

FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA



PROYECTO DE FIN DE CARRERA DE INGENIERÍA
ELÉCTRICA

Implementación de un UAV con arquitectura de cuadricóptero

Autores:

Santiago PATERNAIN
Rodrigo ROSA
Matías TAILANIÁN

Tutor:

Ing. Rafael CANETTI



10 de julio de 2012

Queremos agradecer en primer lugar a nuestras familias, amigos y novias por el apoyo que nos han brindado a lo largo de toda la carrera y fundamentalmente durante este proyecto.

Dentro del instituto queremos agradecer a Leonardo Steinfeld y a Pablo Toscano por su colaboración en más de una oportunidad y a Sergio y Roberto del Taller que nos ayudaron constantemente con los distintos dispositivos que tuvimos que implementar.

Por último queremos agradecer al panadero por prestarnos en varias ocasiones la balanza para realizar la caracterización de los motores.

RESÚMEN

El proyecto tiene por objetivo fundamental el diseño y la integración de un sistema de control que permita el vuelo autónomo de un cuadricóptero comercial radiocontrolado. Se presentan en esta documentación la caracterización y el modelado del cuadricóptero comercial TurboAce X720 (o LotusRC T850), la integración del mismo con una inteligencia adquirida y sensores. Se presentan además los detalles del filtro de Kalman implementado para la estimación del estado y los detalles del controlador LQR con acción integral diseñado. Dichos algoritmos fueron testeados exhaustivamente con diversas simulaciones y pruebas sobre el sistema real, las cuales también se presentan. Se desarrolló también un simulador en MatLab para probar los algoritmos de control.

ÍNDICE GENERAL

Introducción	1
I. Descripción del sistema	3
1. Visión general del proyecto	5
1.1. Acciones de control básicas	5
1.2. Componentes del sistema y su interacción	6
2. Hardware	9
2.1. Plataforma física: Cuadricópteros	9
2.2. Inteligencia	13
2.3. Instrumentación	14
2.3.1. IMU	14
2.3.2. GPS	16
2.3.3. Resumen de sensores	17
II. Modelado del sistema	19
3. Modelo Físico	21
3.1. Hipótesis de trabajo	21
3.1.1. La Tierra como sistema de referencia inercial	21
3.1.2. Curvatura de la Tierra	23
3.1.3. Atracción gravitacional	23
3.1.4. Fuerzas aerodinámicas	23
3.2. Sistema de referencia	23
3.3. Cinemática	25
3.4. Dinámica del Sistema	26
3.4.1. Primera Cardinal	26
3.4.2. Segunda Cardinal	27
3.5. Fuerzas y momentos involucrados	28
3.5.1. Peso	29

Índice general

3.5.2. Fuerzas y momentos de las hélices	29
3.6. Modelo en variables de Estado	32
4. Simulaciones del modelo físico	35
4.1. Estructura del Simulador	35
4.1.1. Lazo Abierto	36
4.1.2. Lazo cerrado	37
4.2. Simulaciones	40
III. Motores	45
5. Ingeniería inversa	47
5.1. Introducción al protocolo i^2c	48
5.2. Pruebas en régimen	49
5.3. Prueba de arranque	52
5.4. Prueba de frenado	53
5.5. Conclusiones	55
5.6. Verificaciones	56
6. Caracterización de los motores y las hélices	59
6.1. Objetivo	59
6.2. Materiales	59
6.3. Procedimiento	60
6.3.1. Consideraciones previas	60
6.3.2. Relación entre comando I^2C , velocidad angular y empuje	61
6.3.3. Relación entre velocidad angular y torque	62
6.3.4. Respuesta al Escalón	63
6.4. Resultados y análisis	64
6.4.1. Comparación entre motores	64
6.4.2. Obtención de la curva fuerza contra velocidad angular	65
6.4.3. Obtención de la curva I^2C contra velocidad angular	66
6.4.4. Obtención de la curva velocidad angular contra torque	67
6.4.5. Respuesta al escalón	68
IV. Sensores y estimación del estado	69
7. Calibración del acelerómetro	71
7.1. Objetivos	71
7.2. Materiales	71
7.3. Marco Teórico	71
7.4. Procedimiento	73
7.4.1. Caracterización de las no idealidades variables	73
7.4.2. Determinación de parámetros estáticos	73
7.4.3. Variación de la medida con la temperatura	74
7.5. Resultados y análisis	75
7.5.1. Medida estática durante una hora	75
7.5.2. Medias estáticas para el acelerómetro	76

7.5.3. Variación con la temperatura	78
8. Calibración del giróscopo	81
8.1. Objetivos	81
8.2. Materiales	81
8.3. Marco teórico	81
8.4. Procedimiento	82
8.4.1. Caracterización de las no idealidades variables	82
8.4.2. Determinación de parámetros estáticos	82
8.5. Resultados y análisis	83
8.5.1. Medias para la determinación de los parámetros estáticos del giróscopo	83
8.5.2. Variación con la temperatura	84
9. Calibración del Magnetómetro	87
9.1. Objetivos	87
9.2. Procedimiento	87
9.3. Resultados y Análisis	88
10.GPS	91
10.1. Objetivos	91
10.2. Materiales	91
10.3. Procedimiento	91
10.3.1. Adquisición de datos	92
10.3.2. Verificación del polígono	93
10.3.3. Punto fijo - 2 minutos	93
10.3.4. Polígono	96
10.3.5. Polígono - Análisis	97
10.4. Caminata por el borde del polígono	98
10.4.1. Caminata - Análisis	99
10.5. Error en altura	99
10.6. Tiempo de <i>warmup</i>	100
10.7. Tasa de muestreo	100
10.8. Posición Absoluta	100
10.9. Conclusiones	101
11.Barómetro	103
11.1. Objetivos	103
11.2. Materiales	104
11.3. Consideraciones previas	104
11.4. Procedimiento	104
11.5. Análisis y resultados	105
11.5.1. Drift y <i>warm-up</i>	105
11.5.2. Caracterización del ruido	106
11.5.3. Altura absoluta	107
11.5.4. Distancias de varios metros	107
11.5.5. Distancias de un metro	108
11.5.6. Distancias de decenas de centímetros	111
11.6. Conclusiones	112

Índice general

12.Filtro de Kalman	113
12.1. Filtro de Kalman Extendido (EKF)	114
12.1.1. Modelo matemático	114
12.1.2. Esquema general del estimador de estados	116
12.2. Resultados: Kalman inercial	118
12.2.1. Simulación de vuelo	118
12.2.2. Vuelo real	120
12.3. Resultados Kalman inercial + GPS	120
V. Desarrollo del controlador	123
13.Linealización y puntos de operación	125
13.1. Concepto general	125
13.2. Puntos de operación	126
13.2.1. Hovering	127
13.2.2. Vuelo en linea recta a velocidad constante	127
13.2.3. Vuelo en círculos	128
14.Diseño del controlador	131
14.1. Conceptos generales sobre LQR	132
14.2. Consideraciones particulares respecto del sistema a controlar	133
14.2.1. Discretización del sistema	133
14.2.2. Agregado de integradores	134
14.3. Controlabilidad y observabilidad	135
14.4. Matriz de realimentación	137
15.Simulaciones realizadas sobre el controlador	139
15.1. Respuesta al escalón	139
15.1.1. Trayectoria de hovering	139
15.1.2. Vuelo en linea recta	141
15.1.3. Vuelo en círculos	142
15.2. Robustez frente a la presencia de ruido de medición	143
15.2.1. Hovering	143
15.3. Robustez frente a variaciones en el modelo	146
16.Pruebas del controlador	149
16.1. Control de subsistemas Roll y Pitch	149
16.2. Control del subsistema del Yaw	151
16.3. Control del sistema completo	154
VI.Implementación	157
17.Montaje	159
17.1. BeagleBoard	159
17.2. IMU	160
17.3. GPS	161
17.4. Conversor lógico	161
17.5. Protección	161

18. Software	163
18.1. Esquema general	163
18.2. Software en la <i>BeagleBoard</i>	165
18.2.1. Requerimientos e inicialización del programa principal	165
18.2.2. Loop	166
18.2.3. Módulo <i>imu_comm</i>	168
18.2.4. Módulo <i>kalman</i>	170
18.2.5. Módulo de <i>control</i>	171
18.2.6. Generador de rutas	173
18.2.7. Driver de los motores y módulo <i>motor</i>	173
18.2.8. Tiempos	174
18.2.9. Comunicación	174
18.2.10. Configuración	174
18.3. Software en la <i>IMU</i>	175
18.3.1. Cambios al software original	176
18.3.2. Diagrama de flujo	176
19. Electrónica	177
20. Switcheo Manual/Automático	179
20.1. Señales del control remoto	179
20.2. Señal elegida	179
20.3. Análisis de la señal	180
20.4. Detección de la posición de la llave	180
Conclusión	183
Anexos	187
A. Aplicación con cámara	189
A.1. Sistemas de coordenadas	190
A.2. Marcadores	190
A.3. Detección del marcador	191
A.4. Resultados	192
A.4.1. Prueba N° 5	192
B. Especificaciones técnicas de la Mongoose 9DoF IMU	195
B.1. Acelerómetro	195
B.2. Giróscopo	195
B.3. Magnetómetro	196
B.4. Sensor de presión	196
B.5. Sensor de temperatura	197
C. Cálculo del tensor de inercia	199
C.1. Magnitudes a considerar	199
C.2. Tensor de inercia del sistema	200
C.3. Resultados	201

Índice general

D. Cálculos necesarios para la linealización del sistema	203
D.1. Linealización para cualquier trayectoria	203
D.2. Linealización para la condición de hovering	205
D.3. Vuelo en linea recta a velocidad constante	206
D.4. Vuelo a velocidad angular constante	206
E. GPS - Información adicional	207
E.1. Geometría: DOP - <i>Dilution of precision</i>	207
F. Configuración y compilación del código	209
F.0.1. Obtención del código - Git	209
F.0.2. Comunicación con la <i>BeagleBoard</i>	209
F.0.3. Tiempos	210
F.0.4. Consideraciones de seguridad	211
F.1. Salida - Logs	211
F.2. Debugging	212
F.2.1. Ejecución a partir de un log	212
F.3. Kernel	212
F.3.1. <i>Compilación: Bitbake+OpenEmbedded</i>	213
F.4. IMU	215
F.4.1. Compilación	215
F.5. Trabajo a futuro	216
G. Hardware - Cuidados	217
G.1. Baterías del Cuadricóptero (LiPo)	217
G.2. Baterías de la electrónica - Beaglejuice	217
G.3. Motores y Hélices	218
G.3.1. Hélices	218
G.3.2. Motores	218
H. Manual de usuario	219
H.1. Montaje y conexionado	219
H.2. Ajuste del Offset	220
H.3. Pruebas del Controlador	221
I. Manual de usuario del simulador	223
J. Costos	229
Bibliografía	231

INTRODUCCIÓN

Existe actualmente en el Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de la República una línea de investigación orientada a la robótica móvil y al desarrollo de técnicas de control. En este marco, se han desarrollado diversas plataformas experimentales tanto por docentes del Departamento de Control como por estudiantes durante los proyectos de fin de carrera. En particular, dos proyectos de fin de carrera se orientaron al diseño de un UAV¹ con arquitectura de avión. El primero de ellos, *AMFO1-Bobby*, se avocó al diseño y construcción de un avión de dimensiones reducidas con capacidad de vuelo relativamente prolongada. Dicha plataforma fue diseñada para ser volada gracias a un control remoto. El proyecto AUION, continuó el trabajo resolviendo el diseño del controlador del avión en la situación de vuelo, quedando excluido el despegue y el aterrizaje del vehículo.

En el presente proyecto se continúa trabajando en la línea presentada, diseñando e integrando un sistema de control a un vehículo aéreo comercial con arquitectura de cuadricóptero de forma de lograr el vuelo autónomo del mismo. Por vuelo autónomo se entiende la capacidad de seguir una trayectoria autogenerada a partir de puntos del espacio introducidos por un usuario por los cuales se desea “pasar”, de ahora en más *waypoints*.

La arquitectura escogida ofrece una gran variedad de aplicaciones entre las cuales se puede nombrar la fotografía y el cine, el relevamiento de zonas peligrosas o contaminadas, la vigilancia y la navegación bajo techo.

No es el objetivo de este proyecto desarrollar un producto final, sino que por el contrario lo que se plantea es desarrollar una plataforma experimental sobre la cual otros grupos puedan trabajar, profundizando en diversas áreas como la estimación de estados, técnicas de control más avanzadas, la generación de rutas óptimas y la cooperación entre varias plataformas idénticas.

El objetivo del presente proyecto incluye instrumentar dos cuadricópteros de forma de poder obtener todas las variables de estado con las que se elige representar al sistema. La comprensión del comando de los motores y la caracterización de los mismos son parte fundamental para el control del sistema. Se desarrollará una pri-

¹Unmanned aerial vehicle

Índice general

mer versión de un generador de rutas que defina la trayectoria a seguir a partir de waypoints introducidos por el usuario. Se desarrollará un controlador que permita seguir la trayectoria generada por el generador de rutas. Se deberá programar e integrar una CPU a cada cuadricóptero para realizar las tareas hasta aquí planteadas, manteniendo la posibilidad de trabajar en modo automático y manual.

Como ya fue explicitado, es de interés que este proyecto sea la plataforma para el desarrollo de proyectos que permitan una profundización en las líneas de trabajo planteadas, por dicho motivo se incluye dentro del alcance el diseño de un simulador que permita, a lo largo de este y de futuros proyectos, testear los algoritmos de control en desarrollo.

Parte I.

Descripción del sistema

CAPÍTULO 1

VISIÓN GENERAL DEL PROYECTO

En este capítulo se describen brevemente las distintas partes que componen el proyecto de forma de que el lector se haga una idea general del mismo.

El esquema general de un cuadricóptero se puede apreciar en la figura 1.1¹. Se compone de dos ejes perpendiculares, en cuyos extremos se ubican los propulsores (motores y hélices). En dicha figura también se observan las velocidades angulares de los motores (ω), así como las fuerzas (T) y torques (M) producidos por ellos. Tanto los torques como las fuerzas de los propulsores dependen de la velocidad angular de las hélices. Por dicho motivo, lo que se busca es actuar sobre los motores, variando su velocidad angular para realizar las distintas acciones de control.

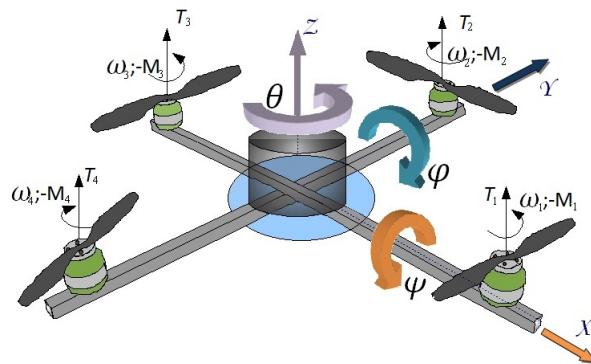


Figura 1.1.: Esquema general de un cuadricóptero

1.1. Acciones de control básicas

Existe una velocidad angular de los motores para la cual la fuerza total producida es igual al peso, esa velocidad angular permite que el cuadricóptero permanezca suspendido con velocidad vertical nula, situación conocida como *hovering*.

¹Las imágenes 1.1, 1.2a, 1.2b y 1.3 se basan en la original tomada de <http://www.isys.uni-stuttgart.de/>

1. Visión general del proyecto

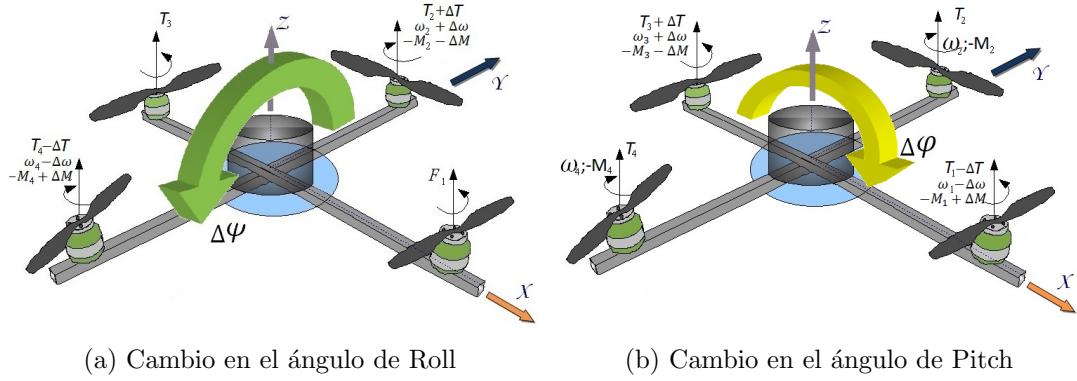


Figura 1.2.

Si se aumenta uniformemente la velocidad angular de los motores la fuerza producida por los mismos supera al peso y el sistema se acelera hacia arriba. La reducción de la velocidad angular produce el efecto contrario.

Para realizar una rotación se debe crear un desbalance entre los torques producidos por los motores. Si se desea aumentar el ángulo de Roll (ψ), debe disminuirse la velocidad angular del motor 4 y aumentar la del motor 2, manteniendo la fuerza neta igual a la fuerza necesaria para lograr el hovering. Esta situación se encuentra representada en la figura 1.2a. Análogamente, para aumentar el ángulo de Pitch debe aumentarse la velocidad angular del motor 3 y disminuir la velocidad angular del motor 1 (ver figura 1.2b). Finalmente, si se desea aumentar el ángulo de Yaw se debe disminuir la velocidad angular de los motores 1 y 3 y aumentar la de los motores 2 y 4, manteniendo la fuerza neta igual a la fuerza de hovering. Esta última situación es la presentada en la figura 1.3.

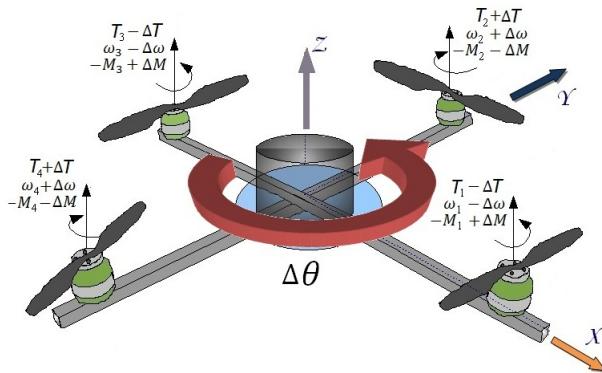


Figura 1.3.: Cambio en el ángulo de Yaw

Las acciones de control descriptas anteriormente pueden ser combinadas de forma de lograr trayectorias más complejas, gran parte de los capítulos siguientes intentan explicar que acción debe realizarse sobre cada motor para lograr diversos objetivos.

1.2. Componentes del sistema y su interacción

A continuación se dará una visión general del sistema que se desea implementar. En la figura 1.4 se presenta un diagrama de bloques de la plataforma física comercial

1.2. Componentes del sistema y su interacción

adquirida, y en la figura 1.5 el esquema general completo. En la proxima sección se verá en detalle el hardware utilizado.

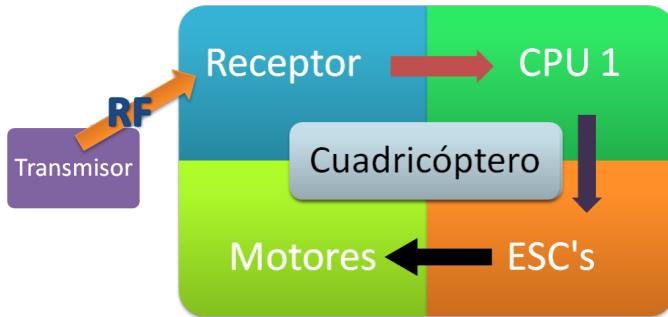


Figura 1.4.: Esquema general de la plataforma física comercial

La plataforma física elegida es un cuadricóptero comercial radio controlado. El mismo cuenta con los elementos indispensables para poder volarlo manualmente: transmisor, receptor, CPU1, motores y ESCs². La solución adoptada agrega:

- IMU³ y GPS: Sensores que permiten obtener una medida directa o indirecta de las variables de estado del sistema.
- CPU2: Es el centro del sistema de control a implementar, se encarga de procesar los datos de los sensores y de decidir la acción de control a realizar.
- WiFi: Permite la comunicación con el mundo exterior de forma de facilitar la programación de los algoritmos y de modificar o agregar waypoints durante el vuelo.
- Switcheo: Una de las señales del receptor será utilizada para definir si el control de los motores estará comandado por la CPU1 o por la CPU2.

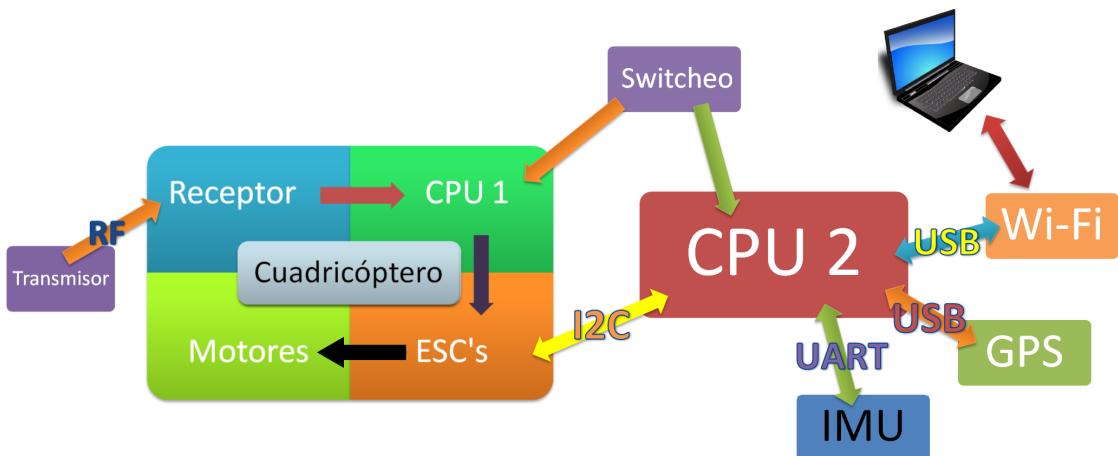


Figura 1.5.: Esquema general de interconexión

²Electronic Speed Controller

³Inertial Measurement Unit

CAPÍTULO 2

HARDWARE

La elección del hardware representa una parte muy importante del Proyecto. Una mala elección de alguno de los componentes puede acarrear complicaciones no previstas a la hora de la ejecución, causando contratiempos inesperados y trabajo excesivo. Es necesario, entonces, el estudio detallado de cada uno de los componentes a utilizar.

