))) - (\tan(q(i)) \cdot \sin(yaw(i)))$$

$$+ \text{acelVientoX};$$

$$uy(i+1) = 9.80665 \cdot (-\tan(o(i)) \cdot \sin(yaw(i))) + (\tan(q(i)) \cdot \cos(yaw(i)))$$

$$+ \text{acelVientoY};$$

$$x(i+1) = x(i) + h \cdot dx(i) + ((h^2)/2) \cdot ux(i+1);$$

$$y(i+1) = y(i) + h \cdot dy(i) + ((h^2)/2) \cdot uy(i+1);$$

$$dx(i+1) = dx(i) + h \cdot ux(i+1);$$

$$dy(i+1) = dy(i) + h \cdot uy(i+1);$$



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO INTEGRAL DE UN
QUADROTOR: DISEÑO DE UN ALGORITMO
DE CONTROL PARA LA POSICIÓN X-Y
BASADO EN SEÑALES GPS.**

Presupuesto

AUTOR: Castillo Frasquet, Alberto

TUTOR: García Gil, Pedro

Curso Académico: 2013-14

Presupuesto

Índice

1. Introducción.	3
2. Cuadro de precios de mano de obra.	3
3. Cuadro de precios de materiales.	4
4. Cuadro de precios de maquinaria.	6
5. Cálculo de horas dedicadas a cada tarea.	6
5.1. Capítulo 1: Desarrollo general del quadrotor.	7
5.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.	8
6. Cuadro de precios unitarios.	10
6.1. Capítulo 1: Desarrollo general del quadrotor.	10
6.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.	11
7. Cuadro de precios descompuestos.	11
7.1. Capítulo 1: Desarrollo general del quadrotor.	11
7.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.	17
8. Presupuesto de ejecución material, presupuesto de inversión y presupuesto base de licitación.	20

1. Introducción.

El presente presupuesto tiene como objetivo proporcionar toda la información sobre los costes que ha supuesto la ejecución del proyecto “*Desarrollo integral de un quadrotor: Diseño de un algoritmo de control para la posición x-y basado en señales GPS*”. Aquí se expone toda la información tanto del coste del hardware como del tiempo de desarrollo, así como el coste de la maquinaria empleada.

El proyecto se ha dividido en dos capítulos que, a su vez, se han dividido en diversas unidades de obra. El primer capítulo corresponde con la fase de desarrollo integral del quadrotor que incluye todo el proceso constructivo del quadrotor, mientras que el segundo capítulo se corresponde con el diseño del algoritmo de control de posición basado en señales GPS. No obstante, como en este proyecto también han intervenido los estudiantes *Pablo Albiol Graullera* y *Daniel Verdú Torres* de manera paralela, se deberían tener en cuenta los costes de sus tareas específicas para obtener el presupuesto total de la plataforma completa.

En primer lugar se definen los precios de la mano de obra, de los materiales y los costes de amortización de la maquinaria. Seguidamente se definen cada uno de los capítulos con sus respectivas unidades de obra, los cuales estructurado por grupos de la siguiente forma:

1. Capítulo 1: Desarrollo general del quadrotor.
 - a) Montaje de la estructura de la plataforma.
 - b) Montaje y disposición de la electrónica.
 - c) Programación del software embebido.
 - d) Implementación del control de estabilidad.
 - e) Ensayos y test de vuelo.
2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.
 - a) Montaje del hardware para la obtención de medidas GPS.
 - b) Desarrollo del software para el procesamiento de la señal GPS.
 - c) Diseño de la ley de control x-y.
 - d) Diseño del simulador MatLab.

2. Cuadro de precios de mano de obra.

Como mano de obra han intervenido 3 graduados en ingeniería en tecnologías industriales, así como 2 técnicos de laboratorio. Se ha tomado el coste mensual y teniendo en cuenta las horas de trabajo diarias así como los días laborales se obtiene el precio por hora.

Posteriormente se multiplicará el salario por las horas dedicadas a cada una de ellas.

CUADRO DE PRECIOS DE LA MANO DE OBRA					
Código	Empleado	Salario (€ / mes)	Horas diarias	Días laborales	Salario (€ / h)
MO.GITI	Graduado en Tecnologías Industriales	1200	8	20	7,5
MO.TEC	Técnico de laboratorio	1000	8	20	6,25

3. Cuadro de precios de materiales.

El cuadro siguiente muestra el precio, junto con la cantidad que se debería adquirir, de los diferentes materiales para el correcto desempeño del proyecto.