2.1. Plataforma física: Cuadricópteros

A la hora de la planificación del Proyecto se plantea la opción de partir de cero, diseñando completamente el cuadricóptero. Si bien esta opción tiene la ventaja de que se puede conocer completamente su funcionamiento, implica problemas de ingeniería mecánica que no se pretenden resolver, conocimientos de resistencia, flexibilidad y peso de materiales, así como también diversas complicaciones adicionales a la hora de fabricar y armar los distintos componentes. Dados los plazos del proyecto, y que su objetivo se centra en el control del vehículo, la necesidad de partir de un dispositivo electromecánico ya construido resulta imperiosa.

La plataforma escogida es el cuadricóptero **Turbo Ace X720**. Esta plataforma es controlada por un usuario a través de un control remoto comercial estándar (Walkera 2801). El sistema receptor se encarga de *setear* la velocidad angular de cada motor en función de la señal recibida.

La plataforma escogida consta de un receptor, una CPU, una batería, 4 motores y 4 ESCs¹ montados sobre un cuerpo de fibra de carbono. Además incluye un transmisor (control remoto), imprescindible para volarlo. La electrónica original (ESCs y CPU)



Figura 2.1.: Turbo Ace X720

¹Electronic Speed Controller

2. Hardware

también se puede encontrar bajo el nombre de **Lotus RC T580**, otra plataforma comercial.

Motores

Los motores de la plataforma son de tipo *Brushless* (sin escobillas): son eléctricos con el rotor de imán permanente y tres pares de bobinas en el estator alimentados con corriente continua conmutada. Necesitan de un sistema de conmutación eléctrico (**ESC**) que los controlen, y presentan relaciones lineales entre corriente y torque, y entre frecuencia y velocidad. Son comúnmente utilizados en vehículos radio-controlados por su eficiencia, potencia, durabilidad y su bajo peso (en comparación con los de tipo *Brushed*).

Como se explica en [1], la energía eléctrica es convertida a mecánica mediante las fuerzas de atracción magnéticas entre el imán permanente del rotor y el campo magnético giratorio inducido por los devanados del rotor. En la figura 2.2, también tomada de [1], se muestra un diagrama simplificado de un motor *Brushless*.

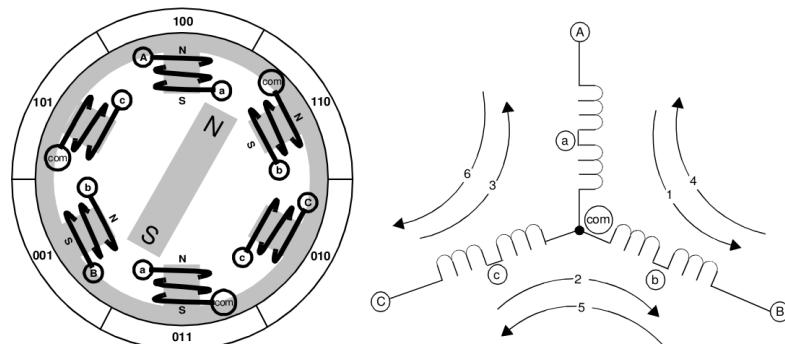


Figura 2.2.: Esquema simplificado de motor *Brushless*

En este ejemplo se muestran 3 circuitos electromecánicos conectados a un punto común y separados en el centro para permitir el movimiento del rotor. La mayoría de los motores *Brushless* tienen un devanado en 3 fases con topología de estrella, como se muestra en la figura anterior. Un motor con esta topología se maneja energizando 2 de las fases al mismo tiempo. Para energizar en el momento adecuado cada pareja de fases se utiliza un controlador electrónico de velocidad, como se explicó anteriormente.

Controlador electrónico de velocidad (ESC)

Los ESCs incluídos en la plataforma reciben comandos mediante el protocolo I^2C , a $333kHz$. Se encargan de conmutar la corriente continua que reciben los motores de modo de energizar la pareja de fases correspondiente en el momento justo para que el campo electromagnético inducido genere el máximo torque posible sobre el imán del rotor. En la práctica el máximo torque es alcanzado cuando el imán permanente del rotor está desfasado 90° respecto del campo magnético resultante en el rotor.

La clave del funcionamiento de un motor *Brushless* es poder sensar la posición del rotor y en función de ello energizar las fases que ejercerán mayor torque. Para dicho sensado comúnmente se utiliza un sensor de posición de 3 elementos basado

en el efecto Hall.

El rotor se mueve 60 grados por cada paso de la conmutación. La ruta apropiada para la corriente del estator es activada cuando el rotor se encuentra a 120° de la dirección del campo magnético del estator y es desactivada cuando el rotor pasa 60° dicha dirección, momento en el cual se activa el siguiente circuito y se repite el proceso. En cada instante una de las fases del motor es conducida a nivel alto, otra a nivel bajo y la restante se deja “flotando”.

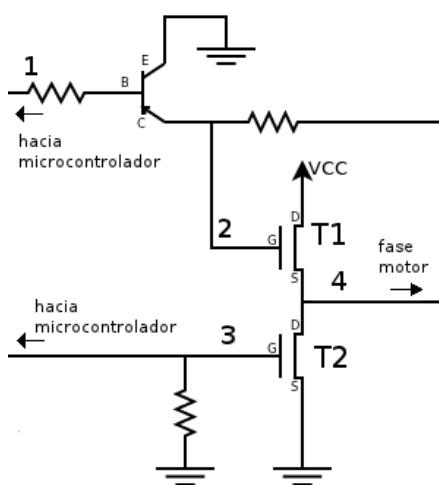


Figura 2.3.: Conmutación BLDC

Para realizar la conmutación se utiliza una configuración conocida como “puente H”, donde por cada fase se utilizan 2 transistores encargados de conducir a los niveles alto y bajo y un tercer transistor (*driver*) que maneja al primero. En la figura 2.3 se muestra un diagrama típico de un circuito utilizado para conmutar corriente continua en una fase de un motor *Brushless*. Una precaución muy importante que se debe considerar para este tipo de motores es no activar nunca al mismo tiempo los caminos al nivel bajo y al nivel alto, es decir que nunca conduzcan al mismo tiempo los transistores **T1** y **T2**. Los estados viables para los transistores son las mostradas en la tabla 2.1.

Configuración	T1	T2	Fase
A	Cortado	Cortado	Flotando
B	Conduciendo	Cortado	Nivel alto
C	Cortado	Conduciendo	Nivel bajo

Cuadro 2.1.: Configuraciones posibles para los transistores

Si se impone un voltaje mayor a $0,7V$ en la base del *driver* (punto 1 de la figura 2.3), dicho transistor conducirá, lo cual fijará un voltaje de $0,2V$ en el punto 2 del diagrama, correspondiente al *gate* del transistor **T1**. Este voltaje en el *gate* provoca que el transistor quede cortado, no dejando pasar corriente entre el *source* y el *drain*. En esta situación hay 2 escenarios posibles: que el transistor **T2** se encuentre conduciendo o que se encuentre cortado. Para que se dé el primer escenario (configuración **C**) es necesario que el microcontrolador imponga un voltaje positivo en el *gate* del transistor **T2**, logrando un camino de baja impedancia entre tierra y la fase en cuestión del motor. En este caso se estaría conduciendo dicha fase a nivel bajo. El otro escenario implica que dicha fase quede en estado “flotando” (configuración **A**) ya que ambos transistores permanecen cortados. Por último la otra situación posible es que el transistor **T1** conduzca (configuración **B**), situación en la cual el transistor **T2** deberá permanecer cortado, lo cual se logra llevando a $0V$ el punto 1 del diagrama.

En la figura 2.4 se muestra el diagrama de un ESC obtenido al relevar el con-

2. Hardware

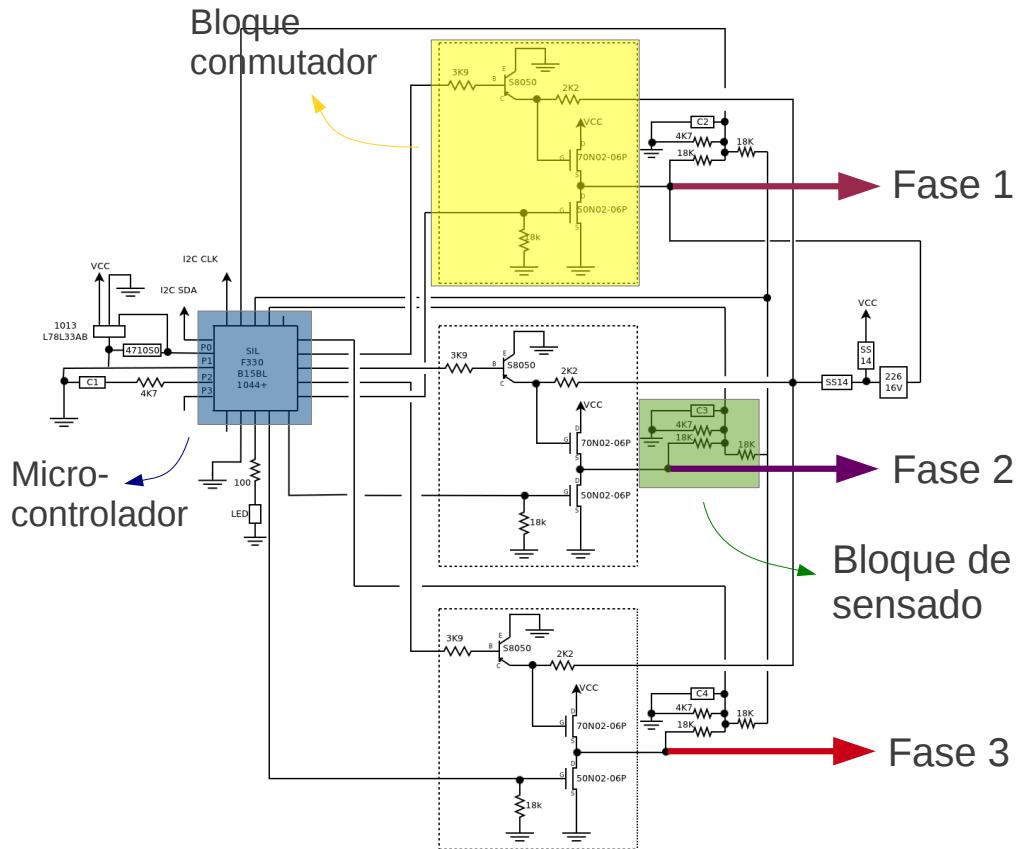


Figura 2.4.: Esquemático de un ESC

trolador que originalmente incluía el cuadricóptero comercial adquirido². Se pueden identificar los diversos bloques mencionados a lo largo de esta sección. El bloque de sensado es utilizado para obtener la posición del rotor en cada instante de tiempo, el bloque de conmutación es el encargado de energizar las fases correspondientes en cada paso y microcontrolador es el encargado de interpretar la información del sensado de la posición del rotor y actuar sobre los transistores para generar el campo magnético rotatorio. Las fases **1**, **2** y **3** se conectan a los terminales **A**, **B** y **C** mostrados en la figura 2.2. El denominado “puente H” se forma entre cada pareja de bloques de conmutación y las bobinas del motor que quedan energizadas.

Instrumentación e inteligencia

La instrumentación con la que cuenta el dispositivo está destinada a simplificar el manejo a control remoto, estabilizando el sistema. La misma incluye un barómetro y acelerómetro y giróscopo de 3 ejes. Cuenta además con una CPU que se encarga de la lectura de los datos proporcionados por los sensores y de definir las acciones

²Este controlador se dejó de fabricar, existen dos versiones nuevas de ESC, una de ellas mantiene el mismo protocolo de comunicación que el aquí presentado, esto fue verificado, ya que en la última versión de la plataforma implementada se utilizó este controlador. Los fabricantes indican que hay la otra versión que utiliza un protocolo de comunicación distinto, no se trabajó con estos ESC.

de control sobre los motores. No se dispone del código del programa presente en la CPU, y por lo tanto no es posible reutilizarlo, ni tampoco obtener la lectura de los sensores presentes. Se realizaron esfuerzos para conseguir dicha información del fabricante, quien se negó a proporcionarla. Por dicho motivo se opta por dotar al sistema de instrumentación adicional.

Tiempo de vuelo

El tiempo de vuelo utilizando una batería de LiPo de 5300mAh (450g) y una carga de 500g es cercano a los 10-15 minutos.

Carga útil

Además de toda la instrumentación que incluye el cuadricóptero, se incorporará una placa de desarrollo, una de sensores (IMU), un GPS, una interfaz para comunicación Wi-Fi, y una batería dedicada exclusivamente a alimentar la electrónica agregada. Es de interés conservar la posibilidad de integrar una cámara fotográfica. La carga que puede levantar el cuadricóptero es de 1300g, de la que se debe descontar el peso de las baterías de la electrónica y de la alimentación a los motores.

2.2. Inteligencia



Figura 2.5.: Beagleboard

Además de la plataforma física, deben seleccionarse componentes electrónicos capaces de procesar la información proveniente de los sensores, computar y ejecutar los algoritmos de control, y generar las señales necesarias para transmitir las instrucciones a los motores.

La solución implementada se basa en la plataforma de desarrollo *Beagleboard*. Esta plataforma posee un microprocesador TI ARM Cortex A8 de 1GHz y un DSP TMS320C64x+ de 800MHz. La memoria RAM es de 512 Mb. No tiene memoria no volátil, pero soporta tarjetas SD de hasta 4GB y es posible conectar memorias flash USB. Ver figura 2.5.

Dimensiones y peso

Cuanto menor sean las dimensiones y peso de la placa elegida, mayor carga útil quedará disponible. El peso total de la BeagleBoard con una batería, diseñada especialmente alimentarla es de aproximadamente 200 gramos, y sus dimensiones son de $10\text{cm} \times 8,5\text{cm} \times 5\text{cm}$. La autonomía de dicha batería es de aproximadamente 2 horas.

Sistema Operativo, Puertos y I/O

La BeagleBoard ofrece la posibilidad de cargar un kernel de *Linux* desde una tarjeta SD. El sistema operativo permite:

- Comunicación a través de una interfaz ethernet.

2. Hardware

- Conexión de cuatro dispositivos USB de manera simultánea.
- Debugging a través de un puerto serie.
- Uso de bibliotecas para visión por computadora.
- Compatibilidad con módulos de cámaras VGA.

Existen *drivers* que permiten agregar una interfaz para WiFi conectando un módulo USB (Belkin wireless G).

La BeagleBoard cuenta con un puerto de expansión de 28 pines de propósito general. Los pines 23 y 24 de dicho puerto pueden ser configurados para exponer un puerto I^2C , fundamental para enviar comandos a los ESC. Los pines 23 y 24 corresponden a las líneas de datos y al reloj respectivamente. Por otra parte, los pines 8 y 6 pueden ser configurados para funcionar como pines de transmisión y recepción de un puerto serie, esto será fundamental para la comunicación con algunos de los sensores.

2.3. Instrumentación

Para poder controlar el sistema es importante poder conocer los valores que toman las variables de estado del mismo. Como se verá en el capítulo sobre el desarrollo del modelo físico del cuadricóptero, las variables que se deben conocer son:

- La orientación.
- La posición.
- La velocidad.
- La velocidad angular.

Por dicho motivo resulta imprescindible dotar al sistema de sensores capaces de medir dichas magnitudes, directa o indirectamente. Se compararon distintas alternativas comerciales, finalmente se optó por un dispositivo que integra varios sensores y un GPS.

2.3.1. IMU

Acelerómetro

Un acelerómetro es un dispositivo capaz de medir su aceleración propia en el marco de referencia de un sistema en caída libre. Se encarga de procesar las medidas, y convertirlas a una señal eléctrica (puede ser analógica o digital).

Existen diversos tipos de acelerómetros. Se eligió trabajar con un acelerómetro contenido en un circuito integrado, tecnología MEMS. Los acelerómetros basados en esta tecnología miden cambios internos de la transferencia de calor causada por la aceleración, ofreciendo ventajas significativas sobre el empleo de una estructura tradicional sólida de masas de prueba.

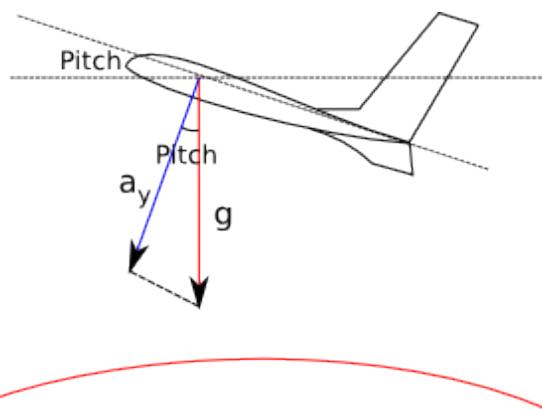


Figura 2.6.: Determinación del ángulo de Pitch

Se optó por utilizar tecnología MEMS fundamentalmente por tamaño, peso y costo. Los mismos son más pequeños, livianos y baratos que otras tecnologías.

El acelerómetro cumple dos roles fundamentales:

1. Bajo la hipótesis de que las aceleraciones a las cuales se verá sometido el sistema **no** son comparables con g , sirve para determinar los ángulos de Pitch y de Roll. En la figura 2.6 se ilustra la forma de lograr esto para el ángulo de Pitch. Las ecuaciones utilizadas pueden encontrarse en 18.
2. Permite obtener una estimación de la velocidad, integrando su medida.

Giróscopo

Un giróscopo es un instrumento capaz de medir la velocidad angular de un sistema solidario a si mismo respecto a un sistema inercial. Las mismas restricciones sobre tamaño, peso y costo que se aplicaban para el acelerómetro se aplican aquí. Se vuelve a optar por tecnología MEMS.

Desde el punto de vista teórico, procesando la información obtenida a partir del acelerómetro y del giróscopo y partiendo de condiciones iniciales conocidas, se puede conocer, en todo momento, la posición del sistema y su orientación. En la práctica, sin embargo, esto no es así, ya que todas las medidas tienen un cierto error. Para obtener la orientación y la posición en cada instante se debe integrar y, por lo tanto, se integra también el error cometido. Esto produce una acumulación de errores que afecta, de forma considerable, el resultado final luego de cierta cantidad de muestras. Este problema se puede resolver integrando un GPS. Este último permitiría, cada cierto intervalo de tiempo, contar con una medida absoluta para comparar contra los resultados obtenidos de la integral de las medidas del acelerómetro y del giróscopo, permitiendo corregir errores debido a la deriva por integración del error.

Sensor de presión

El GPS, mencionado en la sección anterior, provee una estimación de la altura absoluta, pero es poco precisa. Depende fuertemente de la cantidad de satélites disponibles y se ve fuertemente deteriorada por el efecto del rebote de las ondas en estructuras cercanas (en especial por los rebotes en el piso, que deterioran la

2. Hardware

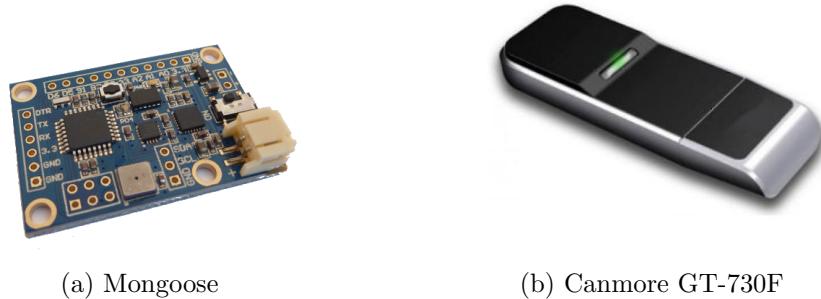


Figura 2.7.: Instrumentación

precisión en la medida de altura cuando se está muy cerca del suelo). Por este motivo es necesario contar con otra fuente de información. Se optó por incorporar un barómetro que, midiendo la presión absoluta, permite determinar la altura del cuadricóptero en forma independiente al resto de los sensores.

Magnetómetro

El *drift* en la estimación del ángulo de Yaw a partir de la integración de datos del giróscopo y la falta de una referencia absoluta en dicha estimación, se resuelven agregando un magnetómetro. Permite ubicar el norte magnético (medida absoluta), complementando la información diferencial obtenida del giróscopo.

Sensor de temperatura

El control del sistema no depende directamente de la temperatura, pero la instrumentación presenta una dependencia con (y por ende tendrá errores asociados a) la temperatura. El contar con un sensor de temperatura permitirá utilizar una calibración que tome en cuenta dicha dependencia.

La Mongoose 9DoF IMU (2.7a) posee todos los sensores que se han nombrado hasta aquí, excepto el GPS. Las características de dichos sensores se resumen en el anexo B. Posee además un microprocesador sencillo de programar, lo que permite realizar un preprocessamiento de los datos antes de la estimación del estado. Las modificaciones realizadas al código original se explican en el anexo ??.

2.3.2. GPS

La elección del GPS se basó fundamentalmente en lograr la simplicidad del sistema. Existían muchas opciones, placas de diversos tamaños, con distintos tipos de antenas, pero todas con especificaciones similares.

Se optó por utilizar un GPS *Canmore GT-730F*, como el mostrado en la figura 2.7b. Dentro de un margen de precios razonable las características de los distintos GPS comerciales no difieren considerablemente en lo que refiere a tasa de muestreo, resolución y exactitud. La razón fundamental para elegir dicho instrumento sobre otros fue la facilidad de conexión ya que permite utilizar los puertos USB de la plataforma a cargo de la inteligencia y los drivers existentes para linux.

2.3.3. Resumen de sensores

A continuación se presenta un resumen de las prestaciones de los sensores a utilizar, en la configuración en la que se utilizaron. Por más detalles referirse a B.

	Dato nuevo	Resolución
Acelerómetro XY	10ms (x2)	4mg
Acelerómetro Z	10ms (x2)	4mg
Giróscopo XY	10ms (x2)	0.07 °/s
Giróscopo Z	10ms (x2)	0.07 °/s
Barómetro	10ms	1Pa
Magnetómetro XY	10ms (x2)	5 mGa
Magnetómetro Z	10ms (x2)	5 mGa
GPS	1s	-

Parte II.

Modelado del sistema

CAPÍTULO 3

MODELO FÍSICO

Para controlar el cuadricóptero resulta imprescindible comprender cabalmente su comportamiento. Con esta óptica, lo que se busca es obtener el modelo más sencillo que sea capaz de representar adecuadamente al sistema. El objetivo de este capítulo es el de realizar el desarrollo de dicho modelo. La forma de modelar el sistema elegida es un Modelo en Variables de Estado, de ahora en más MVE. Lo que se busca es obtener una representación del sistema de la forma $\dot{\vec{x}} = f(\vec{x}, \vec{u}, t)$, donde \vec{x} es el vector de estados del sistema, \vec{u} es el vector que representa las entradas del sistema y t es el tiempo.

Al tratarse de una plataforma comercial no se dispone de todos los parámetros fundamentales para el desarrollo de dicho modelo, a modo de ejemplo, no se conoce como es la respuesta de los motores ni el tensor de inercia del sistema. La obtención de dichos parámetros se detalla en el capítulo 6 y en el anexo C respectivamente. En el presente análisis nos limitaremos a obtener las ecuaciones que gobiernan el comportamiento del cuadricóptero.

3.1. Hipótesis de trabajo

No debe perderse de vista que el cuadricóptero con el que se trabaja fue diseñado para ser comandado a través de un control remoto. Por dicho motivo es razonable que sus características sean tales que le permitan a una persona volarlo desde el suelo. Parece perfectamente razonable que se haya diseñado el sistema para volar por tiempos limitados, distancias relativamente cortas y a bajas velocidades. Estas consideraciones permiten introducir diversas simplificaciones en el modelado.

3.1.1. La Tierra como sistema de referencia inercial

La Tierra **no** es un sistema de referencia inercial ya que la misma se encuentra sometida a la traslación en torno al Sol y a una rotación en sobre su eje. Supongamos una partícula que se encuentra en movimiento relativo a la Tierra. Su aceleración respecto de un sistema verdaderamente inercial puede escribirse como:

$$\vec{a} = \vec{a}' + \vec{a}_T + \vec{a}_C \quad (3.1)$$

3. Modelo Físico

Donde:

$$\vec{a}_T = \vec{a}_{O'} + \dot{\vec{\omega}} \times \vec{r}' + \vec{\omega} \times (\vec{\omega} \times \vec{r}') \quad (3.2)$$

$$\vec{a}_C = 2\vec{\omega} \times \vec{v}' \quad (3.3)$$

En las ecuaciones anteriores \vec{a}' es la aceleración de la partícula en el sistema relativo, \vec{v}' es la velocidad relativa de la partícula, \vec{r}' la posición relativa, $\vec{\omega}$ es la velocidad angular de la Tierra y $\vec{a}_{O'}$ es la aceleración del centro de masa de la Tierra. a_T y a_C son la aceleración de transporte y de Coriolis respectivamente. El objetivo es ahora analizar si es posible considerar $\vec{a} \approx \vec{a}'$. Si esto último se cumple se puede aproximar a la Tierra como un sistema inercial.

El radio promedio de la órbita Heliocéntrica (R_H) es de $1,5 \times 10^8 \text{ Km}$, dicha órbita se recorre en 365 días lo que nos da una velocidad promedio de:

$$V_T = \frac{2\pi \times 1,5 \times 10^{11} \text{ m}}{365 \times 24 \times 60 \times 60 \text{ s}} \approx 29,89 \times 10^3 \text{ ms}^{-1} \quad (3.4)$$

Esta velocidad implica que la aceleración del centro de masa de la Tierra ($\vec{a}_{O'}$) es de aproximadamente:

$$a_{O'} = \frac{V_T^2}{R_H} \approx 5,95 \times 10^{-3} \text{ ms}^{-2} \quad (3.5)$$

Por otra parte sabemos que la Tierra rota sobre su eje una vez cada 24 horas con velocidad angular constante, tenemos así que:

$$\omega = \frac{2\pi}{24 \times 60 \times 60 \text{ s}} \approx 7,3 \times 10^{-5} \text{ rads}^{-1} \quad (3.6)$$

Finalmente debemos considerar el radio promedio de la Tierra ($R_T = 6,731 \times 10^6 \text{ m}$), y notar que las alturas que alcanzará nuestro sistema son despreciables respecto del radio de la Tierra. Por lo tanto,

$$a_T \approx 4,18 \times 10^{-2} \text{ ms}^{-2} \quad (3.7)$$

Para que el término de la aceleración de Coriolis sea comparable con la aceleración de transporte las velocidades del cuadricóptero relativas a la Tierra deberían ser del orden de cientos de metros por segundo, condición que evidentemente no se cumple, por lo tanto este término puede ser despreciado.

La aceleración relativa al sistema de la Tierra difiere en aproximadamente $4,18 \times 10^{-2} \text{ ms}^{-2}$ de la aceleración medida en un sistema verdaderamente inercial. Por otra parte la resolución del acelerómetro utilizado es de $4mg \approx 4,81 \times 10^{-2} \text{ ms}^{-2}$. Con el sensor elegido para trabajar es imposible distinguir entre ambas aceleraciones, por dicho motivo parece razonable despreciar el término de la aceleración que corresponde a la aceleración de transporte. Esto nos permite afirmar que la Tierra puede aproximarse como un sistema de referencia inercial y por lo tanto se cumple que:

$$\sum \vec{F}_{ext} = m\vec{a} \approx m\vec{a}' \quad (3.8)$$

3.1.2. Curvatura de la Tierra

El cuadricóptero no se desplazara una distancia superior a una centena de metros paralelo a la superficie de la Tierra. Consideremos un caso extremo en el cual el cuadricóptero se desplaza 1km en una dirección. Esto corresponde a recorrer un arco de círculo de $\theta = \frac{1\text{km}}{6,731 \times 10^3 \text{km}} \approx 1,5 \times 10^{-4} \text{rad}$. Si consideramos la superficie terrestre como un plano, la distancia recorrida es:

$$d = R_T \sin(\theta) \approx 0,999999996\text{km} \quad (3.9)$$

Como era de esperar, la diferencia entre la distancia recorrida como arco de círculo y asumiendo una aproximación local de la Tierra por un plano es despreciable. Por lo tanto trabajaremos con un sistema de coordenadas cartesiano.

3.1.3. Atracción gravitacional

Todos los objetos se encuentran relacionados entre sí por medio de la Fuerza de atracción gravitacional. Sin embargo en las cercanías de la Tierra la atracción gravitacional con el resto de los objetos es completamente despreciable. Dicha fuerza vale:

$$F_G = G \frac{M_T m}{d^2} = G \frac{M_T m}{(R_T + h)^2} \quad (3.10)$$

Donde G es la constante gravitacional y h la altura a la cual se encuentra una partícula. Como ya se analizo anteriormente las alturas a las cuales se desenvolverá el cuadricóptero son despreciables respecto del radio de la Tierra. Por lo tanto la ecuación anterior queda:

$$F_G \approx \frac{M_T m}{R_T^2} = gm \quad (3.11)$$

donde g es la constante gravitacional de la Tierra, su valor es aproximadamente $9,81\text{m s}^{-2}$

3.1.4. Fuerzas aerodinámicas

Debido a las bajas velocidades que logrará el cuadricóptero se decidió despreciar las fuerzas de tipo aerodinámicas, salvo aquellas responsables de las fuerzas y momentos de las hélices. Estas fuerzas serán analizadas con mayor profundidad en la sección 3.5.

3.2. Sistema de referencia

A lo largo del presente desarrollo se trabajará constantemente con dos sistemas de referencia: uno inercial¹ solidario a la tierra (S_I) y otro solidario al cuadrcóptero (S_q) como se muestra en la figura 3.1. El sistema S_I es un sistema local donde la dirección \vec{i} , \vec{j} y \vec{k} corresponden a las direcciones Norte, Oeste y hacia arriba y el origen es la posición inicial del cuadricóptero. En la figura 3.1 se pueden apreciar ambos sistemas de referencia. El sistema S_q se puede obtener realizando tres rotaciones compuestas

¹De acuerdo a la aproximación realizada

3. Modelo Físico

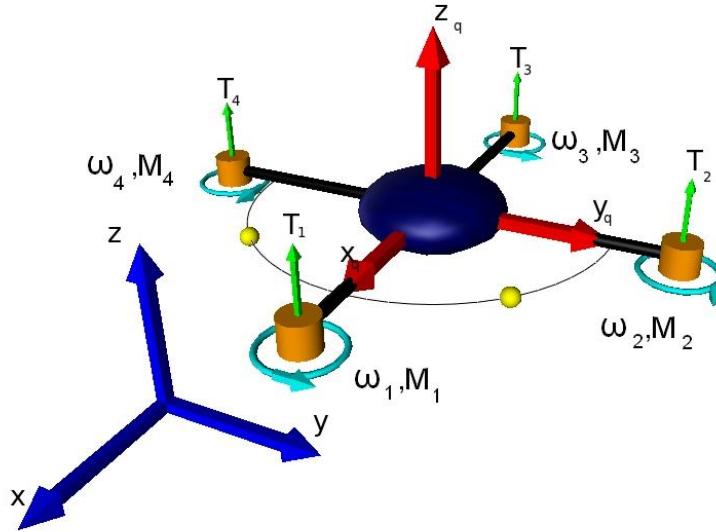


Figura 3.1.: Modelo del cuadrcóptero

del sistema S_I , dichas rotaciones se muestran en la figura 3.2. Los ángulos θ , φ y ψ son conocidos como ángulos de Euler.

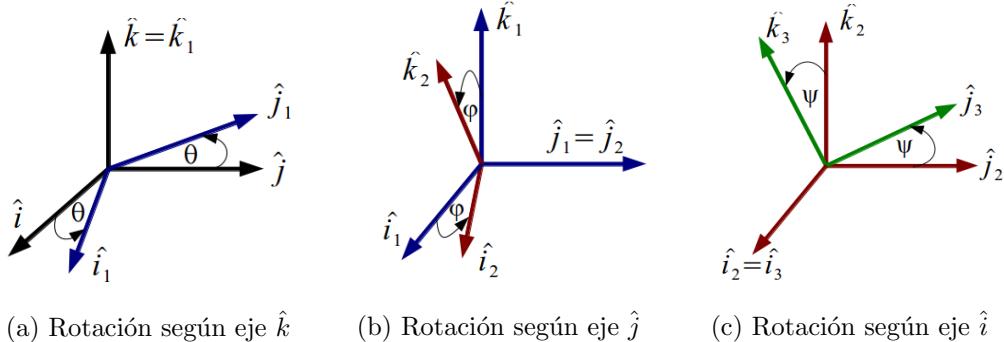


Figura 3.2.: Rotaciones

La importancia del sistema S_q radica en las simplificaciones que introduce a la hora de escribir las ecuaciones, ya que por ejemplo en dicho sistema las direcciones del empuje de las hélices, de los torques que introducen y de las velocidades angulares de los motores del cuadricóptero son constantes. Asimismo, algunos de los sensores del sistema de navegación (acelerómetro, giróscopo y magnetómetro) realizan medidas referenciadas al sistema de coordenadas solidario al cuadricóptero, es decir en el sistema S_q . Considerar este sistema resulta en una simplificación del procesamiento de los datos obtenidos por la IMU.

Las tres transformaciones pueden representarse matricialmente de la siguiente forma:

$$H_I^1 = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} H_1^2 = \begin{pmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{pmatrix} H_2^q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{pmatrix} \quad (3.12)$$

La transformación de las coordenadas del sistema inercial al sistema solidario al cuadricóptero se obtiene realizando el producto de las tres matrices de rotación definidas.

$$H_I^q = H_2^q \cdot H_1^2 \cdot H_I^1 = \begin{pmatrix} \cos \varphi \cos \theta & \cos \varphi \sin \theta & -\sin \varphi \\ \cos \theta \sin \varphi \sin \psi - \cos \psi \sin \theta & \cos \psi \cos \theta + \sin \varphi \sin \psi \sin \theta & \cos \varphi \sin \psi \\ \sin \psi \sin \theta + \cos \psi \cos \theta \sin \varphi & \cos \psi \sin \varphi \sin \theta - \cos \theta \sin \psi & \cos \varphi \cos \psi \end{pmatrix} \quad (3.13)$$

A su vez, la transformación inversa puede obtenerse multiplicando las coordenadas del sistema S_q por la matriz H_q^I . Dicha matriz puede obtenerse de la siguiente forma:

$$H_q^I = (H_2^q \cdot H_1^2 \cdot H_I^1)^{-1} = (H_I^1)^{-1} \cdot (H_1^2)^{-1} \cdot (H_2^q)^{-1} = H_1^I \cdot H_2^1 \cdot H_q^2 \quad (3.14)$$

Los tres ángulos ψ , φ y θ serán llamados a lo largo del presente trabajo Roll, Pitch y Yaw respectivamente.

3.3. Cinemática

En primer lugar nos enfocaremos en comprender la relación que existe entre la velocidad angular del sistema y las derivadas de los ángulos de Euler. Sea $\vec{\omega}$ la velocidad angular del cuadricóptero (y por ende la del sistema S_q). La expresión de la velocidad angular en el sistema de referencia del cuadricóptero es la siguiente:

$$\vec{\omega} = w_{q_x} \vec{i}_q + w_{q_y} \vec{j}_q + w_{q_z} \vec{k}_q \quad (3.15)$$

Donde w_{q_x} , w_{q_y} y w_{q_z} son las proyecciones ortogonales de la velocidad angular en el sistema S_q . Por como fue construido el sistema de referencia solidario al cuadricóptero y utilizando el teorema de adición de velocidades angulares, se deduce trivialmente que la velocidad angular del cuadricóptero puede escribirse como:

$$\vec{\omega} = \dot{\theta} \vec{k} + \dot{\varphi} \vec{j}_1 + \dot{\psi} \vec{i}_2 \quad (3.16)$$

El vector \vec{i}_2 es invariante respecto de la tercera rotación, es decir que $\vec{i}_2 = \vec{i}_q$. Por otra parte, multiplicando los vectores \vec{k} y \vec{j}_1 por las matrices $H_1^2 \cdot H_2^q$ y H_2^q respectivamente se puede obtener la velocidad angular del cuadricóptero en el sistema de coordenadas referido a él. Operando se obtiene:

$$w_{q_x} \vec{i}_q + w_{q_y} \vec{j}_q + w_{q_z} \vec{k}_q = (\dot{\psi} + \dot{\theta} \sin \varphi) \vec{i}_q + (\dot{\varphi} \cos \psi + \dot{\theta} \cos \varphi \sin \psi) \vec{j}_q + (\dot{\theta} \cos \varphi \cos \psi - \dot{\varphi} \sin \psi) \vec{k}_q \quad (3.17)$$

De esta ecuación se obtienen tres relaciones entre las velocidades angulares respecto de cada eje principal del sistema de coordenadas solidario al cuadricóptero y las derivadas de los ángulos de Euler. Podemos re escribir dicha ecuación de la siguiente forma:

$$\begin{pmatrix} \dot{\psi} \\ \dot{\varphi} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \omega_{q_x} + \omega_{q_z} \tan \varphi \cos \psi + \omega_{q_y} \tan \varphi \sin \psi \\ \omega_{q_y} \cos \psi - \omega_{q_z} \sin \psi \\ \omega_{q_z} \frac{\cos \psi}{\cos \varphi} + \omega_{q_y} \frac{\sin \psi}{\cos \varphi} \end{pmatrix} \quad (3.18)$$

Realizando un razonamiento similar se deduce la relación que existe entre la velocidad del sistema expresada en el marco de referencia inercial con la velocidad

3. Modelo Físico

expresada en el sistema de referencia solidario al cuadricóptero. Sea \vec{r} la posición del centro de masa del cuadricóptero en el sistema S_I . La velocidad en dicho sistema es:

$$\vec{v} = \dot{\vec{r}} = (x\vec{i} + y\vec{j} + z\vec{k})' = \dot{x}\vec{i} + \dot{y}\vec{j} + \dot{z}\vec{k} \quad (3.19)$$

Al igual que con la velocidad angular se puede escribir la velocidad absoluta del cuadricóptero en el sistema S_q . Lo que se tiene es que $\vec{v} = v_{q_x}\vec{i}_q + v_{q_y}\vec{j}_q + v_{q_z}\vec{k}_q$. Donde v_{q_x}, v_{q_y} y v_{q_z} son las proyecciones ortogonales de la velocidad absoluta en el sistema solidario al cuadricóptero. Igualando ambas expresiones de la velocidad se obtiene:

$$\dot{x}\vec{i} + \dot{y}\vec{j} + \dot{z}\vec{k} = v_{q_x}\vec{i}_q + v_{q_y}\vec{j}_q + v_{q_z}\vec{k}_q \quad (3.20)$$

Para transformar las coordenadas de un sistema de referencia al otro alcanza con multiplicar por una de las matrices de cambio de base definidas previamente. En particular nos interesa tener una expresión para las derivadas de la posición en el sistema S_I , por esta razón se multiplica la expresión de la velocidad en el sistema S_q por la matriz H_q^I definida en la ecuación 3.14.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} v_{q_x} \cos \varphi \cos \theta + v_{q_y} (\cos \theta \sin \varphi \sin \psi - \cos \varphi \sin \theta) + v_{q_z} (\sin \psi \sin \theta + \cos \psi \cos \theta \sin \varphi) \\ v_{q_x} \cos \varphi \sin \theta + v_{q_y} (\cos \psi \cos \theta + \sin \theta \sin \varphi \sin \psi) + v_{q_z} (\cos \psi \sin \theta \sin \varphi - \cos \theta \sin \psi) \\ -v_{q_x} \sin \varphi + v_{q_y} \cos \varphi \sin \psi + v_{q_z} \cos \varphi \cos \psi \end{pmatrix} \quad (3.21)$$

Hasta aquí hemos obtenido simplemente relaciones cinemáticas; por un lado entre la velocidad angular del sistema S_q y las derivadas de los ángulos de Euler, por el otro se tiene el vínculo entre las derivadas de la posición expresadas en el sistema S_I y la velocidad absoluta expresada en el sistema S_q y los ángulos de Euler. Sin embargo aún no conocemos cuales son las Fuerzas y Momentos presentes en el sistema, ni que efectos producen sobre el mismo. Dejaremos de lado la cinemática del sistema y pasaremos a analizar la dinámica del mismo.

3.4. Dinámica del Sistema

Existen diversas formas de atacar el problema de la dinámica de un sistema, en particular se puede encarar el problema desde la mecánica analítica o realizando consideraciones energéticas. Se elige trabajar con las ecuaciones cardinales.

3.4.1. Primera Cardinal

La primer cardinal indica que en un sistema de referencia inercial la suma de las fuerzas externas a un objeto es igual a su masa total por su aceleración. Esto se puede escribir:

$$\sum \vec{F}_{ext} = M\vec{a} \quad (3.22)$$

Anteriormente fue expresado el interés de trabajar en el sistema de referencia solidario al cuadricóptero. El vector aceleración se puede obtener derivando la velocidad. Para realizar la derivada de un vector expresado en un sistema móvil puede utilizarse la siguiente formula:

$$\frac{d\vec{A}}{dt} = \frac{d'\vec{A}}{dt} + \vec{\Omega} \times \vec{A} \quad (3.23)$$

En la ecuación anterior $\frac{d}{dt}$ representa la derivada temporal, mientras que $\frac{d'}{dt}$ representa la derivada temporal respecto del sistema móvil. Por otra parte $\vec{\Omega}$ es la velocidad angular del sistema móvil respecto al inercial.

$$\vec{a} = (\dot{v}_{q_x} + v_{q_z}\omega_{q_y} - v_{q_y}\omega_{q_z})\vec{i}_q + (\dot{v}_{q_y} + v_{q_x}\omega_{q_z} - v_{q_z}\omega_{q_x})\vec{j}_q + (\dot{v}_{q_z} + v_{q_y}\omega_{q_x} - v_{q_x}\omega_{q_y})\vec{k}_q \quad (3.24)$$

Operando se puede reescribir la primer cardinal de la siguiente forma:

$$\begin{pmatrix} \dot{v}_{q_x} \\ \dot{v}_{q_y} \\ \dot{v}_{q_z} \end{pmatrix} = \begin{pmatrix} v_{q_y}\omega_{q_z} - v_{q_z}\omega_{q_y} \\ v_{q_z}\omega_{q_x} - v_{q_x}\omega_{q_z} \\ v_{q_x}\omega_{q_y} - v_{q_y}\omega_{q_x} \end{pmatrix} + \frac{\vec{F}_{ext}}{M} \quad (3.25)$$

3.4.2. Segunda Cardinal

La segunda cardinal para un sistema de partículas toma la forma:

$$\frac{d\vec{L}_Q}{dt} = M_Q^{ext} + M\vec{v}_G \times \dot{\vec{r}_Q} \quad (3.26)$$

Donde L_Q y M_Q son el momento angular y el torque respecto del punto Q. En el presente trabajo los movimientos que realizará el cuadricóptero no implicarán velocidades angulares considerables respecto de otro eje distinto de \vec{k}_q (esto se desarrollará en el capítulo 13). Adicionalmente, la distancia entre el centro de masa del sistema y del centro de coordenadas del sistema S_q es pequeña (inferior a los 10cm). Estas consideraciones permiten afirmar que $\vec{V}_G \approx \vec{r}_Q$, podemos entonces obtener una expresión simplificada para la segunda cardinal:

$$\frac{d\vec{L}_O}{dt} = M_O^{ext} \quad (3.27)$$

Por otra parte el momento angular del sistema se compone del momento angular del cuadricóptero y el momento angular de cada motor. Consideraremos el cuadricóptero sin los motores como un primer rígido y los motores como cuatro rígidos independientes.

El momento angular de un rígido respecto a un punto Q del mismo puede calcularse como:

$$\vec{L}_Q = M_i(G_i - Q) \times \vec{V}_Q + \Pi_Q \vec{\Omega}_i \quad (3.28)$$

donde M_i , G_i , Π_Q y $\vec{\Omega}_i$ son la masa de cada rígido, su centro de masa, su tensor de inercia y su velocidad angular respectivamente.

La derivada de dicho momento tiene la forma:

$$\dot{\vec{L}}_{Q'} = M_i(\vec{v}_G - \dot{Q}) \times \vec{v}_Q + M(G - Q) \times \frac{d\vec{v}_Q}{dt} + \frac{d\Pi_Q \vec{\Omega}_i}{dt} \quad (3.29)$$

Nuevamente, asumiendo que la velocidad del centro de masa y la velocidad del origen del sistema de coordenadas S_q son similares, y agregando que las aceleraciones del centro de masa serán de valores pequeños podemos despreciar los términos $M_i(\vec{v}_G - \dot{Q}) \times \vec{v}_Q$ y $M(G - Q) \times \frac{d\vec{v}_Q}{dt}$.