La columna *Ud.* indica el número de unidades de producto que vienen por el precio indicado.

Cuadro de materiales					
Código	Ud.	Denominación del material	Precio (€)	Aporte unitario	Total (€)
MT.EST	u	Estructura Mikrokopter (41 cm diagonal)	62,05	1	62,05
MT.MBR	u	Tren de aterrizaje de Mikrokopter	33,85	1	33,85
MT.MBR	u	Motores brushless Robbe-Roxy 2827-35	51,2	4	204,8
MT.VAR	u	Variadores YGE 30i	69	4	276
MT.HEL	x2u	Hélices 10x4,5	5	2	10
MT.CMC	u	Contrachapado de madera de chopo 160x160 cm	18,7	1	18,7
MT.PEE	u	Poliestireno expandido 100x100 cm	2	1	2
MT.COB	x100u	Conectores bala 3,5 mm	12,25	1	12,25
MT.CACP	x25u	Conectores atornillados para cables de potencia	7,6	1	7,6
MT.JTO	u	Juego de tornillos 330 uds	3,8	1	3,8
MT.CDTB	x5u	Conector DeansT para batería o fuente	4,9	1	4,9
MT.INT	u	Interruptores	1,95	2	3,9
MT.BRI	x100u	Bridas	1,9	1	1,9
MT.CAS	u	Cinta aislante	2,3	1	2,3

MT.BAT	u	Batería LiPo 3S GENS-ACE 5300 mAh	59	1	59
MT.CPE	u	Bote de cola para poliestireno expandido	6,75	1	6,75
MT.MCB	u	Medidor de carga de baterías LiPo	5,99	1	5,99
MT.ARDU	u	Arduino Due	47,19	1	47,19
MT.IGE	u	Igep v2	179	1	179
MT.ARM1	u	Arduino Micro	24,2	1	24,2
MT.ULT	u	Sensor de ultrasonidos Ping	18,03	1	18,03
MT.IMU	u	Drotek IMU 10DOF v2 = MPU6050 + HMC5883 + MS5611	22,9	1	22,9
MT.GPS	u	Sensor GPS Ultimate v3 Adafruit	29,25	1	29,25
MT.MRA	u	Modulo de radiofrecuencia Xbee Pro S1	27,78	1	27,78
MT.RRC	u	Receptor RC Corona RP8D1	18,2	1	18,2
MT.CAL	m	Cables de alimentación	2,32	1,5	3,48
MT.RTS	u	Regulador de tensión	1,95	2	3,9
MT.CNL	u	Convertidor de niveles lógicos	2,16	2	4,32
MT.LED	x20u	Led	1,63	1	1,63
MT.BSN	u	Bobina de estaño	4,54	1	4,54
MT.IDC	x10u	Conectores IDC	2,95	1	2,95
MT.PWM	x10u	Conectores PWM	1,9	1	1,9
MT.RES	u	Kit de resistencias	5,9	1	5,9
MT.CCO	x40u	Cables para comunicaciones	3,75	1	3,75
MT.CJ	u	Conector Jack 2,1 x 5,5 mm	1,05	1	1,05
MT.EF	u	Emisora Futaba T6EX 35MGHz	124,02	1	124,02
MT.JSTK	u	Joystick logitech xtreme x3 pro	46,21	1	46,21
MT.MSD	u	Tarjeta micro SD 2GB	2,92	1	2,92
MT.ADPT	u	Adaptador FTDI	12,95	1	12,95
MT.PBA	u	Placa baquelita 10 x 10 cm	13,3	1	13,3
MT.PMF	u	PCB manufacturada	3	3	9
MT.MTLB	u	Licencia MatLab	1900	1	1900

4. Cuadro de precios de maquinaria.

El cuadro siguiente describe toda la maquinaria que sería preciso adquirir (o alquilar), junto con el número de unidades necesarias, para la correcta ejecución del presente proyecto. También se detalla el número de horas de uso de cada una de ellas. Se asume que toda la maquinaria se adquirirá, única y exclusivamente, para la realización de este proyecto por lo que para el cálculo de este presupuesto se ha reducido la vida útil al número de horas de uso.