Asumiendo que los cuatro motores son idénticos podemos escribir que:

$$\frac{d\vec{L}_{O'}}{dt} = \frac{d\Pi_{O'_q}^{\{i_q, j_q, k_q\}} \vec{\omega}_q}{dt} \quad (3.30)$$

3. Modelo Físico

Donde $\Pi_{O'_q}^{\{i_q, j_q, k_q\}}$ el tensor de inercia del cuadricóptero sin considerar los motores.

Para el cálculo del momento angular del motor 1 respecto del punto O' supondremos, por simplicidad, que el motor se compone de elementos de masa discretos. En este caso, por definición de momento angular tenemos que:

$$\vec{L}_{O'_{m_1}} = \sum m_i \vec{r}_i' \times \vec{v}_i \quad (3.31)$$

Donde \vec{r}_i' corresponde a la posición de cada elemento de masa (m_i) respecto de O' y \vec{v}_i es la velocidad absoluta de cada elemento de masa. Del Teorema de Roverbal, se puede expresar dicha velocidad como:

$$\vec{v}_i = \vec{v}_{O'} + \vec{\omega}_q \times \vec{r}_i' + \vec{\omega}_1 \times \vec{r}_i'' \quad (3.32)$$

En la ecuación anterior \vec{r}_i'' es la posición de cada elemento de masa expresada desde el centro de masa de cada motor y $\vec{v}_{O'}$ la velocidad del punto O' . Operando con las ecuaciones anteriores se concluye que:

$$\vec{L}_{O'_{m_1}} = M_m d_v \vec{i}_q \times \vec{v}_{O'} + \Pi_{O'_m}^{\{i_q, j_q, k_q\}} \vec{\omega}_q + \Pi_{G_{M_1}}^{\{i_q, j_q, k_q\}} \vec{\omega}_1 \quad (3.33)$$

Donde d_v es la distancia entre el origen del sistema solidario al cuadricóptero y el eje del motor 1. Al considerar los cuatro motores, el primer término del momento angular del motor 1 se anula con el primer término del momento angular del motor 3. Sucede lo mismo con los motores 2 y 4. Calculando en forma análoga los momentos angulares de los restantes motores se obtiene:

$$\vec{L}_{O'_{motores}} = 4 \Pi_{O'_m}^{\{i_q, j_q, k_q\}} \vec{\omega}_q + \Pi_{G_{M_m}}^{\{i_q, j_q, k_q\}} (\vec{\omega}_1 - \vec{\omega}_2 + \vec{\omega}_3 - \vec{\omega}_4) \quad (3.34)$$

Derivando esta expresión y sumándola al resultado obtenido en 3.30 podemos escribir la derivada del momento angular del sistema como:

$$\frac{d\vec{L}_O}{dt} = \Pi_{O'}^{\{i_q, j_q, k_q\}} \frac{d\vec{\omega}_q}{dt} + \Pi_{G_{M_m}}^{\{i_q, j_q, k_q\}} (\dot{\vec{\omega}}_1 - \dot{\vec{\omega}}_2 + \dot{\vec{\omega}}_3 - \dot{\vec{\omega}}_4) + \vec{\omega}_q \times (\Pi_{O'}^{\{i_q, j_q, k_q\}} \vec{\omega}_q + \Pi_{G_{M_m}}^{\{i_q, j_q, k_q\}} (\vec{\omega}_1 - \vec{\omega}_2 + \vec{\omega}_3 - \vec{\omega}_4)) \quad (3.35)$$

A partir del cálculo de esta derivada podemos reescribir la segunda cardinal:

$$\Pi_{O'}^{\{i_q, j_q, k_q\}} \begin{pmatrix} \dot{\omega}_{q_x} \\ \dot{\omega}_{q_y} \\ \dot{\omega}_{q_z} \end{pmatrix} = \begin{pmatrix} \omega_{q_y} \omega_{q_z} (I_{yy} - I_{zz}) - \omega_{q_y} I_{zz_m} (\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ \omega_{q_x} \omega_{q_z} (-I_{xx} + I_{zz}) + \omega_{q_x} I_{zz_m} (\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -I_{zz_m} (\dot{\omega}_1 - \dot{\omega}_2 + \dot{\omega}_3 - \dot{\omega}_4) \end{pmatrix} + M_G^{ext} \quad (3.36)$$

3.5. Fuerzas y momentos involucrados

Para completar el análisis del sistema es fundamental conocer las expresiones de las fuerzas y los momentos involucrados. Por un lado se debe considerar la acción del peso en el sistema. Además debemos conocer con precisión las fuerzas y momentos que introducen las hélices del cuadricóptero ya que son estos los elementos sobre los cuales se pueden realizar las acciones de control. Recordamos que las bajas velocidades a las que el sistema se mueve nos permiten despreciar las fuerzas aerodinámicas, a excepción de aquellas que tienen como origen el giro de las hélices.

3.5.1. Peso

Como ya se aclaró anteriormente, debido a la pequeña variación de la altura, el peso del cuadricóptero será considerado constante y de valor

$$\vec{P} = -Mg\vec{k} \quad (3.37)$$

Insistimos en que nos interesa conocer las fuerzas en el sistema solidario al cuadricóptero. Para expresar el peso en dicho sistema se lo multiplica por la matriz H_I^q . Lo que se obtiene es:

$$\vec{P} = \begin{pmatrix} \sin \varphi \\ -\cos \varphi \sin \psi \\ -\cos \varphi \cos \psi \end{pmatrix} Mg \quad (3.38)$$

Debido a que el centro de masa se encuentra por debajo del origen de coordenadas de la base solidaria al cuadricóptero, el peso tiene un momento. El mismo vale:

$$\vec{M}_P = \vec{d} \times \vec{P} = - \begin{pmatrix} \cos \varphi \sin \psi \\ \sin \varphi \\ 0 \end{pmatrix} dMg \quad (3.39)$$

Donde d es la distancia entre el centro de masa del sistema y el origen del sistema S_q .

3.5.2. Fuerzas y momentos de las hélices

La teoría de fluidos que explica el comportamiento de las hélices es compleja y ampliamente difundida por lo que no es de interés desarrollarla en profundidad en este trabajo. Diversa bibliografía puede ser consultada en donde se explican adecuadamente los fenómenos involucrados en las fuerzas aerodinámicas presentes en una hélice. Sin embargo, sí parece pertinente realizar un breve resumen de dichas fuerzas. Para el presente análisis consideraremos una hélice rotando en sentido antihorario, el razonamiento para una hélice rotando en sentido opuesto es análogo. En la figura 3.3 ² se observan dos vistas de la misma hélice. En la figura 3.3a puede observarse una hoja de la hélice. En la figura 3.3b se representa una vista transversal de la hélice con las fuerzas aerodinámicas involucradas.

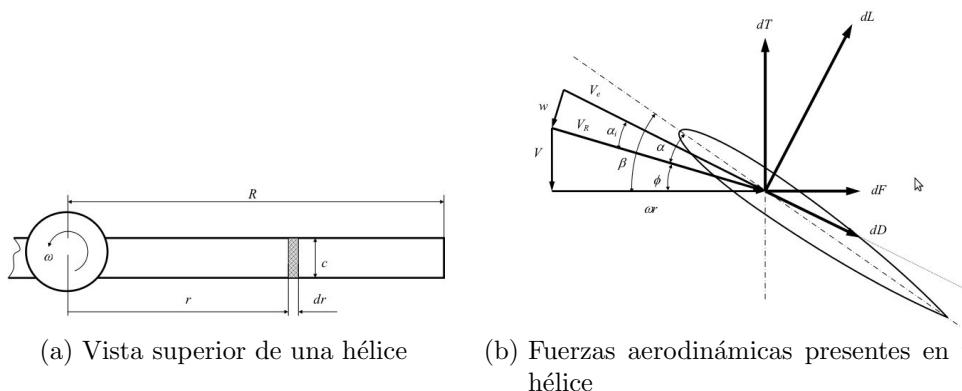


Figura 3.3.: Vistas de una hélice y diagrama de fuerzas aerodinámicas presentes

²Imagen tomada de [2]

3. Modelo Físico

La *Blade Element Theory* (BET) intenta explicar las fuerzas presentes en la hélice considerando en primer lugar las fuerzas en un elemento de área infinitesimal de la hélice. Una vez halladas estas fuerzas se integra sobre el total de la superficie obteniendo así las fuerzas y momentos totales. Como se explica en [2], las fuerzas presentes sobre un elemento de área de la hélice son la fuerza de *lift* y la fuerza de *drag*, dichas fuerzas se encuentran representadas en la figura 3.3 como dL y dD respectivamente. La forma que tienen dichas fuerzas es:

$$dL = \frac{1}{2} \rho_A \omega_p^2 C_L c r^2 dr \quad (3.40)$$

$$dD = \frac{1}{2} \rho_A \omega_p^2 C_D c r^2 dr \quad (3.41)$$

Donde ρ_A es la densidad del aire, ω_p la velocidad angular de la hélice, r la distancia del elemento de hélice al eje de la hélice, c es la longitud promedio de la cuerda de la hélice ³, C_L y C_D son coeficientes adimensionados.

La fuerza infinitesimal de empuje (dT) puede escribirse en función de las fuerzas de *lift* y de *drag* de la siguiente forma:

$$dT = dL \cos \varphi_I - dD \sin \varphi \quad (3.42)$$

Realizando la aproximación que φ es un ángulo pequeño y que la fuerza de *lift* es al menos un orden mayor que la de *drag* se puede afirmar que:

$$dT \approx dL \quad (3.43)$$

El empuje por lo tanto puede calcularse como la integral de dT respecto de r . Considerando que la hélice consta de dos hojas, se obtiene que:

$$T = \frac{1}{3} \rho_A C_L c R_P^3 \omega_p^2 \quad (3.44)$$

donde R_P es el radio radio de la hélice. En todo momento tenemos dos fuerzas (una sobre cada hoja de la hélice) en dirección vertical y hacia arriba. Si nos referimos a la configuración de la figura 3.3b, el momento de las fuerzas dT es hacia la derecha para la hoja considerada, sin embargo para la otra hoja este momento será hacia la izquierda. Se puede concluir entonces que el torque neto que aportan estas fuerzas en el plano perpendicular al eje de rotación de la hélice es nulo.

Intentaremos ahora obtener la resultante de las fuerzas horizontales sobre la hélice y el torque de dichas fuerzas. Comenzaremos por analizar que sucede con las sumas de las fuerzas, para luego proceder a calcular el momento de las mismas. Consideremos ahora la fuerza dF como la que se muestra en la figura 3.3b. Como se observa en la figura dicha fuerza es hacia la derecha. Si consideramos la fuerza horizontal sobre la otra hoja de la hélice, se obtiene una fuerza hacia la izquierda. Por lo tanto la suma de las fuerzas horizontales es nula. Sin embargo los momentos de las fuerzas en una y otra hoja de la hélice no se anulan, el momento que producen ambas es en la dirección de $-\vec{k}_q$. Por lo tanto el momento total será la suma de los momentos infinitesimales en toda la superficie de una hoja de la hélice multiplicado por la cantidad de hojas, es decir 2.

³segmento imaginario que une el borde de ataque con el borde de fuga

En primer lugar escribimos la fuerza horizontal como una composición de la fuerza de *lift* y la fuerza de *drag*.

$$dF = dD \cos \varphi + dL \sin \varphi \approx dD + dL \left(\frac{V}{\omega_p} \right) \quad (3.45)$$

En esta ecuación V y $\omega_p r$ representan la velocidad del flujo de aire en las direcciones vertical y horizontal respectivamente. Operando se obtiene el momento total de la hélice en la dirección entrante:

$$Q = \frac{1}{4} \rho_A c R_P^4 (C_D + K) \omega_p^2 \quad (3.46)$$

En resumen, la teoría BET nos permite afirmar que sobre cada hélice del cuadricóptero que rota en sentido antihorario se aplicará una fuerza en la dirección \vec{k}_q cuyo módulo se expresó en la ecuación 3.44 y un momento en la dirección $-\vec{k}_q$ cuyo módulo es lo expresado en la ecuación 3.46. Para una hélice rotando en sentido horario se obtiene exactamente los mismos resultados excepto que el momento es en sentido opuesto.

Hasta aquí sabemos que sucede las fuerzas y momentos aplicados sobre una hélice. Nos concentraremos ahora en estudiar como influyen estas fuerzas y momentos en el cuadricóptero. Para lo que sigue continuaremos trabajando con las convenciones adoptadas en 3.1.

A partir de lo estudiado anteriormente se deduce trivialmente que el empuje de las hélices puede expresarse en el sistema S_q como:

$$\sum_{i=1}^{i=4} \vec{T}_i = \sum_{i=1}^{i=4} T_i \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.47)$$

Se determinó que las fuerzas de empuje no introducen un momento neto en el eje de las mismas. Sin embargo no debemos perder de vista que la segunda cardinal fue planteada en el centro de masa del sistema, por lo tanto respecto de dicho punto las fuerzas sí introducen un momento que puede calcularse como $M_1 = L \vec{x}' \times T_1 \vec{k}_q$ para el motor 1. L es la distancia del centro de masa del cuadricóptero al eje del motor 1. La expresión del momento es análoga para los restantes motores.

Debemos considerar además el momento obtenido para cada hélice en la dirección \vec{k}_q . Cabe recordar que se dedujo que para hélices rotando en sentido anti-horario se tiene un momento negativo, mientras que para una hélice rotando en sentido horario el momento es positivo. Realizando estas consideraciones es posible afirmar que la suma total de los momentos es:

$$M_G^{ext} = \begin{pmatrix} L(T2 - T4) \\ L(T3 - T1) \\ -Q_1 + Q2 - Q3 + Q4 \end{pmatrix} \quad (3.48)$$

En base al estudio realizado se podrían conocer dichas relaciones calculando los parámetros que dependen de la geometría de la hélice, sin embargo este estudio resulta tedioso y la mayor parte de los métodos existentes para determinar dichos parámetros con buena precisión son destructivos. Por lo tanto se optó por obtener

3. Modelo Físico

dichas respuestas en forma experimental. El proceso detallado puede consultarse en el capítulo 6. En dichos experimentos se obtuvo que:

$$T = 4,6016 \times 10^{-5} \omega^2 - 1,0380 \times 10^{-3} \omega \quad (3.49)$$

$$Q = 3,4734 \times 10^{-6} \omega^2 - 1,3205 \times 10^{-4} \omega \quad (3.50)$$

Estos resultados parecen adecuados respecto de lo desarrollado teóricamente ya que ambas respuestas son cuadráticas.

3.6. Modelo en variables de Estado

Luego de realizados los estudios sobre la cinemática y dinámica del sistema y luego de comprender cabalmente las fuerzas y momentos involucrados se procede a construir el modelo en variables de estado.

Se debe aclarar que la elección realizada del vector de estados se debe exclusivamente a la conveniencia práctica que se encuentra al trabajar con las variables expresadas en el sistema del cuadricóptero. Esta conveniencia se ve reflejada en dos aspectos, por un lado en las simplificaciones que introduce trabajar con estas variables en el marco de un desarrollo teórico de las ecuaciones que gobiernan al sistema, y por otra parte, al disponer de sensores montados sobre el cuadricóptero, los mismos medirán las velocidades angulares y las aceleraciones lineales en el sistema S_q . Bajo esta elección parece razonable escoger el vector de estados de la siguiente manera:

$$X = \{x, y, z, \theta, \varphi, \psi, v_{q_x}, v_{q_y}, v_{q_z}, \omega_{q_x}, \omega_{q_y}, \omega_{q_z}\} \quad (3.51)$$

donde “ x ”, “ y ” y “ z ” representan la posición en el espacio referenciado al sistema inercial S_I , los ángulos “ θ ”, “ φ ”, “ ψ ” son los ángulos de Euler mostrados anteriormente, “ v_{q_x} ”, “ v_{q_y} ” y “ v_{q_z} ” las velocidades lineales en los 3 ejes referenciados a S_I y “ ω_{q_x} ”, “ ω_{q_y} ” y “ ω_{q_z} ” las velocidades angulares en los 3 ejes referenciados al cuadricóptero.

Las ecuaciones desarrolladas hasta ahora son las que gobiernan el comportamiento mecánico del sistema y son las que serán utilizadas para el desarrollo del simulador. Dichas ecuaciones dependen tanto de la velocidad angular de las hélices como de sus derivadas. Para realizar el control resulta más sencillo poder trabajar sin dichas derivadas. La razón es que las velocidades angulares y sus derivadas no son entradas independientes. Por lo tanto no se pueden imponer valores a unas sin considerar el comportamiento de las otras. En este sentido se realiza última aproximación. El término de las derivadas de las velocidades angulares se encuentra presente exclusivamente en la ecuación 3.36. Dicho término se encuentra multiplicado por el momento de inercia de los motores. En el capítulo 6 se obtiene el orden de la variación de la velocidad angular de los motores. Adelantando el resultado de esta variación de la velocidad angular y el valor del término del tensor de inercia de los motores (ver anexo C), se puede despreciar el término de las derivadas de las velocidades angulares respecto al término asociado a los torques en la ecuación 3.36.

Luego de realizada esta simplificación tenemos que la entrada está compuesta por el vector:

$$u = \{\omega_1, \omega_2, \omega_3, \omega_4\} \quad (3.52)$$

Cabe aclarar que esta última simplificación se toma simplemente para el desarrollo del control, pero la misma no se realiza en el simulador (ver capítulo 4), de modo de poder verificar que efectivamente es despreciable.

El modelo en variables de estado resulta:

$$\begin{aligned} \dot{x} &= v_{q_x} \cos \varphi \cos \theta + v_{q_y} (\cos \theta \sin \varphi \sin \psi - \cos \varphi \sin \theta) + v_{q_z} (\sin \psi \sin \theta + \cos \psi \cos \theta \sin \varphi) \\ \dot{y} &= v_{q_x} \cos \varphi \sin \theta + v_{q_y} (\cos \psi \cos \theta + \sin \theta \sin \varphi \sin \psi) + v_{q_z} (\cos \psi \sin \theta \sin \varphi - \cos \theta \sin \psi) \\ \dot{z} &= -v_{q_x} \sin \varphi + v_{q_y} \cos \varphi \sin \psi + v_{q_z} \cos \varphi \cos \psi \\ \dot{\psi} &= \omega_{q_x} + \omega_{q_z} \tan \varphi \cos \psi + \omega_{q_y} \tan \varphi \sin \psi \\ \dot{\varphi} &= \omega_{q_y} \cos \psi - \omega_{q_z} \sin \psi \\ \dot{\theta} &= \omega_{q_z} \frac{\cos \psi}{\cos \varphi} + \omega_{q_y} \frac{\sin \psi}{\cos \varphi} \\ \dot{v}_{q_x} &= v_{q_y} \omega_{q_z} - v_{q_z} \omega_{q_y} + g \sin \varphi \\ \dot{v}_{q_y} &= v_{q_z} \omega_{q_x} - v_{q_x} \omega_{q_z} - g \cos \varphi \sin \psi \\ \dot{v}_{q_z} &= v_{q_x} \omega_{q_y} - v_{q_y} \omega_{q_x} - g \cos \varphi \cos \psi + \frac{1}{M} \sum_{i=1}^4 T_i \\ \dot{\omega}_{q_x} &= \frac{\omega_{q_y} \omega_{q_z} (I_{yy} - I_{zz}) + \omega_{q_y} I_{zzm} (\omega_1 - \omega_2 + \omega_3 - \omega_4) - dMg \cos \varphi \sin \psi}{I_{xx}} + \frac{L(T_2 - T_4)}{I_{xx}} \\ \dot{\omega}_{q_y} &= \frac{\omega_{q_x} \omega_{q_z} (-I_{xx} + I_{zz}) + \omega_{q_x} I_{zzm} (\omega_1 - \omega_2 + \omega_3 - \omega_4) - dMg \sin \varphi}{I_{yy}} + \frac{L(T_3 - T_1)}{I_{yy}} \\ \dot{\omega}_{q_z} &= \frac{-Q_1 + Q_2 - Q_3 + Q_4}{I_{zz}} \end{aligned} \quad (3.53)$$

Las ecuaciones desarrolladas son similares a las consideradas por otros autores. En el trabajo propuesto por [3] se describe inicialmente un modelo similar al aquí considerado con la excepción que el centro de masa del sistema se considera en el centro del origen de coordenadas del sistema móvil. Esta consideración es fundamental y ya que la ubicación del centro de masa por debajo del plano en el cual se realizan las fuerzas aporta una estabilidad intrínseca al sistema. Adicionalmente se puede agregar que la matriz de control determinada a través de la técnica LQR (ver capítulo 14) que surge de considerar o no el centro de masa difieren sensiblemente y también los resultados obtenidos. Al no considerar el centro de masa se obtiene una realimentación proporcional mayor en los ángulos de pitch y de roll desestabilizando el sistema, mientras que los resultados obtenidos al considerar el centro de masa son aceptables.

En [4] se propone un modelo en el cual inicialmente se consideran las fuerzas y torques aerodinámicos además de los causados por las hélices. A la hora de diseñar el control del sistema se desprecian dichos términos. La comparación con la bibliografía existente permite asegurar que el modelo desarrollado abarca la gran mayoría de las consideraciones fundamentales a realizar.

CAPÍTULO 4

SIMULACIONES DEL MODELO FÍSICO

Luego de desarrollado un modelo físico se construye un entorno para realizar simulaciones. En primer lugar resulta fundamental para comprobar que el modelo realizado se comporta acorde a lo que uno espera a priori del sistema. Para este tipo de pruebas se trabajará con las situaciones más sencillas en las cuales se puede calcular la trayectoria trivialmente. El segundo objetivo del simulador es poder conocer el comportamiento de nuestro sistema frente a algunas acciones de control determinadas. Por último, el simulador será clave para "pre evaluar" y mejorar los algoritmos de control desarrollados. Previo a testear con el sistema real y a fin de evitar daños sobre el mismo, se deben verificar dichos algoritmos en el simulador. Por los motivos expresados es necesario que el simulador represente fielmente el modelo físico y se comporte acorde a la realidad.

En el anexo I se encuentra disponible un manual de usuario del simulador.

4.1. Estructura del Simulador

El simulador fue desarrollado gracias a la herramienta *Simulink* de *Matlab*, el mismo se compone de dos partes fundamentales. La primera es el lazo abierto, es decir las ecuaciones que gobiernan al cuadricóptero , donde se consideran como entradas las velocidades angulares del sistema sobre las cuales realizaremos las acciones de control y como salidas tenemos el vector de estados del sistema en todos los instantes desde el tiempo inicial establecido en la simulación hasta el tiempo final. La otra parte se encarga de simular las acciones de control.

4. Simulaciones del modelo físico

4.1.1. Lazo Abierto

Las ecuaciones que gobiernan al sistema (Ecuaciones 3.53¹) son continuas y pueden representarse como:

$$\dot{X} = f(X, u, t) \quad (4.1)$$

Donde X es el vector de estados del sistema definido en el capítulo 3, u el vector de entradas del sistema y t el tiempo. Para la simulación se utiliza el método de resolución ODE45 (Dormand-Prince). Este método de resolución de ecuaciones diferenciales, se utiliza un paso de resolución variable: el paso máximo y mínimo es de 10ms y 1ms respectivamente. La estructura que se eligió para desarrollar esta sección se corresponde con el camino que se recorrió para determinar el modelo físico. El lazo abierto consta de tres bloques principales. En primer lugar tenemos un bloque encargado de generar las fuerzas y momentos a partir de las velocidades angulares de las hélices. Luego tenemos un bloque que se encarga de resolver la dinámica del sistema y un tercer bloque encargado de la cinemática. En la figura 4.1 se observa la estructura global del lazo abierto. En la figura 4.2 se observa una captura de pantalla que representa la vista general de la parte encargada de simular el lazo abierto.

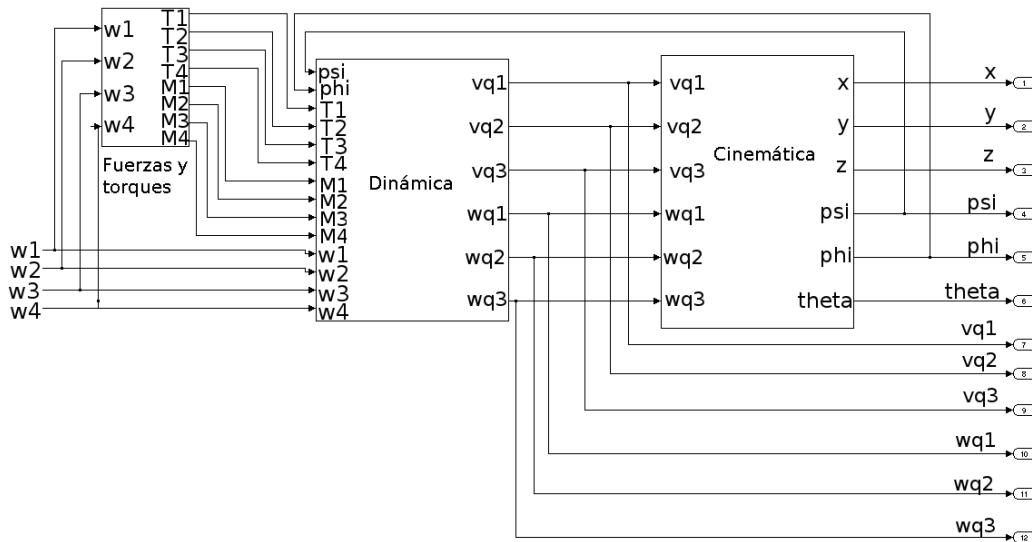


Figura 4.1.: Bloque de lazo abierto

Cinemática

En la figura 4.3a se puede observar un diagrama de bloques de la parte del sistema que transforma las velocidades lineales y angulares en posiciones y ángulos de Euler. Se distinguen dos sub-bloques principales, uno encargado de devolver la posición y otro encargado de devolver los ángulos de Euler

¹Se agrega también la dependencia de la última ecuación del modelo con las derivadas de las velocidades angulares del sistema

4.1. Estructura del Simulador

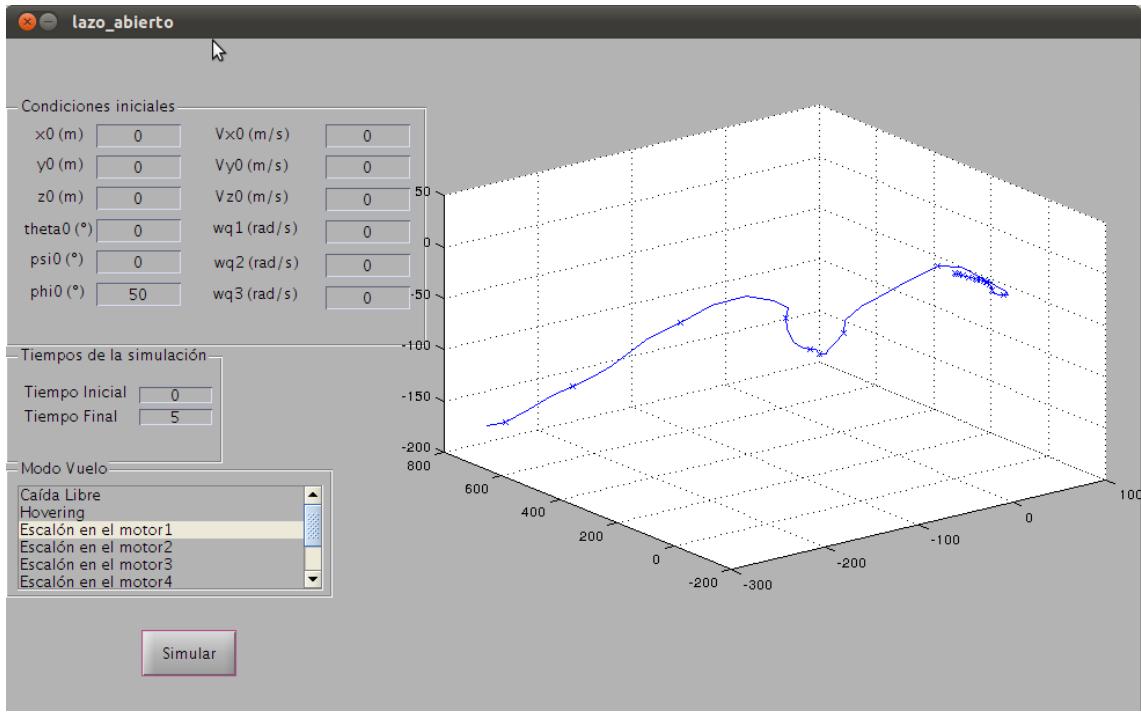


Figura 4.2.: Interfaz del simulador de lazo abierto

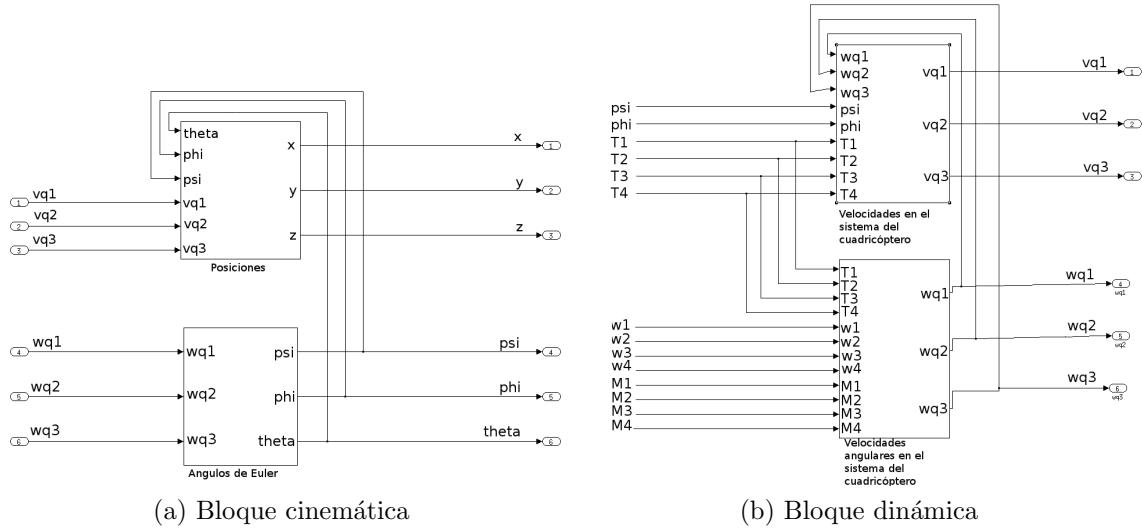


Figura 4.3.: Bloques en mayor detalle

Dinámica

Al igual que el bloque anterior, se divide este en dos sub-bloques más (ver figura 4.3b): el que devuelve las velocidades angulares, y el que devuelve las velocidades lineales.

4.1.2. Lazo cerrado

El lazo cerrado simula las acciones de control del sistema. Además de las ecuaciones del lazo abierto (ecuación 4.1) se agrega la realimentación de las variables de estado ($x, y, z, \psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}, \omega_{qx}, \omega_{qy}, \omega_{qz}$) y de las integrales del error entre

4. Simulaciones del modelo físico

x, y, z, θ y los respectivos setpoint. Llamamos X_I al vector compuesto por dichas integrales. La ecuación de realimentación es:

$$u(t) = u_{setpoint}(t) - K \begin{pmatrix} X(t) - X_{setpoint} \\ X_I(t) \end{pmatrix} \quad (4.2)$$

Donde K es la matriz de realimentación determinada para cada trayectoria. Como se explica en I, desde la interfaz gráfica se puede seleccionar el tipo de trayectoria² que se desea realizar y los valores de las variables de estado con las que se desea realizar dicha trayectoria³. Las velocidades angulares objetivo para cada motor serán determinadas a partir de la información anterior. Al igual que en el simulador de lazo abierto, se tiene la posibilidad de establecer tanto el tiempo inicial de la simulación como las condiciones iniciales. La matriz de realimentación quedará determinada por la trayectoria.

Tiempo discreto

Como se explicó en la sección 1 el control será realizado con un microprocesador, esto implica que las acciones de control no podrán ser modificadas en forma continua, cada cierto período se indicará un nuevo valor de velocidad angular para cada motor. Del mismo modo, no se tiene conocimiento del estado en todo instante sino que se tienen datos cada un cierto intervalo de tiempo (no necesariamente igual al período con el cual se actúa sobre los motores). Estas consideraciones hacen necesaria una modificación en el sistema que hasta ahora había sido considerado como continuo, es necesario convertir el sistema de tiempo continuo desarrollado a un sistema de tiempo discreto. Dicha modificación se logra sustituyendo los bloques integradores y derivadores que formaban parte del sistema por integradores y derivadores discretos con un período de muestreo que también puede ser impuesto desde la interfaz gráfica. La simulación se realiza mediante el método *discrete* que ofrece *matlab*. El paso de la simulación es variable y el valor máximo que toma dicho paso es 10ms. En la figura 4.4 se puede observar la misma trayectoria para tres tiempos de muestreo diferentes. En dicha trayectoria se muestra la subida del cuadricóptero desde la altura inicial $z = 0m$ hasta $z = 3m$.

Se observa claramente un deterioro de la performance en la subida al aumentar el período de muestreo y de acción sobre los motores. Por dicho motivo es importante incluir esta variable a la hora de realizar diversas simulaciones ya que el sistema real debe realizar una gran cantidad de operaciones y si bien su capacidad es considerable no es infinita. Esto puede producir que se tenga acotado inferiormente el período de muestreo.

Restricciones físicas

Se agregó una restricción sobre la velocidad angular de los motores ya que esta no puede tener cualquier valor, para esto se agregaron los bloques de saturación a

²Por tipo de trayectoria nos referimos a círculos, rectas o hovering, mientras que por trayectoria nos referimos a un tipo de trayectoria y los valores objetivo de las variables de estado

³Evidentemente existen restricciones a la hora de elegir las variables de estado, a modo de ejemplo no seremos capaces de controlar una trayectoria en linea recta si las velocidades angulares no son nulas

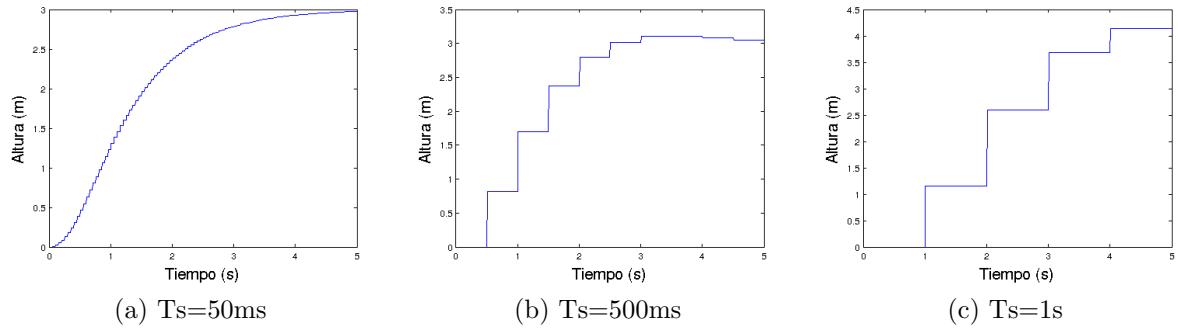


Figura 4.4.: Trayectoria de ascenso del cuadricóptero desde $z = 0$ a $z = 3$

la entrada del subsistema que representa la dinámica del cuadricóptero.

No se consideraron restricciones adicionales sobre las velocidades ni sobre las aceleraciones del sistema ya que las mismas no fueron caracterizadas. Asimismo tampoco se imponen restricciones en los valores de setpoint del sistema. Estas características son una debilidad del simulador, sin embargo para realizar simulaciones a bajas velocidades el mismo es adecuado.

Ruido y no idealidades

Se desea incluir la posibilidad de agregar ruido a los estados medidos y perturbaciones en las velocidades angulares de los motores. Las medidas que se obtienen de los sensores no son exactas, por dicho motivo la posibilidad de agregar ruido es muy interesante de modo de testear la robustez del controlador implementado. A cada variable de estado se le puede agregar un ruido de la forma:

$$\eta(t) = \eta_A \cos(\omega_\eta t) + \varepsilon(\mu, \sigma) \quad (4.3)$$

La amplitud y la frecuencia de la componente sinusoidal del ruido y el valor medio y la desviación estándar del ruido gaussiano pueden establecerse independientemente para cada variable.

Asimismo, la velocidad angular de los motores no es exactamente la que se espera de acuerdo a la caracterización de los motores realizada (ver capítulo 6), por el contrario, se producen variaciones en la velocidad angular de los mismos dada una velocidad angular objetivo. Se agrega la posiblidad de agregar a la velocidad angular de los motores un ruido gaussiano.

Modificación para trayectorias circulares

Como se explica en el capítulo 13, para tratar las trayectorias circulares en el plano horizontal es necesario introducir un cambio de variables en el sistema. Este cambio de variables consiste en expresar la posición del cuadricóptero en el sistema S_q solidario a él y considerando como origen el centro de la trayectoria circular, en lugar de expresar la posición en el sistema cartesiano inercial. Esta modificación implica realizar un cambio en el modelo para trabajar con dichas trayectorias, simplemente se agrega una matriz de rotación para trabajar con la posición expresada

4. Simulaciones del modelo físico

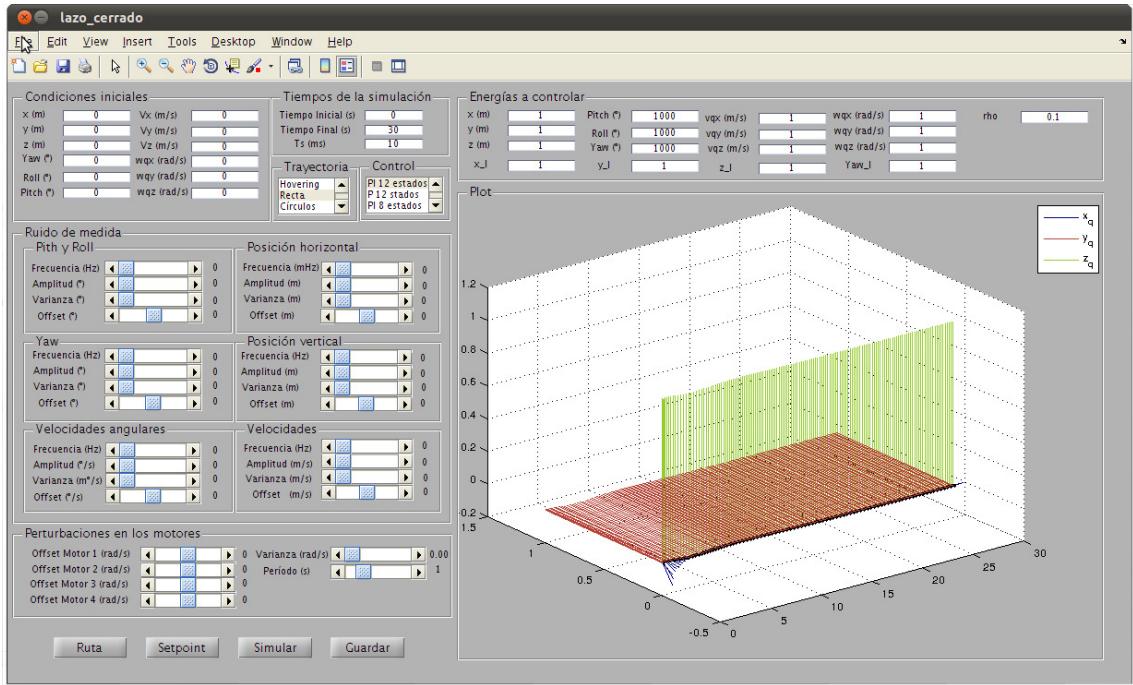


Figura 4.5.: Interfaz del simulador de lazo cerrado

en el sistema solidario al cuadricóptero.

4.2. Simulaciones

En esta sección procederemos a realizar algunas simulaciones a fin de verificar que los resultados arrojados se corresponden con lo esperado a priori.

Caída libre con velocidad inicial nula

Se simula una caída libre con condiciones iniciales nulas excepto la altura que se fija a 100m. El tiempo de simulación considerado es de tres segundos. En la figura 4.6a se observa la trayectoria obtenida. En este caso se grafica uno de cada veinte puntos obtenidos. La misma se corresponde con lo que se espera a priori: puntos equiespaciados en el tiempo se encuentran cada vez más apartados a medida que transcurre el tiempo. En la figura 4.6b se representa la altura en función del tiempo. La altura final es $z_f = 55,855m$. La altura en una caída libre puede calcularse como $z(t) = -\frac{gt^2}{2} + Z_0$. En este caso se obtiene $z(3) = 55,855m$.

Caída libre con velocidad inicial

Se realiza la misma simulación que en la sección anterior excepto que se inicia el vuelo con $V_0 = 1ms^{-1}\vec{i} + 3ms^{-1}\vec{k}$. Los resultados de la simulación pueden encontrarse expresados graficamente en la figura 4.7. La coordenada de la posición según \vec{i} aumenta con el tiempo con pendiente igual a la velocidad inicial. La altura cumple que $z(t) = -\frac{gt^2}{2} + 3ms^{-1}t + Z_0$. Por lo tanto la misma aumenta hasta un tiempo $t_{max}/\dot{z}(t) = 0$. Lo cual implica que $t_{max} = \frac{3ms^{-1}}{g} \approx 0,31s$. Por otra parte tiempo para el cual se da el máximo en la simulación es $t_{max_{sim}} = 0,306s$. Considerando que

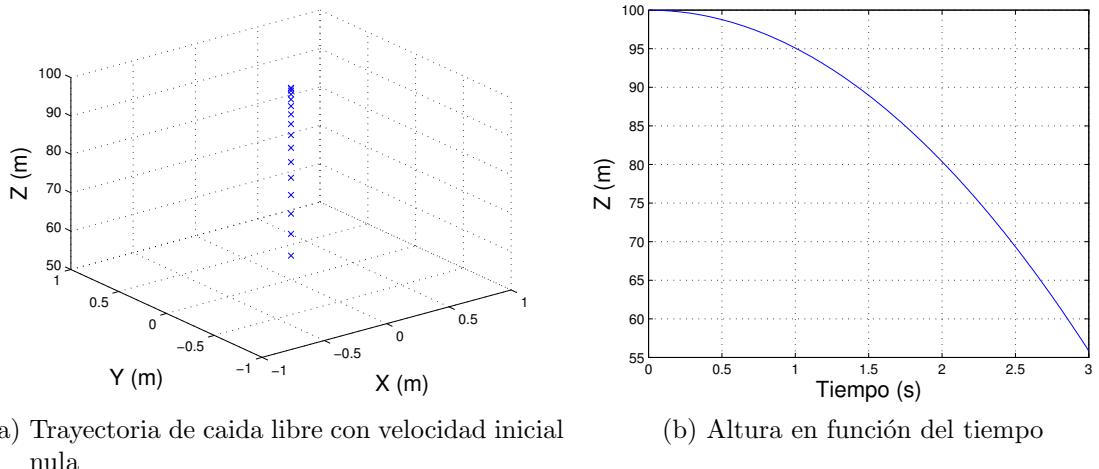


Figura 4.6.: Caida libre con velocidad inicial nula

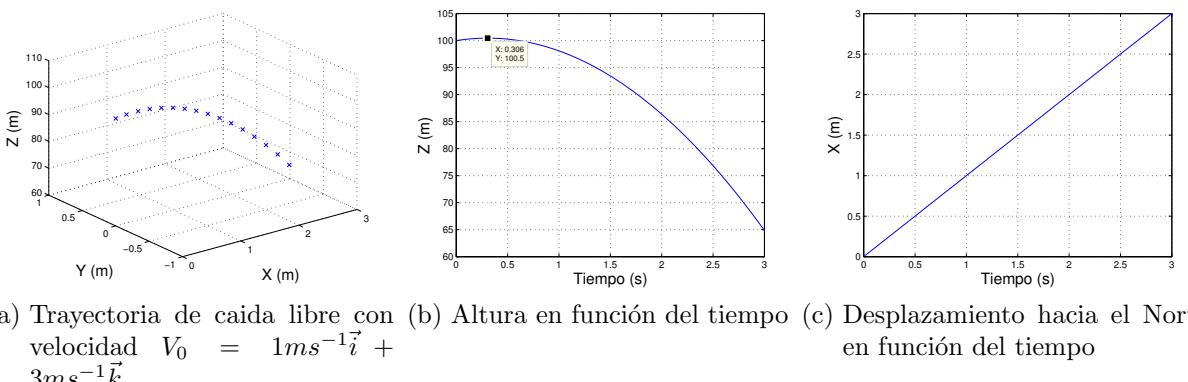


Figura 4.7.: Caida libre con velocidad inicial no nula

las simulaciones se realizan con un paso variable el cual puede ser de hasta 0,01s se considera un resultado aceptable. El siguiente valor para el tiempo simulado es 0,316s, por lo tanto es razonable que dicho valor se presente en $t_{max_{sim}}$. La altura máxima teórica vale $z_{max_{teo}} = 100,459m$, la altura máxima obtenida a través de la simulación es igual⁴. A partir de este punto tenemos una caída libre como la que ya estudiamos en el caso anterior. Las alturas finales, tanto en la simulación como en la teoría son 64,885m.