CUADRO DE MAQUINARIA						
Código	Ud.	Denominación de la maquinaria	Precio (€)	Unidades	Horas de uso	Amortización
M.OSM	u	Ordenador completo de sobremesa	518	1	150	3,45
M.OSC	u	Osciloscopio	1064,95	1	10	106,50
M.OP	u	Ordenador Portátil	719	2	500	2,88
M.FA	u	Fuente de alimentación 40A	146,85	1	100	1,47
M.CDB	u	Cargador baterías LiPo	14,6	1	50	0,29
M.POL	u	Polímetro	22,8	1	125	0,18
M.TAL	u	Taladradora con juego de brocas y limas	42	1	20	2,10
M.PTF	u	Pistola termo-fusible	8,5	1	2	4,25
M.SDE	u	Soldador de estaño	11,9	1	30	0,40
M.AHC	u	Arco cortador de hilo caliente	37,81	1	4	9,45
M.HER	u	Kit de herramientas	53,23	1	100	0,53
M.CLS	h	Alquiler cortadora laser de control numérico	25	1	1	25,00

5. Cálculo de horas dedicadas a cada tarea.

Para obtener los precios de cada unidad de obra se han calculado el número de horas dedicadas a ella tanto por cada graduado en ingeniería en tecnologías industriales (GITI en las tablas), como por cada técnico de laboratorio (TEC en las tablas).

Las siguientes tablas muestran el desglose de todas las tareas realizadas en cada unidad de obra con las horas dedicadas a ellas.

5.1. Capítulo 1: Desarrollo general del quadrotor.

C1-UO 1		Montaje de la estructura de la plataforma				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Ensamblaje del chasis y tren de aterrizaje	9	3	0,5	1	27	0,5
Diseño de la protección	11	2	0	0	22	0
Montaje del sistema de alimentación	4	3	0,5	1	12	0,5
Montaje de los actuadores	4	3	0	0	12	0
				TOTAL	73	1

C1-UO 2		Montaje y disposición de la electrónica				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Montaje PCB del Arduino	5	2	1	1	10	1
Instalación de la IGEP en la plataforma	1	3	1	2	3	2
Conexionado de sensores	2	3	0	0	6	0
Pruebas de sensores	10	3	0	0	30	0
				TOTAL	49	3

C1-UO 3		Programación del software embebido				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Programación del código del Arduino Due	13	3	0	0	39	0
Instalación Linux + Xenomai en la IGEP	7	2	0	0	14	0
Programación del código de la IGEP	17	3	0	0	51	0
				TOTAL	104	0

C1-UO 4		Implementación del control de estabilidad				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Calibración hélices	1	1	0	0	1	0
Programación de los variadores	3	2	0	0	6	0
Ajuste motores	3	2	0	0	6	0
Comprobación señales y comunicaciones	10	3	0	0	30	0
Ajuste experimental de las constantes	9	3	0	0	27	0
				TOTAL	70	0

C1-UO 5		Ensayos y test de vuelos				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Test de vuelos en el laboratorio	12	3	0	0	36	0
Test de vuelos en exterior	13	3	0	0	39	0
				TOTAL	75	0

5.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.

C2-UO 1		Montaje del hardware para la obtención de medidas GPS				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Conexión de todos los elementos necesarios	9	1	0	0	9	0
Comprobaciones y validaciones	1	1	0	0	1	0
				TOTAL	10	0

C2-UO 2		Desarrollo del software para el procesamiento de la señal GPS.				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Software de lectura del sensor GPS	5	1	0	0	5	0
Gestión de comunicaciones A.Micro-A.Due	10	1	0	0	10	0
Procesamiento de la señal GPS	8	1	0	0	8	0
Implementación algoritmo Kalman	40	1	0	0	40	0
Comprobaciones y validaciones	30	1	0	0	30	0
				TOTAL	93	0

C2-UO 3		Diseño de la ley de control x-y.				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Estudio del modelo teórico del quadrotor	5	1	0	0	5	0
Estudio de la ley x-y sin control de velocidad	3	1	0	0	3	0
Desarrollo ley de control con control de velocidad	12	1	0	0	12	0
Mejoras a la ley de control	22	1	0	0	22	0
Ajuste de constantes	4	1	0	0	4	0
				TOTAL	46	0

C2-UO 4		Diseño del simulador MatLab.				
Tareas	Tiempo GITI (h)	nº GITI	Tiempo TEC (h)	nº TEC	TOTAL GITI (h)	TOTAL TEC (h)
Programación del simulador	30	1	0	0	30	0
				TOTAL	30	0

6. Cuadro de precios unitarios.

Se exponen aquí los costes finales de cada una de las unidades de obra.