Condición de Hovering

Se aplica una fuerza constante en los cuatro motores tal que la resultante es igual al peso. Las condiciones iniciales son todas nulas, excepto $Z_0 = 10m$. Se logra el equilibrio mecánico. Todas las variables permanecen constantes. Se simula durante diez segundos

⁴Considerando tres cifras después de la coma

4. Simulaciones del modelo físico

Escalón en los cuatro motores

Con condiciones iniciales nulas, en condición de hovering se aumenta la velocidad angular de los motores en 100rads^{-1} en $t = 5\text{s}$. Se simula durante diez segundos. En la figura 4.8 se presentan gráficamente los resultados obtenidos en la simulación. La altura máxima alcanzada por el cuadricóptero en la simulación es de $93,61\text{m}$ mientras que en la teoría dicha altura es de $93,78\text{m}$. Nuevamente la diferencia entre el valor simulado y el esperado difieren de manera despreciable y es atribuible a aproximaciones realizadas.

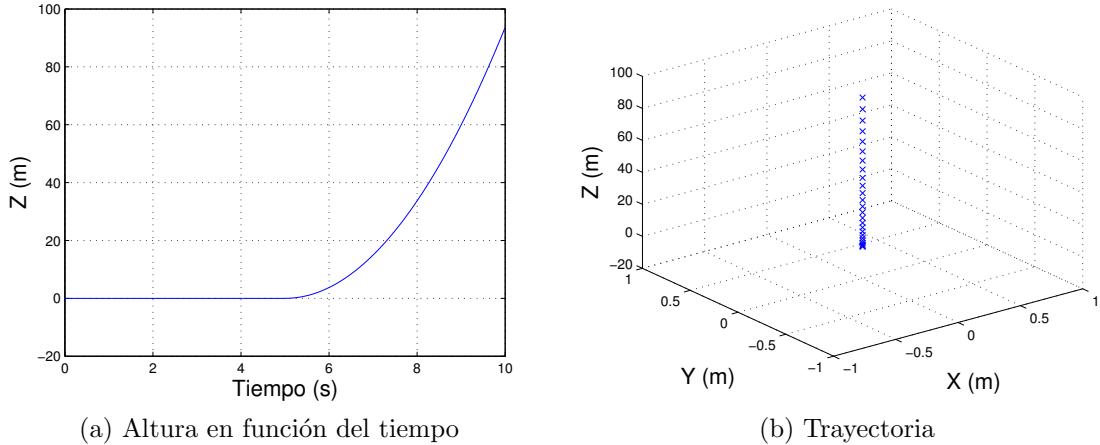


Figura 4.8.: Escalón en los cuatro motores

Cambio en el Pitch y el Roll

Con condiciones iniciales nulas excepto la altura fijada en $z = 10$, se realiza un giro según \vec{i}_q y según \vec{j}_q . En el primer caso se aumenta la velocidad del motor 2 en tres radianes por segundo y se disminuye la velocidad del motor 4 en la misma cantidad. Para la segunda simulación se aumenta en tres radianes por segundo la velocidad del motor 3 y se disminuye en la misma cantidad la velocidad del motor 1. Los resultados obtenidos se muestran en la figura 4.9.

En estos dos casos también se obtiene un resultado acorde a lo esperado. En ambos casos el ángulo aumenta de forma cuadrática en la zona donde el ángulo es inferior a los dos grados, en esta zona puede despreciarse el momento realizado por el peso, por lo tanto se tiene una aceleración angular constante, lo cual explica el comportamiento del ángulo. Para ángulos superiores dicho torque del peso comienza a ser considerable respecto del torque producido por los motores y se observa que el ángulo crece con menor pendiente.

Giro según el eje \vec{k}

En las mismas condiciones que la simulación anterior, en el tiempo $t = 5\text{s}$ se aumenta repentinamente la velocidad angular de los motores que rotan en sentido horario con un valor tal que la fuerza de cada uno de esos motores aumenta en 1N . Para los motores que rotan en sentido anti-horario se disminuye la velocidad angular de forma que la fuerza de cada uno de ellos disminuye 1N . Estas velocidades son $349,88\text{rads}^{-1}$ y $278,09\text{rads}^{-1}$ respectivamente. La fuerza neta permanece constante

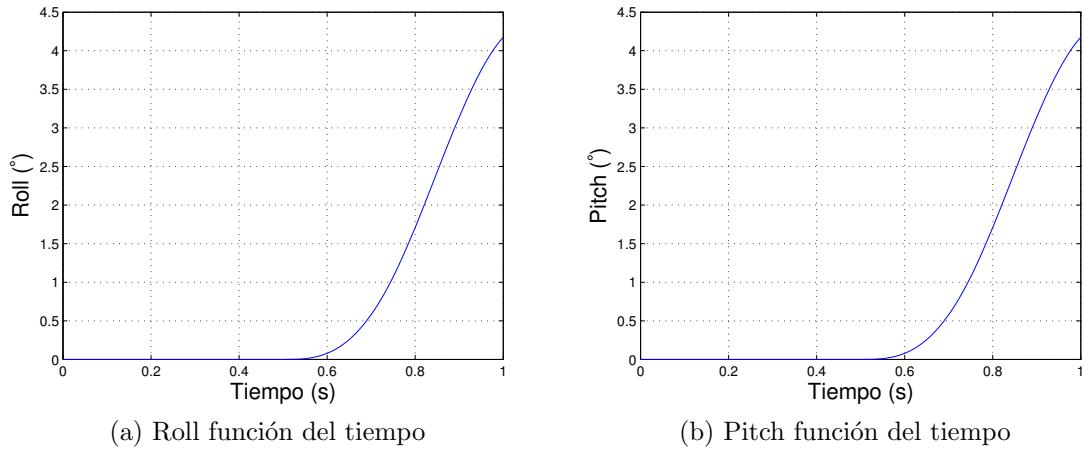
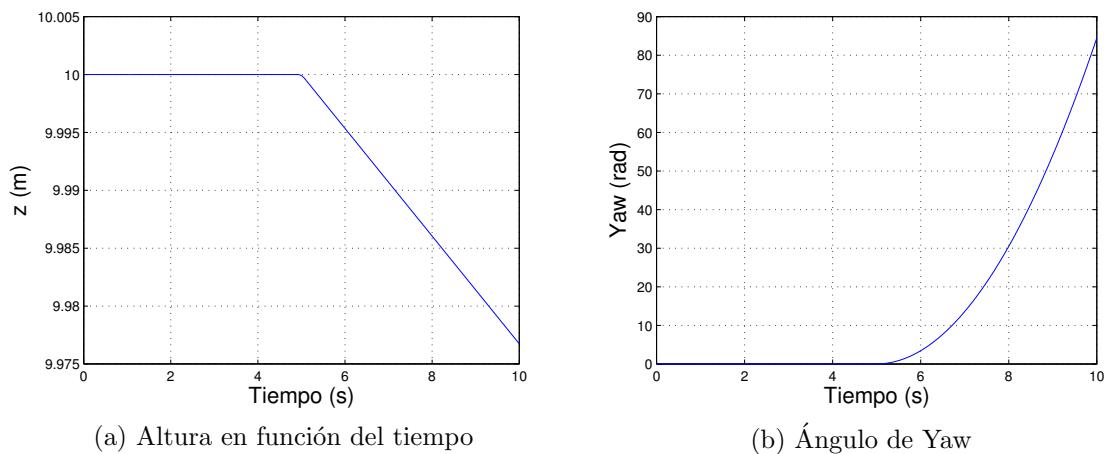


Figura 4.9.: Aumento en los ángulos de Pitch y Roll

y el momento según los versores \vec{i}_q y \vec{j}_q es nulo. Sin embargo aparece un torque positivo según el versor \vec{k}_q .

En la figura 4.10a se presenta la altura en función del tiempo. La misma debería permanecer constante, sin embargo se observa una pequeña diferencia en la altura de $2,3cm$. Esta diferencia es atribuida a la aproximación realizada al calcular las velocidades con las cuales deben girar los motores. Por otra parte en la figura 4.10b se observa como el ángulo aumenta hasta el valor de $84,43rad$. El torque neto vale $Q = 0,29Nm$. Por lo tanto en 5 segundos se debe rotar un ángulo de $\theta_f = 84,16rad$. Nuevamente se percibe una pequeña diferencia entre el valor teórico y el simulado, pero dicho error es aceptable.

Figura 4.10.: Giro según el \vec{k}_q

Hasta aquí hemos testeado el simulador en situaciones conocidas. Nos concentramos en analizar la caida libre y las cuatro acciones de control básicas que se pueden realizar descriptas en 1. De acuerdo a las pruebas realizadas puede afirmarse que su funcionamiento es el adecuado ya que en ninguna prueba se obtuvieron errores considerables. Sin embargo, es fundamental aclarar que hasta aquí no es posible

4. Simulaciones del modelo físico

afirmar que el modelado del sistema sea adecuado, lo único que puede concluirse es que el simulador representa fielmente las ecuaciones que han sido deducidas hasta el momento. Un error en las ecuaciones no se reflejará hasta el momento de testear el cuadricóptero. El trabajo realizado a la hora del modelado y la comparación con diversas bibliografías ([3] y [4] entre otras) nos permite a esta altura estar convencidos de que dichas ecuaciones son adecuadas para modelar el sistema.

Parte III.

Motores

CAPÍTULO 5

INGENIERÍA INVERSA

El cuadricóptero adquirido resuelve la comunicación entre los controladores de los motores (**ESCs**) y el microprocesador mediante el protocolo **i²c**.

Para el presente proyecto resulta de vital importancia conocer dicho protocolo, ya que se utilizará otro microprocesador que deberá comandar a esos mismos ESCs, supliendo el trabajo del anterior. Es entonces imprescindible conocer al detalle el funcionamiento de este protocolo, para luego poder reproducirlo.

Dado que no se cuenta con la mínima colaboración de los fabricantes y la información es completamente privativa, es necesario realizar un proceso de ingeniería inversa para poder analizar, decodificar, entender y reproducir el protocolo existente. Dicho proceso de ingeniería inversa se realiza utilizando el hardware existente del cuadricóptero comercial adquirido y un analizador lógico¹ que es capaz de leer e interpretar las líneas del bus i^2c sin intervenir en las mismas.

Antes de presentar los resultados obtenidos en el proceso, se realiza una breve introducción al protocolo i^2c . En la figura 5.1 se presenta la definición de los ejes a utilizar, lo cual será de utilidad más adelante.



Figura 5.1.: Definición de ejes

¹ChronoVu

5.1. Introducción al protocolo i^2c

El bus i^2c es un bus de comunicaciones serie. Su nombre viene de *Inter-Integrated Circuit* (Circuitos Inter-Integrados).

Utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. Además será necesaria una tercera línea de tierra, como referencia.

En la imagen 5.2² se muestra un diagrama de un circuito equivalente simplificado de la conexión i^2c entre 2 dispositivos.

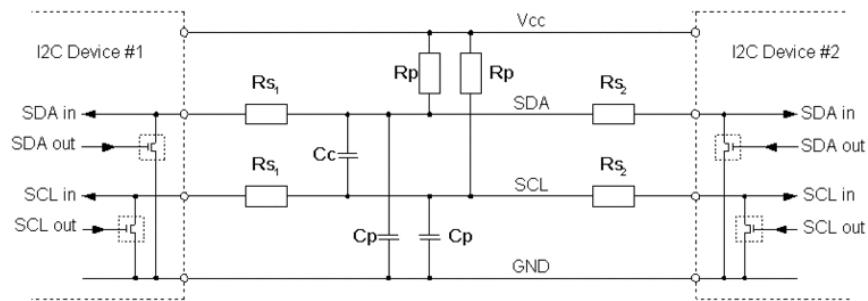


Figura 5.2.: Conexión i^2c .

donde:

- **Vcc**: Voltaje de entrada, típicamente varía entre 1.2 V y 5.5 V
- **GND**: Tierra común
- **SDA**: Línea serial de datos
- **SCL**: Línea serial de reloj
- **R_p** : Resistencia de "Pull-up"
- **R_s** : Resistencia serie
- **C_p** : Capacitancia del cable
- **C_c** : Capacitancia de canal cruzado

Las líneas **SDA** y **SCL** son de drenador abierto, lo que significa que tanto el maestro como los esclavos solamente pueden conducir a nivel bajo estas líneas, o dejarlas abiertas. Si ningún dispositivo i^2c está conduciendo hacia abajo la línea, la resistencia de *pull-up* R_p se encarga de llevar la línea a V_{cc} .

El reloj siempre es generado por el maestro. El bus de datos debe mantenerse estable mientras el reloj está en nivel alto ("1" lógico) y se le permite cambiar de nivel mientras el reloj está en nivel bajo ("0" lógico). Se toma como dato válido el valor del bus de datos cuando el reloj está en "0".

Cada dispositivo tiene asignada una dirección que lo identifica. Típicamente para establecer una comunicación el maestro envía una secuencia de comienzo de conexión, seguida de la dirección del esclavo con el cual desea comunicarse. Seguidamente

²Imagen tomada de www.i2c-bus.org

el maestro envía un bit que determina si la acción que desea realizar es escritura o lectura, a lo que el esclavo correspondiente responde con un bit de *acknowledge* (**Ack**). Luego el maestro envía la dirección de memoria interna del esclavo donde debe ser almacenada la información enviada, y por último envía los datos. Para finalizar la conexión, el maestro envía una secuencia de fin de conexión.

5.2. Pruebas en régimen

- Materiales

- *Cuadricóptero*: transmisor, receptor, CPU y ESCs
- *Analizador lógico (“sniffer”)*: Chrono Vu
- *PC*

- Procedimiento

- La forma de operar es enviar comandos conocidos con el control remoto y analizar el flujo de datos en las líneas del bus.
- En particular se analizará el funcionamiento en régimen, durante el arranque y durante el frenado.

La comunicación entre la CPU y los ESCs es mediante el protocolo i^2c , y es precisamente allí donde interviene el analizador lógico, “olfateando” todo lo que pasa por el bus. En la figura 5.3a se muestra un diagrama de bloques del proceso descripto, y en la figura 5.3b su implementación.

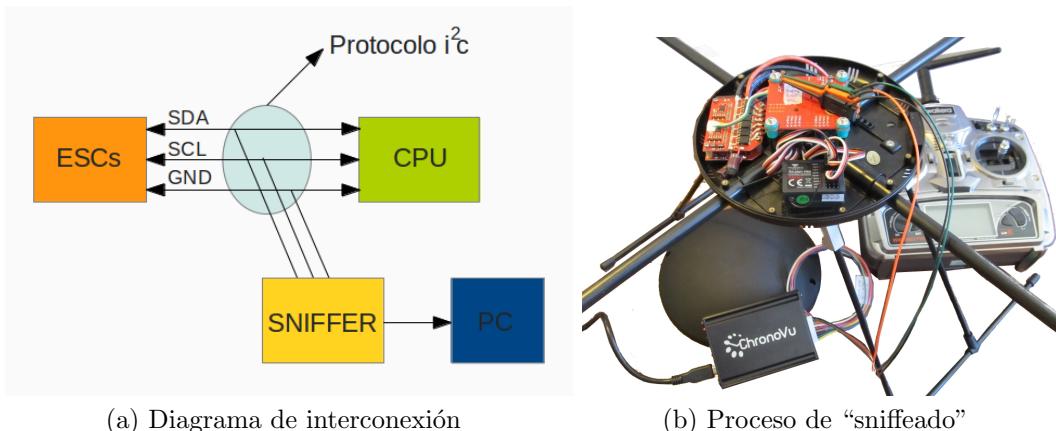


Figura 5.3.: Proceso de lectura del bus.

Al realizar este proceso se puede observar que se repiten, cada $2ms$, bloques similares al mostrado en la figura 5.4.

De acuerdo a la observación de dichos bloques, teniendo en cuenta la secuencia de comunicación típica entre maestro y esclavos mencionada anteriormente, se deduce que hay 4 esclavos correspondientes a los 4 ESC's de los 4 motores, cuyas direcciones en hexadecimal son **D0**, **D2**, **D4** y **D6**. La dirección de la memoria interna de los esclavos donde se almacenan los datos que envía el maestro es, en todos los casos **A2**. Además el maestro envía un tercer conjunto de datos que refiere, de alguna manera, a la velocidad con la que debe girar cada motor. Este conjunto de órdenes,

5. Ingeniería inversa

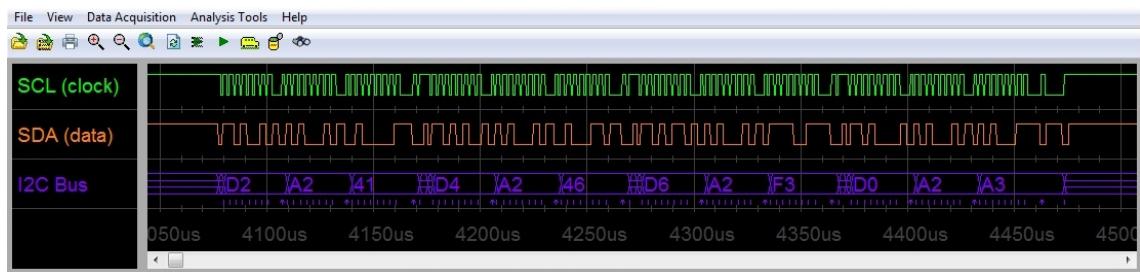


Figura 5.4.: Bloque de transmisión i^2c

agrupadas por bloques como el que se muestra en la figura 5.4, se repite periódicamente, indicando la velocidad con la que debe girar cada motor.

Para comprender las pruebas realizadas es importante dejar en claro algunos aspectos previos. En la figura 5.5 se muestra el transmisor utilizado para enviar comandos al cuadricóptero, un **Walkera WK-2801** y se indican los nombres de los mandos más importantes del mismo.



Figura 5.5.: Transmisor

Al mover el mando de la izquierda (**Elev/Rudder**) en la dirección vertical (**Elev**) se logra que el cuadricóptero se eleve verticalmente, dando igual potencia a todos los motores, mientras que al moverlo en la dirección horizontal, el cuadricóptero presenta un movimiento de rotación según su eje vertical.

Al mover el mando de la derecha (**Throttle/Aile**) en la dirección horizontal (**Aile**) y vertical (**Throttle**), se logran movimientos de rotación según los ejes horizontales x e y del cuadricóptero. Las definiciones de los ejes se pueden ver en la figura 5.1.

Se analizan las siguientes situaciones:

5.2. Pruebas en régimen

Id	Elev	Rudder	Aile	Throttle	Movimiento
0	atrás	medio	medio	medio	idle
1	medio	medio	medio	medio	vertical hacia arriba con aceleración constante
2	adelante	medio	medio	medio	vertical hacia arriba con aceleración constante
3	medio	izquierda	medio	medio	giro según eje z
4	medio	derecha	medio	medio	giro según eje $-z$
5	medio	medio	izquierda	medio	giro según eje $-x$
6	medio	medio	derecha	medio	giro según eje x
7	medio	medio	medio	atrás	giro según eje $-y$
8	medio	medio	medio	adelante	giro según eje y

Cuadro 5.1.: Pruebas realizadas

Se realiza un análisis de los resultados obtenidos para cada motor, graficando el contenido del byte de datos que se le transmite a cada motor. Se obtienen representaciones como la mostrada en la figura 5.6. En dicha figura puede observarse que los 4 motores reciben un byte con el valor promedio en 50, el cual corresponde a la mínima potencia entregada a los motores para encenderlos.

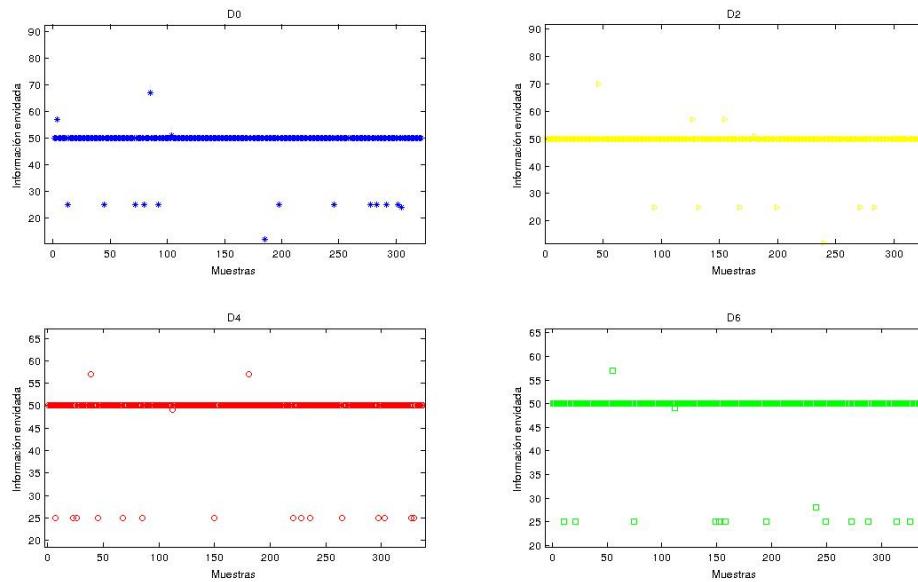


Figura 5.6.: Prueba N° 0

Haciendo un análisis similar con el resto de las pruebas detalladas en la tabla A.1 se construye la tabla 5.2 donde se muestran los valores enviados a cada motor en promedio en todas las pruebas.

5. Ingeniería inversa

Prueba	Valor promedio				Movimiento
	D0	D2	D4	D6	
0	50	50	50	50	Idle
1	80	140	180	140	$a_z = cte \neq 0$
2	180	220	250	240	$a_z = cte \neq 0$
3	160	70	60	250	giro según eje z
4	50	200	200	100	giro según eje $-z$
5	250	160	140	50	giro según eje $-x$
6	50	160	150	250	giro según eje x
7	90	50	250	160	giro según eje $-y$
8	110	250	50	100	giro según eje y

Cuadro 5.2.: Resumen de los resultados obtenidos

En la tabla 5.3 se muestran los registros a los que se mandaron datos para cada dirección.

	D0	D2	D4	D6
Registro	A2	A2	A2	A2

Cuadro 5.3.: Resumen de las direcciones obtenidas

De las anteriores 2 tablas se pueden sacar algunas conclusiones que se analizarán en la sección 5.5.

5.3. Prueba de arranque

En esta sección se analiza la secuencia de arranque del cuadricóptero. Se parte con la palanca de elevación al mínimo y se procede a moverla para arrancar los motores.

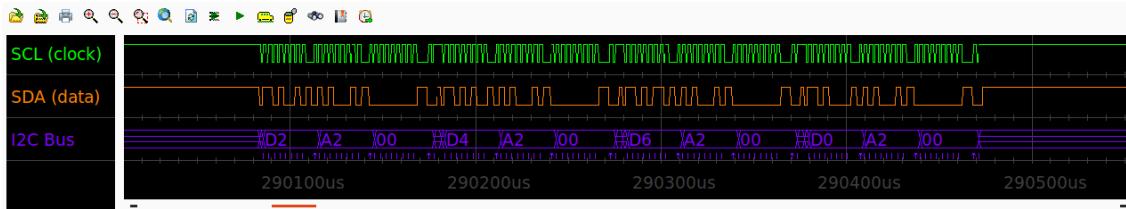
Mientras la palanca de elevación se mantiene al mínimo, se le envía el valor 0 a los cuatro motores. Al mover la palanca, luego de un *tiempo muerto* donde las líneas quedan inactivas, se les empieza a enviar el valor correspondiente a cada motor, como se puede ver en la figura 5.7.



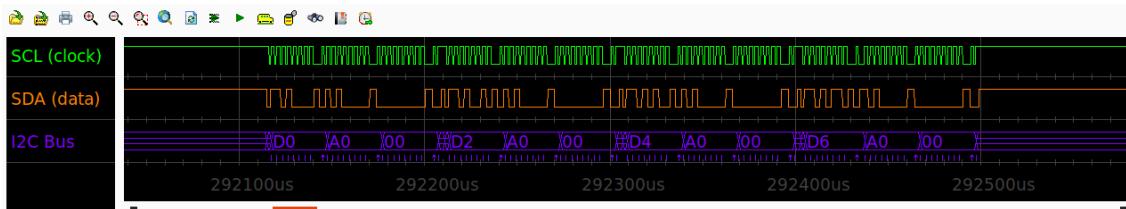
Figura 5.7.: Arranque

Observando más detenidamente los comandos enviados se pueden obtener algunas conclusiones importantes. En la gran mayoría de los casos se envían los comandos en el orden $D2 \rightarrow D4 \rightarrow D6 \rightarrow D0$ y se guardan en el registro 0xA2 del esclavo.

Pero se pueden observar dos importantes diferencias en el último comando que se manda a cada motor antes de arrancar, es decir, el último comando en la tanda de ceros mostrada en la figura 5.7:



(a) Tanda normal de ceros



(b) Última tanda de ceros

Figura 5.8.: Envío de ceros

- **El orden en el que se envían datos a los motores no es mismo** que el anterior. En este caso el orden es: $D0 \rightarrow D2 \rightarrow D4 \rightarrow D6$.
- **El registro en el que se escriben los datos es 0xA0**

En la figura 5.8a se muestra una tanda regular de ceros y en la figura 5.8b se muestra la última tanda de ceros mencionada.

Dichas diferencias se pueden observar claramente al graficar todos los datos obtenidos, como se muestra en la figura 5.9. Se grafica para cada motor el valor que le llega (figura 5.9a) y el registro donde el valor es escrito (figura 5.9b). Al analizar los resultados gráficamente resulta evidente el cambio en el registro al terminar de mandar los ceros. En la figura 5.9a se puede observar que el último cero que se manda a los motores, se manda alrededor de la muestra 150. Analizando la figura 5.9b es clarísimo que alrededor de la muestra 150, el registro al que se envían los comandos cambia al valor 160 (0xA0), tal como se había observado anteriormente.

5.4. Prueba de frenado

Se procede del mismo modo que en la prueba anterior (sección 5.3), analizando los comandos enviados a los motores a la hora de apagarlos. Se realiza una prueba análoga a la anterior partiendo de los motores funcionando y bajando al mínimo la palanca de elevación, causando que estos se detengan.

Una vez que se lleva la palanca de elevación al mínimo, los motores se apagan y se permanece enviando ceros a los motores en el orden $D2 \rightarrow D4 \rightarrow D6 \rightarrow D0$. Se pueden obtener algunas conclusiones importantes de la última tanda de

5. Ingeniería inversa

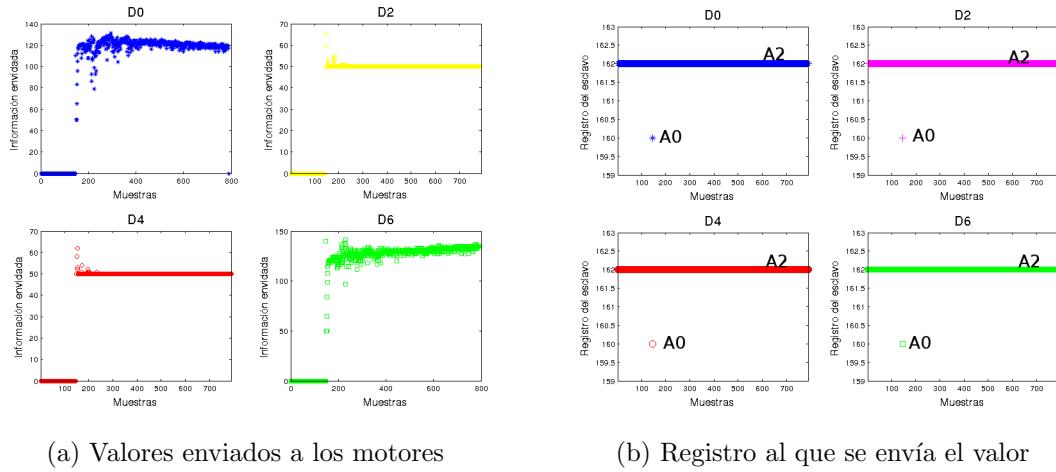


Figura 5.9.: Arranque

valores distintos de cero enviados (comandos que provocan el frenado de los motores), mostrada en la figura 5.10:

- El orden en el que se envían datos es $D0 \rightarrow D2 \rightarrow D4 \rightarrow D6$
- El registro en el que se escriben los datos es 0xA1

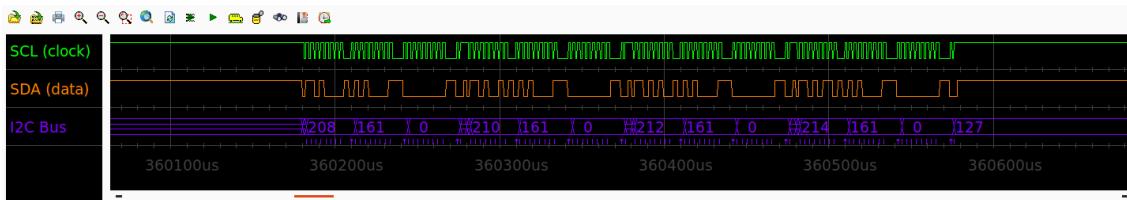


Figura 5.10.: Frenado

Se presenta también un análisis gráfico en la figura 5.11. En este caso, a diferencia del de la sección 5.3, no es posible divisar con claridad el registro 0xA1 (161) en la figura 5.11b, ya que se escriben comandos a una cantidad importante de registros diferentes.

Avalados por la prueba realizada en la sección 5.6 que se verá luego, se puede afirmar que la aparición de otros registros (por ejemplo los números 81, 145, etc) son causados por el propio proceso de intervención en los buses de comunicación y no son el comando transmitido a los motores. Analicemos por ejemplo el valor 81. Es claro que el resultado de dividir el valor 162 (correspondiente al conocido registro 0xA2) entre 2 es 81, por lo cual parece probable que haya sido generado por un error del analizador lógico, ya que un corrimiento hacia la derecha de un bit en una palabra binaria es equivalente a realizar una división entre 2. Es probable entonces que el sniffer haya incluido un bit en 0 antes de la verdadera palabra y haya descartado el último bit, causando una aparente división entre 2. Más graficamente:

$$162 = 10100010 \rightarrow 01010001\cancel{0} = 81$$

En verde se muestra el bit agregado y en rojo el eliminado.

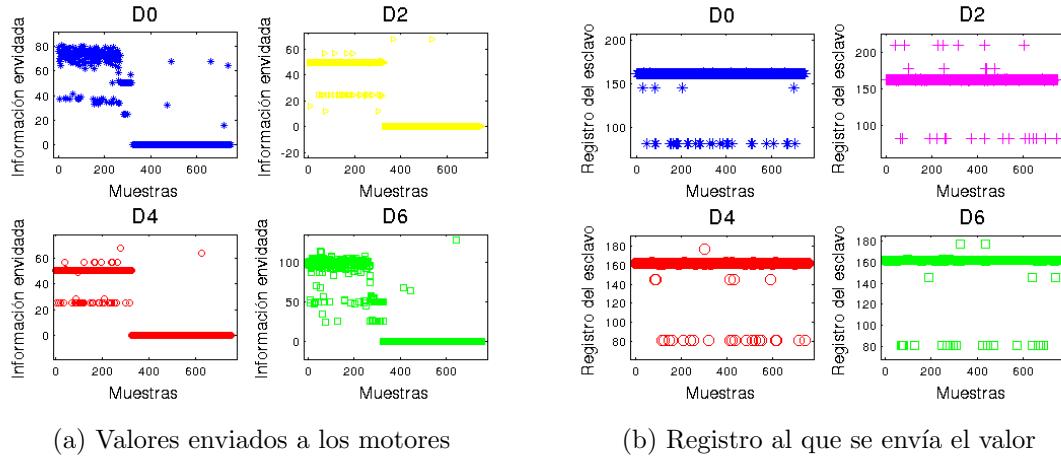


Figura 5.11.: Frenado

5.5. Conclusiones

- La comunicación entre el amo y los esclavos por medio del protocolo i^2c se lleva a cabo mediante un formato del tipo:
Dirección esclavo - Lugar de memoria donde guardar dato - dato
- Dicho formato se repite para todos los esclavos
- Cada esclavo recibe una actualización de estado (un dato nuevo) cada $2ms$
- La velocidad mínima de funcionamiento se logra enviando el valor 50
- La velocidad máxima de funcionamiento se logra enviando el valor 250
- Cuando el mando de Elevación del control se encuentra al mínimo, se les envía el valor 0 a los cuatro motores periódicamente con el formato mencionado.
- La dirección de memoria interna de todos los esclavos donde se guardan los datos recibidos por el maestro es siempre 0xA2 (ó 162), a excepción del arranque y el frenado.
- El último comando enviado antes de arrancar se guarda en el registro 0xA0, comando que se interpreta como una orden de arranque.
- El último comando enviado antes de frenar se guarda en el registro 0xA1, comando que se interpreta como una orden de frenado.
- El orden normal en el que se envían los comandos es $D2 \rightarrow D4 \rightarrow D6 \rightarrow D0$, menos en los casos descriptos en los 2 puntos anteriores donde el orden es $D0 \rightarrow D2 \rightarrow D4 \rightarrow D6$. Se comprobó, experimentalmente, que no es necesario alterar el orden en el que se envían los comandos para lograr arrancar y frenar los motores.
- La correspondencia entre las direcciones y los motores se muestra en la figura 5.12. El motor con dirección 0xD0 es el de adelante, el motor con dirección 0xD4 es el de atrás y mirándolo de frente el de la izquierda se corresponde con 0xD0, y el de la derecha con 0xD6

5. Ingeniería inversa

- Las direcciones 0xD0, 0xD2, 0xD4 y 0xD6 (de 8 bits) en realidad no refieren únicamente a direccionamiento, sino que incluyen además el bit de lectura/escritura, por lo cual cada esclavo tendrá una única dirección de 7 bits. Notar que las direcciones 0xD0 a 0xD6 al pasarlas a binario todas terminan en 0, lo cual implica que se tratan de comandos de escritura. Al omitir el último bit se obtienen las direcciones (sin incluir el bit de lectura/escritura) como se puede ver a continuación:

$$0xD0(11010000) \rightarrow 0x68(1101000) \quad (5.1)$$

$$0xD2(11010010) \rightarrow 0x69(1101001) \quad (5.2)$$

$$0xD4(11010100) \rightarrow 0x6A(1101010) \quad (5.3)$$

$$0xD6(11010110) \rightarrow 0x6B(1101011) \quad (5.4)$$

Las direcciones de los motores son entonces: **0x68, 0x69, 0x6A, 0x6B**.

- En las pruebas 0, 1, 2, 5, 6 y 7 se observa que la suma del valor promedio enviado al motor D0 más el enviado a la dirección D2 es similar a la suma de los valores entregados a los motores con dirección D4 y D6, lo cual implica un equilibrio en los pares realizados en ambos sentidos.

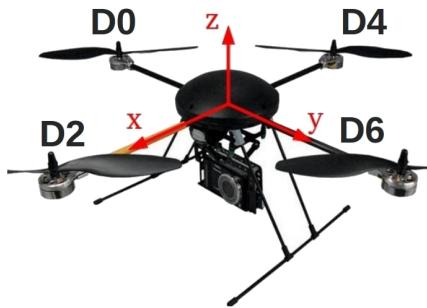


Figura 5.12.: correspondencias

5.6. Verificaciones

Se presentan diversas verificaciones realizadas para corroborar la veracidad de las conclusiones extraídas. Además de la corroboración experimental utilizando el puerto i^2c de la Beagleboard y logrando hacer funcionar los motores, previamente se realizaron 2 verificaciones adicionales:

- **i2cdetect**: comando enviado desde la *BeagleBoard* por el puerto i^2c-2 .
- **MSP430F5438**: microcontrolador que se programa como esclavo con una de las direcciones de un ESC.

i2cdetect

Utilizando la librería *i2c Tools*³ con la Beagleboard es posible enviar el comando **i2cdetect**, el cual “pregunta” a todos los esclavos presentes en el bus por su dirección. El resultado en terminal es el siguiente

³<http://www.lm-sensors.org/wiki/i2cToolsDocumentation>

```
root@beagleboard:~# i2cdetect -r 2
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-2 using read byte commands.
I will probe address range 0x03-0x77.
Continue? [Y/n] y
```

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	68	69	6a	6b	--	--	--	--
70:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Puede verse claramente que las direcciones de los esclavos obtenidas por la función mencionada son: **0x68, 0x69, 0x6A, 0x6B**.

MSP430F5438

El MSP430F5438 (ver figura 5.13) posee un puerto i^2c habilitado para su utilización. Se configurara como esclavo con la dirección 0x68 y se guardan en una variable los comandos recibidos en el bus. De este modo se logra independizarse de los problemas que pueda introducir el sniffer y se obtiene exactamente el comando transmitido al esclavo presente en dicha dirección.

Luego, analizando la variable donde se guardan los comandos, es posible obtener información más concluyente. En la figura 5.14 se muestran los resultados obtenidos al adquirir datos en un arranque, mientras que en la figura 5.15 se muestran los resultados obtenidos al adquirir datos al frenar.

Estas últimas dos figuras son las corroboraciones más concluyentes que se obtuvieron. Ratifica y deja bien en claro lo antes dicho sobre los registros: **para el arranque se escribe en la dirección 0xA0 del esclavo y para el frenado se escribe en la dirección 0xA1**.

A su vez sirve de confirmación para los errores del sniffado detectados en la sección 5.4.



Figura 5.13.: MSP430F5438

5. Ingeniería inversa

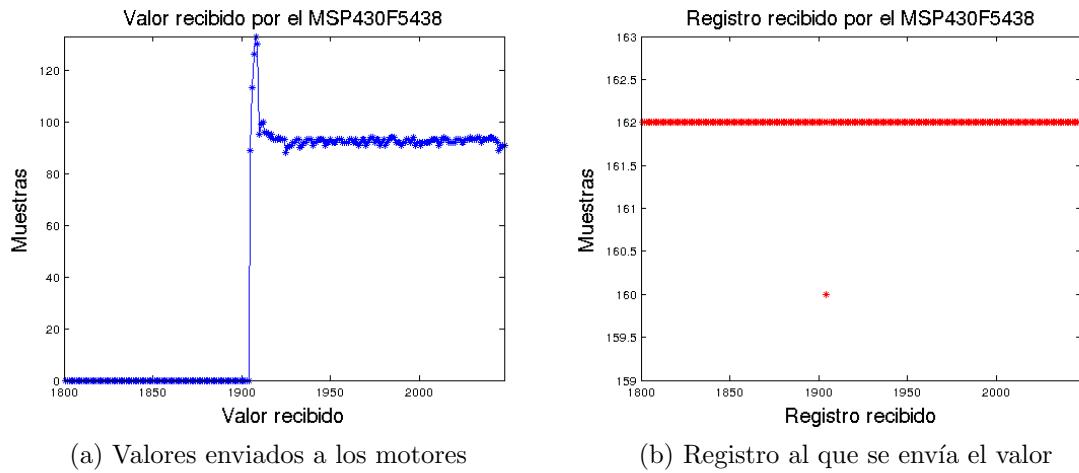


Figura 5.14.: Arranque detectado por el MSP430F5438

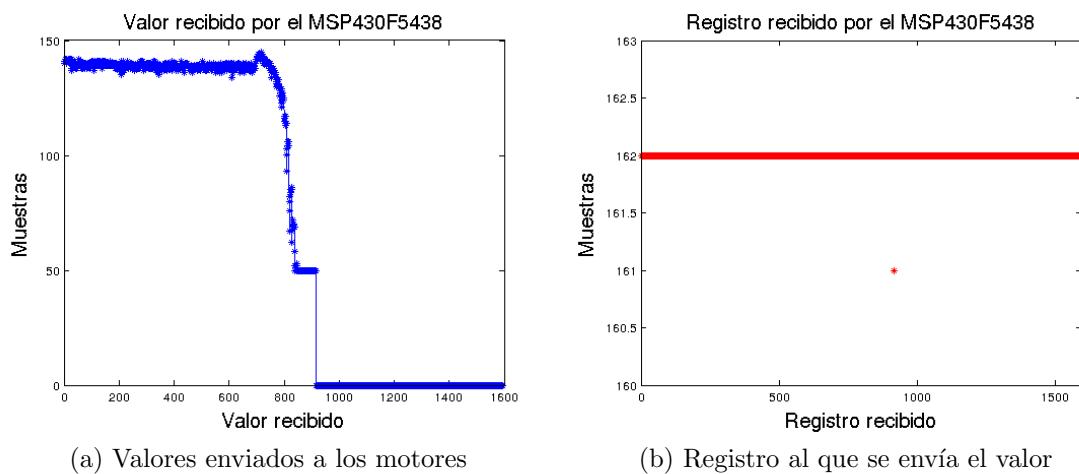


Figura 5.15.: Frenado

CAPÍTULO 6

CARACTERIZACIÓN DE LOS MOTORES Y LAS HÉLICES

6.1. Objetivo

El objetivo es el de caracterizar los motores del cuadricóptero Turbo Ace X720. Se busca:

- Analizar el comportamiento de los cuatro motores individualmente
- Determinar la relación entre velocidad angular y fuerza
- Determinar la relación entre comando I^2C y la velocidad angular.
- Determinar la relación entre velocidad angular y torque.
- Determinar la respuesta al escalón

Los algoritmos de control se encargan de definir la velocidad angular de los motores en cada instante, por dicho motivo es fundamental conocer la relación que existe entre esta última y la fuerza y el torque que se produce. La principal ventaja de esta elección es que el lazo de control implementado es independiente de la tecnología utilizada en el control de los motores (I^2C , PWM,etc) y por ende permite la reutilización del mismo. Dado que actualmente el control de los motores se establece mediante un comando I^2C es fundamental comprender la relación que existe entre este comando y la velocidad angular.

6.2. Materiales

- Cuadricóptero Turbo Ace X720
- LED IR TSAL6200
- Detector IR TSOP38256

6. Caracterización de los motores y las hélices

- Resistencia de 20Ω
- Generador de onda Tektronix CFG250
- Fuente de alimentación de $5V_{DC}$
- Osciloscopio digital GwINSTEC GDS2062
- Balanza Presiser LK-15P presición media III
- Beagleboard XM rev C
- Analizador lógico ChronoVu
- Buffer Octal 74HC245

6.3. Procedimiento

6.3.1. Consideraciones previas

El detector IR TSOP38256 es sensible a radiaciones infrarrojas moduladas a una frecuencia de 56 KHz. Si se lo expone a una onda cuadrada de dicha frecuencia la salida del mismo es un “0” lógico. Si no recibe dicha radiación la salida es un “1” lógico.

El dispositivo de medida de velocidad angular se trata de un cuerpo en forma de “U”, de un lado se tiene el LED infrarrojo y del otro el detector, tal como se puede ver en la figura 6.1. La idea del dispositivo de medida es sencilla. Se trata de hacer “pasar” la radiación infrarroja emitida por el LED a través de la hélice en funcionamiento de uno de los motores. Esta radiación es recogida del otro lado por el receptor IR. De este modo tendremos a la salida del detector, pulsos de frecuencia correspondientes a la velocidad con la que la hélice obstruye el camino entre el sensor y el LED. **La velocidad angular será entonces la mitad de dicha frecuencia.**



Figura 6.1.: Dispositivo en forma de U

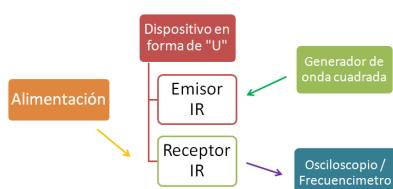


Figura 6.2.: Diagrama de interconexión

Como explicamos anteriormente el LED debe ser comutado con una frecuencia de 56kHz por lo tanto se lo alimentará con un generador de señales funcionando como generador de onda cuadrada a dicha frecuencia. Para lograr un correcto funcionamiento del LED se requiere una corriente superior a los $100mA$. La amplitud de la onda cuadrada se regulará a $5V$ y se trabajará con una resistencia

de 20Ω en serie. La salida del receptor IR se conecta directamente al osciloscopio digital, el cual es capaz de medir frecuencias. El diagrama de interconexión puede apreciarse en la figura 6.2.

6.3.2. Relación entre comando I^2C , velocidad angular y empuje

Objetivos particulares

Este experimento servirá para relevar:

- Diferencias en el comportamiento de los motores
- Relación entre velocidad angular y fuerza
- Relación entre comando I^2C y la velocidad angular.

Modelos de ajuste

Si no se envía ningún comando I^2C los motores no giran. Es evidente que si los motores no giran las hélices no realizan ningún empuje. Por lo tanto los modelos deben “pasar” por el origen.