6.1. Capítulo 1: Desarrollo general del quadrotor.

Código	Descripción	
C1-UO1	<i>Montaje de la estructura de la Plataforma</i>	Precio (€):
		1.519,15
C1-UO2	<i>Montaje y disposición de la electrónica</i>	Precio (€):
		1.383,13
C1-UO3	<i>Programación del software embebido</i>	Precio (€):
		993,22
C1-UO4	<i>Implementación del control de estabilidad</i>	Precio (€):
		683,77
C1-UO5	<i>Ensayos y test de vuelo</i>	Precio (€):
		767,63

6.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.

Código	Descripción	Precio (€):
C2-UO1	Montaje del hardware para la obtención de medidas GPS	100,03
C2-UO2	Desarrollo del software para el procesamiento de la señal GPS.	1.008,04
C2-UO3	Diseño de la ley de control x-y.	491,71
C2-UO4	Diseño del simulador MatLab.	2.278,06

7. Cuadro de precios descompuestos.

7.1. Capítulo 1: Desarrollo general del quadrotor.

C1-UO1	Montaje de la estructura de la Plataforma	<i>Montaje de todos los elementos básicos de la estructura de un quadrotor. Incluyendo chasis, tren de aterrizaje, actuadores, protección, etc.</i>			
Código	Ud.	Descripción	Precio (€)	Rendimiento	Importe (€)
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	74,00	555,00
MO.TEC	h	Técnico de Laboratorio	7,00	1,00	7,00
M.AHC	h	Arco cortador de hilo caliente	9,45	4,00	37,81
M.PTF	h	Pistola termo-fusible	4,25	1,50	6,38

M.TAL	h	Taladradora con juego de brocas y limas	2,10	10,00	21,00
M.HER	h	Kit de herramientas	0,53	75,00	39,92
M.CLS	h	Alquiler cortadora laser de control numérico	25,00	1,00	25,00
MT.EST	u	Estructura Mikrokopter (41 cm diagonal)	62,05	1,00	62,05
MT.MBR	u	Tren de aterrizaje de Mikrokopter	33,85	1,00	33,85
MT.MBR	u	Motores brushless Robbe-Roxy 2827-35	51,20	4,00	204,80
MT.VAR	u	Variadores YGE 30i	69,00	4,00	276,00
MT.HEL	u	Hélices 10x4,5	5,00	2,00	10,00
MT.CMC	u	Contrachapado de madera de chopo 160x160 cm	18,70	1,00	18,70
MT.PEE	u	Poliestireno expandido 100x100 cm	2,00	1,00	2,00
MT.COB	u	Conectores bala 3,5 mm	12,25	1,00	12,25
MT.CACP	u	Conectores atornillados para cables de potencia	7,60	1,00	7,60
MT.JTO	u	Juego de tornillos 330 uds	3,80	1,00	3,80
MT.CDTB	u	Conector DeansT para batería o fuente	4,90	1,00	4,90
MT.INT	u	Interruptores	1,95	2,00	3,90
MT.BRI	u	Bridas	1,90	1,00	1,90
MT.CAS	u	Cinta aislante	2,30	1,00	2,30
MT.BAT	u	Batería LiPo 3S GENS-ACE 5300 mAh	59,00	1,00	59,00
MT.CPE	u	Bote de cola para poliestireno expandido	6,75	1,00	6,75

MT.CAL	m	Cables de alimentación	2,32	1,50	3,48
MT.BAT	u	Batería LiPo 3S GENS-ACE 5300 mAh	59,00	1,00	59,00
MT.CPE	u	Bote de cola para poliestireno expandido	6,75	1,00	6,75
MT.CAL	m	Cables de alimentación	2,32	1,50	3,48
Total					1.474,62
Costes directos complementarios				0,01	14,75
Costes indirectos				0,02	29,79
Total unidad de obra					1.519,15

C1-UO2	Montaje y disposición de la electrónica		<i>Montaje de todos los equipos electrónicos, sensores y conexión para poder controlar el quadrotor</i>		
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	49,00	367,50
MO.TEC	h	Técnico de Laboratorio	7,00	3,00	21,00
M.OSC	h	Osciloscopio	106,50	4,00	425,98
M.OSM	h	Ordenador completo de sobremesa	3,45	20,00	69,07
M.OP	h	Ordenador Portátil	2,88	20,00	57,52
M.FA	h	Fuente de alimentación 40A	2,40	10,00	23,97
M.POL	h	Polímetro	0,18	40,00	7,30
M.TAL	h	Taladradora con juego de brocas y limas	2,10	10,00	21,00
M.PTF	h	Pistola termo-fusible	4,25	0,50	2,13
M.SDE	h	Soldador de estaño	0,40	12,00	4,76
MT.ARDU	u	Arduino Due	47,19	1,00	47,19
MT.IGE	u	Igep v2	179,00	1,00	179,00
MT.ARMÍ	u	Arduino Micro	24,20	1,00	24,20
MT.ULT	u	Sensor de ultrasonidos Ping	18,03	1,00	18,03

MT.IMU	u	Drotek IMU 10DOF v2 = MPU6050 + HMC5883 + MS5611	22,90	1,00	22,90
MT.RTS	u	Regulador de tensión	1,95	1,00	1,95
MT.CNL	u	Convertidor de niveles lógicos	2,16	1,00	2,16
MT.LED	u	Led	1,63	1,00	1,63
MT.BSN	u	Bobina de estaño	4,54	1,00	4,54
MT.IDC	u	Conectores IDC	2,95	1,00	2,95
MT.PWM	u	Conectores PWM	1,90	1,00	1,90
MT.RES	u	Kit de resistencias	5,90	1,00	5,90
MT.CCO	u	Cables para comunicaciones	3,75	1,00	3,75
MT.CJ	u	Conector Jack 2,1 x 5,5 mm	1,05	1,00	1,05
MT.MSD	u	Tarjeta micro SD 2GB	2,92	1,00	2,92
MT.PBA	u	Placa baquelita 10 x 10 cm	13,30	1,00	13,30
MT.PMF	u	PCB manufacturada	3,00	3,00	9,00
Total					1.342,58
Costes directos complementarios				0,01	13,43
Costes indirectos				0,02	27,12
Total unidad de obra					1.383,13

C1-UO3	Programación del software embebido	Programación de todos los algoritmos y unidades de control necesarios para controlar el quadrotor, incluyendo la programación en el Arduino Due y la Igep.			
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	104,00	780,00
M.OSM	h	Ordenador completo de sobremesa	3,45	20,00	69,07
M.OP	h	Ordenador Portátil	2,88	40,00	115,04
Total					964,11
Costes directos complementarios				0,01	9,64
Costes indirectos				0,02	19,47
Total unidad de obra					993,22

C1-UO4	Implementación del control de estabilidad	Programación e implementación de las leyes de control para la estabilidad del quadrotor			
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	70,00	525,00
M.OSM	h	Ordenador completo de sobremesa	3,45	10,00	34,53
M.OP	h	Ordenador Portátil	2,88	25,00	71,90
M.FA	h	Fuente de alimentación 40A	1,47	20,00	29,37
M.CDB	u	Cargador baterías LiPo	0,29	10,00	2,92
Total					663,72
Costes directos complementarios				0,01	6,64
Costes indirectos				0,02	13,41
Total unidad de obra					683,77

C1-UO5	Ensayos y test de vuelo	<i>Realización de vuelos de prueba y comprobación del correcto funcionamiento de todos los subsistemas.</i>			
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	75,00	562,50
M.OSM	u	Ordenador completo de sobremesa	3,45	20,00	69,07
M.OP	u	Ordenador Portátil	2,88	15,00	43,14
M.FA	u	Fuente de alimentación 40A	1,47	40,00	58,74
M.CDB	u	Cargador baterías LiPo	0,29	40,00	11,68
Total					745,13
Costes directos complementarios				0,01	7,45
Costes indirectos				0,02	15,05
Total unidad de obra					767,63

7.2. Capítulo 2: Diseño del algoritmo de control x-y basado en señales GPS.

Código	Ud.	Descripción	Precio	Rendimiento	Importe
C2-UO1	Montaje del hardware para la obtención de medidas GPS		<i>Incluye todo el montaje de todo hardware necesario, así como las comprobaciones pertinentes para incluir el GPS en la plataforma.</i>		
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	6,00	45,00
M.FA	h	Fuente de alimentación 40A	1,47	5,00	7,34
M.POL	h	Polímetro	0,18	6,00	1,09
M.SDE	h	Soldador de estaño	0,40	8	3,17
M.HER	h	Kit de herramientas	0,53	10	5,32
MT.GPS	u	Sensor GPS Ultimate v3 Adafruit	29,25	1	29,25
MT.CNL	u	Convertidor de niveles lógicos	2,16	1	2,16
MT.CCO	u	Cables para comunicaciones	3,75	1	3,75
Total					97,09
Costes directos complementarios				0,01	0,97
Costes indirectos				0,02	1,96
Total unidad de obra					100,03

C2-UO2	Desarrollo del software para el procesamiento de la señal GPS.		<i>Incluye todo el tiempo invertido que ha sido necesario para el correcto desarrollo de un software que procese las medidas GPS.</i>		
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	93,00	697,50
M.OSM	u	Ordenador completo de sobremesa	3,45	15,00	51,80
M.OP	h	Ordenador Portátil	2,88	74,00	212,82
M.FA	h	Fuente de alimentación 40A	2,40	5	11,98
M.POL	h	Polímetro	0,18	24	4,38
Total					978,48
Costes directos complementarios				0,01	9,78
Costes indirectos				0,02	19,77
Total unidad de obra					1.008,04

C2-UO3	Diseño de la ley de control x-y.		<i>Incluye todo el tiempo y recursos invertidos para el diseño y calibración de la ley de control x-y</i>		
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	46,00	345,00
M.OP	h	Ordenador Portátil	2,88	46,00	132,30
Total					477,30
Costes directos complementarios				0,01	4,77
Costes indirectos				0,02	9,64
Total unidad de obra					491,71

C2-UO4	Diseño del simulador MatLab.		<i>Incluye todo el tiempo y recursos invertidos para el diseño y programación del simulador MatLab de algoritmos de control x-y</i>		
Código	Ud.	Descripción	Precio	Rendimiento	Importe
MO.GITI	h	Graduado en Tecnologías Industriales	7,50	30,00	225,00
M.OP	h	Ordenador Portátil	2,88	30,00	86,28
MT.MTLB	u	Licencia MatLab	1900	1	1900,00
Total					2.211,28
Costes directos complementarios				0,01	22,11
Costes indirectos				0,02	44,67
Total unidad de obra					2.278,06

8. Presupuesto de ejecución material, presupuesto de inversión y presupuesto base de licitación.

Código	Concepto	Coste (€)
C1-U01	Montaje de la estructura de la Plataforma	1519,15
C1-U02	Montaje y disposición de la electrónica	1383,13
C1-U03	Programación del software embebido	993,22
C1-U04	Implementación del control de estabilidad	683,77
C1-U05	Ensayos y test de vuelo	767,63
C2-U01	Montaje del hardware para la obtención de medidas GPS	100,03
C2-U02	Desarrollo del software para el procesamiento de la señal GPS.	1008,04
C2-U03	Diseño de la ley de control x-y.	491,71
C2-U04	Diseño del simulador MatLab.	2278,06
PRESUPUESTO TOTAL DE EJECUCIÓN MATERIAL		9224,74
15% gastos generales		1383,71
6% beneficio industrial		553,48
TOTAL PRESUPUESTO DE INVERSIÓN		11161,94
21% IVA		2344,01
PRESUPUESTO BASE DE LICITACIÓN		13505,94

Asciende el presupuesto total de ejecución material a la siguiente cantidad expresada en EUROS:

NUEVE MIL DOSCIENTOS VEINTICUATRO EUROS CON SETENTA Y CUATRO CÉNTIMOS.

Asciende el presupuesto total de inversión a la siguiente cantidad expresada en EUROS:
ONCE MIL CIENTO SESENTA Y UN EUROS CON NOVENTA Y CUATRO CÉNTIMOS.

Asciende el presupuesto base de licitación a la siguiente cantidad expresada en EUROS:
TRECE MIL QUINIENTOS CINCO EUROS CON NOVENTA Y CUATRO CÉNTIMOS.