- Para la relación entre velocidad angular y fuerza, en base a [2], se proponen los siguientes modelos de ajuste:
 - Modelo cuadrático $T = a\omega^2 + b\omega$
 - Modelo cúbico $T = a\omega^3 + b\omega^2 + c\omega$
- Para la relación entre velocidad angular y comando I^2C se proponen los siguientes modelos de ajuste:
 - Modelo cuadrático $x = a\omega^2 + b\omega$
 - Modelo cúbico $x = a\omega^3 + b\omega^2 + c\omega$

Desarrollo del experimento

El procedimiento consiste en enviar distintos comandos I^2C al cuadricóptero y registrar las diferentes lecturas de masa en la balanza para determinar el empuje de los motores y en registrar las diferentes medidas de frecuencia en el osciloscopio para determinar la velocidad angular de los cuatro motores. El setup experimental puede observarse en la figura 6.3. Se solidariza el cuadricóptero a una base de madera. Sobre esta última se añade peso suficiente para asegurar que el cuadricóptero no se eleve, se agregaron 3Kg de sobrepeso.

Al estar el sistema en equilibrio mecánico se cumple que:

$$\sum F_{ext} = 0 \quad (6.1)$$

En este caso las fuerzas presentes son el peso del sistema ($M_{total}g$) la normal de la balanza sobre el sistema ($N = N_1 + N_2$) y el empuje de los motores ($T_{total} = T_1 + T_2 + T_3 + T_4$). Por lo tanto:

$$T_{total} - M_{total}g + N = 0 \quad (6.2)$$

6. Caracterización de los motores y las hélices

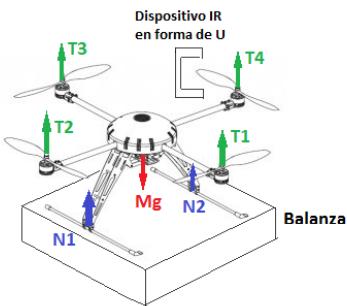


Figura 6.3.: Primer set-up experimental

Por lo tanto el empuje de los motores puede calcularse como:

$$T_{total} = M_{total}g - N \quad (6.3)$$

La lectura de la balanza no es otra cosa que $\frac{N}{g}$. Una vez ubicado el cuadricóptero con el sobrepeso sobre la balanza se tara la balanza. Por lo tanto la lectura que indica la balanza (M_{medida}) luego de realizada esta acción es:

$$M_{medida} = \frac{N - M_{total}g}{g} \quad (6.4)$$

Por lo tanto el empuje de los cuatro motores puede calcularse como:

$$T_{total} = -M_{medida}g \quad (6.5)$$

Asumiendo que los cuatro motores se comportan en forma similar se tiene que:

$$T = -\frac{M_{medida}g}{4} \quad (6.6)$$

La medida de frecuencia se realiza con el dispositivo IR descripto en la sección anterior. La frecuencia medida (f_{medida}) corresponde al inverso del tiempo que transcurre entre el pasaje de una hoja de la hélice y la otra. El período de la rotación de la hélice es el doble y por ende la frecuencia de la rotación es la mitad de la frecuencia medida. Por lo tanto se tiene que:

$$\omega = 2\pi f = \pi f_{medida} \quad (6.7)$$

Se mide la velocidad angular de los cuatro motores y se trabaja con el promedio. En la sección 6.4.1 se verá porqué es adecuado considerar el promedio.

6.3.3. Relación entre velocidad angular y torque

Objetivos específicos

El objetivo de este experimento es obtener la relación entre velocidad angular y torque.

Modelos de ajuste

Nuevamente el modelo se fuerza para obtener un torque nulo a velocidad angular nula. De acuerdo al análisis realizado en 3 se propone un modelo cuadrático:

$$Q = a\omega^2 + b\omega$$

Desarrollo del experimento

Se ubica el cuadricóptero sobre la balanza. Se retiran tres de las cuatro hélices del cuadricóptero. La cuarta hélice se rota de forma que su eje principal se encuentre paralelo al plano de la balanza. Se ubica el detector IR de forma de poder medir la velocidad angular de la hélice. El setup de medida puede verse en la figura 6.4. De acuerdo a lo estudiado la hélice presenta un torque (Q) negativo respecto de su eje central si el motor rota en sentido anti-horario. Nos proponemos calcular el torque total respecto de dicho eje. Las fuerzas presentes en el sistema son: el peso, la fuerza de empuje, la resultante de las normales ($N = N_1 + N_2$) y las fuerzas de fricción en el plano del plato de la balanza. Estas últimas y el empuje no realizan ningún torque en la dirección de interés. Recordamos que el torque de una fuerza respecto de un eje se calcula como:

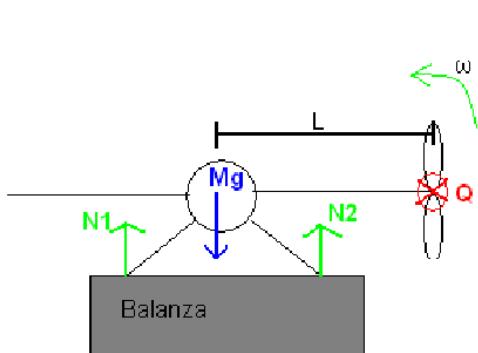


Figura 6.4.: Segundo set-up experimental

Definimos el largo $L = 29,3\text{cm}$ como la distancia entre el eje de la hélice y el centro de masa del cuadricóptero. El peso se aplica en el centro de masa y se asume por simetría que la resultante de las normales se aplica sobre la recta que pasa por el centro de masa del cuadricóptero y es perpendicular a la balanza.

De esta forma se tiene que el torque total vale:

$$\tau = -Q + LM_{total}g - LN = -Q + L(M_{total}g - N) \quad (6.9)$$

Como se explico anteriormente luego de tarar la balanza la masa medida por la misma corresponde a $M_{medida} = \frac{N - M_{total}g}{g}$. De esta forma se puede escribir el torque total como:

$$\tau = -Q - LM_{medida}g \quad (6.10)$$

Al igual que en el experimento anterior el sistema se encuentra en equilibrio mecánico y por lo tanto:

$$\tau = 0 \quad (6.11)$$

De esta forma queda claro que el torque que nos interesa caracterizar puede calcularse como:

$$Q = -LM_{medida}g \quad (6.12)$$

6.3.4. Respuesta al Escalón

Se caracterizará al motor en lo que respecta a su respuesta al escalón. Se verá cuánto demora un motor en pasar de la velocidad angular en régimen correspondiente al valor de comando I^2C 50 a la velocidad angular en régimen correspondiente al valor de comando I^2C 150. El osciloscopio no resulta adecuado para observar las diferencias de velocidad angular obtenidas. Se procede a conectar la salida del sensor

6. Caracterización de los motores y las hélices

IR a un *buffer* octal y la salida de este último al analizador lógico¹. El analizador lógico registra la salida del sensor IR durante 5 segundos a una tasa de muestreo de 2kHz. Se obtendrán las diferencias de tiempos entre flancos de subida sucesivos(t_f). Cada flanco de subida corresponde a una hoja de la hélice siendo detectada por el sensor IR. La velocidad angular de la hélice puede calcularse como:

$$\omega = 2\pi \frac{1}{2t_f} = \frac{\pi}{t_f} \quad (6.13)$$

6.4. Resultados y análisis

6.4.1. Comparación entre motores

En la tabla 6.1 se presentan los resultados obtenidos experimentalmente de acuerdo al procedimiento descrito.

I ² C	Peso medido (g)	Frecuencia (Hz)			
		Motor D0	Motor D2	Motor D4	Motor D6
0	0	0	0	0	0
50	-165	34.2	34.3	35.5	34.3
70	-360	49.2	49.5	49	48.4
90	-590	63.7	62.7	63.3	63.5
110	-865	76.7	74.4	75.5	73.4
130	-1120	87.3	86.3	86.2	84.7
150	-1395	97.5	95.4	96.2	94.2
170	-2030	104.7	104.2	106.2	105.5
200	-2535	115.5	113.4	119.9	117.3

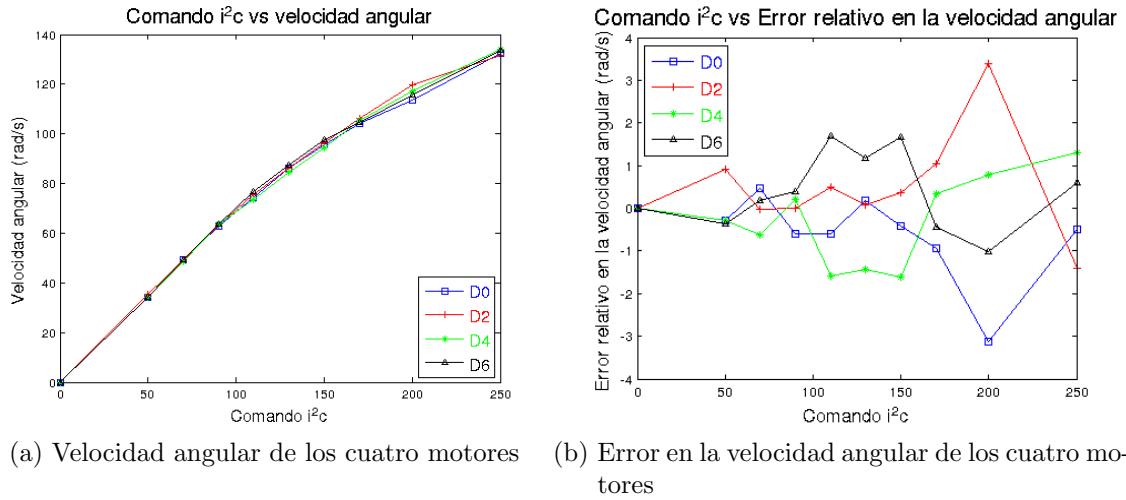
Cuadro 6.1.: Comando I^2C enviado, peso medido y frecuencia medida para los cuatro motores

En la figura 6.5a se presentan las velocidades angulares de los cuatro motores contra el comando I^2C , mientras que en la figura 6.5b se muestra la diferencia entre la velocidad angular de cada motor y el promedio de las mismas para cada comando I^2C enviado. En la figura 6.5a se observa que el comportamiento de los motores es en todos los casos similar. Para velocidades angulares bajas los motores se comportan en forma idéntica y lineal, y comienzan a distanciarse al aumentar la velocidad.

En la figura 6.5b puede observarse que la máxima diferencia entre las velocidades angulares de dos motores es de aproximadamente $6,5rad/s$. Dicha diferencia se obtiene para un valor de comando I^2C igual a 200. Si observamos la velocidad angular a dicho valor de comando I^2C en la tabla 6.1 se observa que el menor valor corresponde a $113,4rad/s$. El error relativo que se obtiene es inferior al 6 %

Se puede afirmar que:

¹ Esto fue necesario ya que la tensión de la salida del IR disminuía drásticamente al conectarlo directamente al analizador lógico a valores que este último no es capaz de identificar como “1” lógicos.


 Figura 6.5.: Curva I^2C vs. velocidad angular para los 4 motores

- Es válido promediar las velocidades angulares de los 4 motores y trabajar con esos promedios como un motor típico. Además de este modo se reducen los posibles errores que se pueden haber causado a la hora de realizar las medidas.
- Por lo tanto la fuerza medida (de los 4 motores juntos) corresponde con el cuádruple de la fuerza ejercida por cada uno de ellos

6.4.2. Obtención de la curva fuerza contra velocidad angular

Se trabaja con la medida de peso presentada en la tabla 6.1 y con el promedio de las medidas de la frecuencia de dicha tabla.² En la figura 6.6a se observan los puntos obtenidos experimentalmente y las curvas obtenidas con los modelos de ajuste propuestos. En dicha figura se observa que ambos modelos aproximan adecuadamente los puntos obtenidos experimentalmente.

Para el modelo cuadrático se obtiene:

- Parámetros:
 $a = 4,6016 \times 10^{-5} Ns^2$,
 $b = -1,0380 \times 10^{-3} Ns$
- Error promedio:
 $e = -3,9951 \times 10^{-4} N$,
 $e = -0,4072 \times 10^{-1} g$
- Desviación estándar :
 $\sigma = 2,3871 \times 10^{-2} N$,
 $\sigma = 2,4333 g$

Para el modelo cúbico se obtiene:

²De acuerdo a las justificaciones realizadas en la sección 6.4.1

6. Caracterización de los motores y las hélices

- Parámetros:

$$a' = 3,0619 \times 10^{-9} Ns^3,$$

$$b' = 4,1319 \times 10^{-5} Ns^2,$$

$$c' = 3,9160 \times 10^{-4} Ns$$

- Error:

$$e = -1,4331 \times 10^{-4} N,$$

$$e = -0,1461 g$$

- Desviación estándar:

$$\sigma = 2,3377 \times 10^{-2} N,$$

$$\sigma = 2,3830 g$$

El cálculo del error promedio y la desviación estándar no presenta diferencias significativas para uno y otro modelo, por lo tanto se opta por trabajar con el modelo más sencillo, es decir el cuadrático. Asimismo el error obtenido es despreciable respecto de la resolución de la balanza y la desviación estándar es del orden de la presición de la misma, por lo que se consluye que la caracterización es satisfactoria. Se tiene entonces que:

$$T = 4,6016 \times 10^{-5} \omega^2 - 1,0380 \times 10^{-3} \omega \quad (6.14)$$

Con ω en rad/s y T en N .

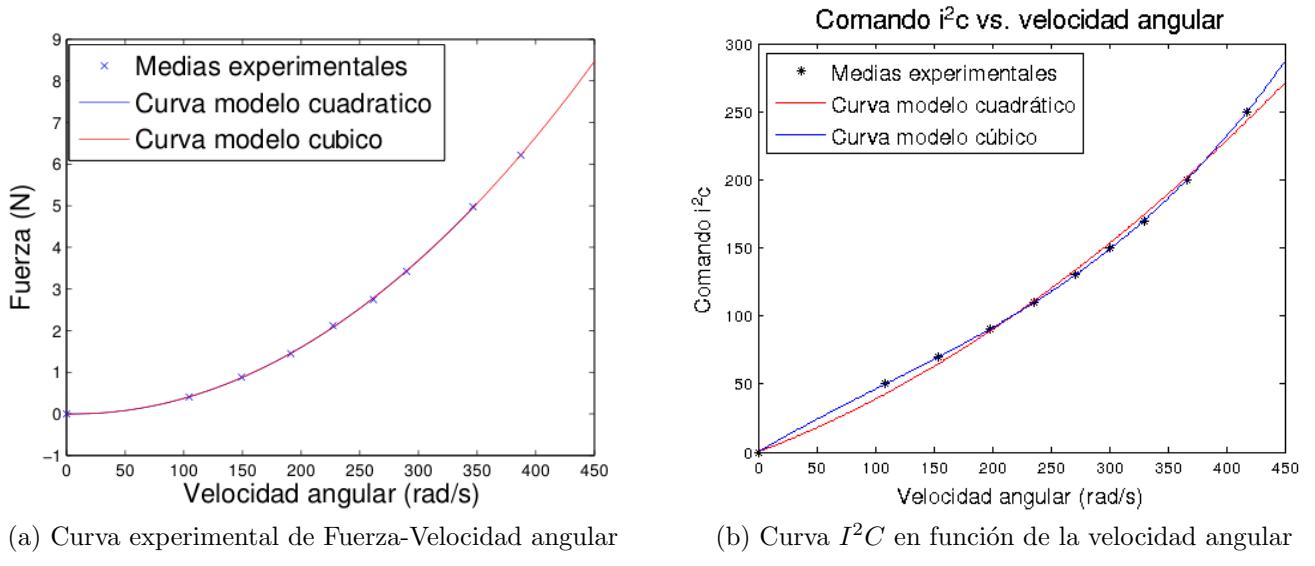


Figura 6.6.

6.4.3. Obtención de la curva I^2C contra velocidad angular

A partir de los resultados presentados en la tabla 6.1 y la ecuación 6.7 se intenta obtener la relación entre comando I^2C y velocidad angular. En la figura 6.6b se muestran los datos obtenidos experimentalmente y las curvas predichas por los modelos de ajuste considerados. A simple vista en la figura puede observarse que el modelo cúbico aproxima mejor que el modelo cuadrático. Los resultados obtenidos para los modelos de ajuste propuesto vienen a confirmar dicha observación.

Para el modelo cuadrático se obtiene:

- Parámetros:
 $a = 6,1226 \times 10^{-4} s^2$,
 $b = 0,3270 \times 10^{-4} s$
- Error promedio: $e = 5,08 \times 10^{-1}$
- Desviación estándar: $\sigma = 4,42$

Para el modelo cúbico se obtiene:

- Parámetros:
 $a' = 2,2118 \times 10^{-6} s^3$,
 $b' = -7,1258 \times 10^{-4} s^2$,
 $c' = 0,5106 s$
- Error promedio: $2,04 \times 10^{-3}$
- Desviación estándar: $\sigma = 4,01 \times 10^{-1}$

El error promedio obtenido con el modelo cúbico y la desviación estándar son menores que en el modelo cuadrático. Además se observa claramente que la curva del modelo cúbico ajusta mejor los puntos. Esta es evidencia suficiente para elegir dicho modelo. Tendremos entonces que:

$$x = 2,2118 \times 10^{-6} \omega^3 - 7,1258 \times 10^{-4} \omega^2 + 0,5106 \omega \quad (6.15)$$

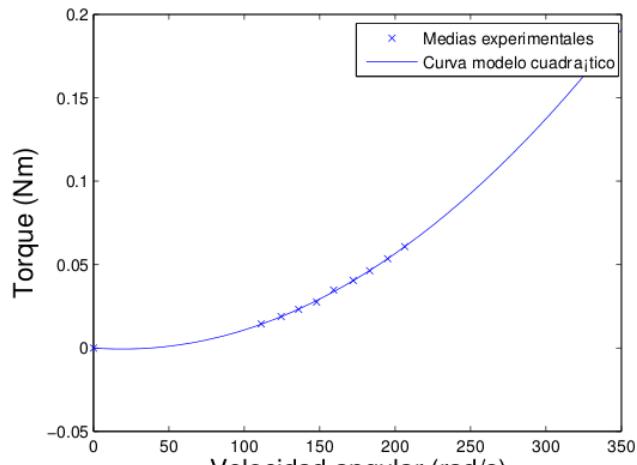
Con ω en rad/s .

6.4.4. Obtención de la curva velocidad angular contra torque

En la tabla 6.7b se presentan las medidas obtenidas de velocidad angular y masa medida por la balanza de acuerdo al procedimiento detallado en 6.3.3.

Recordamos que para la caracterización de la respuesta velocidad angular - torque se propuso un modelo cuadrático. Los resultados obtenidos fueron:

- Parámetros:
 $a = 3,4734 \times 10^{-6} Nms^2$,
 $b = -1,3205 \times 10^{-4} Nms$
- Error promedio:
 $\mu = 1,7824 \times 10^{-6} Nm$
- Desviación estándar:
 $\sigma = 9,2686 \times 10^{-4} Nm$



(a) Curva torque contra velocidad angular

I^2C	Frec. (Hz)	peso (g)
0	0	0
50	35.4	-10
55	39.6	-13
60	43.3	-16
65	47.1	-19
70	50.7	-24
75	54.8	-28
80	58.3	-32

6. Caracterización de los motores y las hélices

Si se asume que el error al realizar la medida en el largo del brazo del cuadricóptero es despreciable, se pueden convertir los errores obtenidos en Nm a gramos.

- $\mu = 0,6265 \times 10^{-3}g$

- $\sigma = 3,2580 \times 10^{-1}g$

Tanto el error como la desviación estándar son inferiores en al menos un orden que la resolución de la balanza. En la figura 6.7a pueden observarse las medidas experimentales obtenidas y la curva que corresponde al modelo elegido para realizar el ajuste, el resultado es satisfactorio.

6.4.5. Respuesta al escalón

En la figura 6.4.5 se muestran las velocidades angulares obtenidas. El tiempo de rise obtenido es de $0,19s$ y la respuesta al escalón no presenta prácticamente sobretiempo.

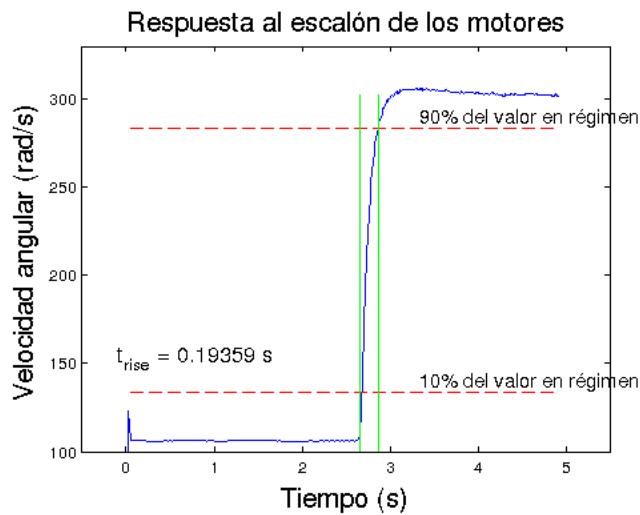


Figura 6.8.: Curva I^2C contra fuerza

Dado que siempre se trabajará en torno a un valor dado de velocidad angular (ver capítulo 13) no se producirán variaciones tan abruptas en la velocidad angular de los motores, por lo tanto podemos considerar que los tiempos de respuesta serán mejores que el obtenido en esta prueba, siendo suficientemente rápidos como para poder despreciar el transitorio de los motores.

Parte IV.

Sensores y estimación del estado

CAPÍTULO 7

CALIBRACIÓN DEL ACELERÓMETRO

7.1. Objetivos

Realizar una serie de pruebas con el fin de caracterizar y calibrar el acelerómetro de tres ejes de la unidad de medida inercial (IMU).

7.2. Materiales

- Mongoose 9DoF IMU de *Ckdevices*
- Prisma de madera
- Escuadra con ángulo de 45°
- Mesa nivelable con tapa móvil ajustable a distintos ángulos
- Nivel

7.3. Marco Teórico

Existen diversas no idealidades que afectan la lectura de los valores de aceleración registrados por el acelerómetro. Las no idealidades a considerar basados en lo desarrollado por [5] son:

- Ruido inherente
- Relación entre aceleración real y lectura del acelerómetro no lineal.
- No ortogonalidad de los ejes
- Drift aleatorio
- Variación de las medidas con la temperatura

7. Calibración del acelerómetro

Ruido Inherente

Existe un ruido inherente del cual no nos preocuparemos en demasiá ya que el mismo se modela como un proceso estocástico de distribución normal y media nula, como veremos luego. Por lo tanto, en esta etapa de calibración donde se tomarán cientos de muestras y se promediarán se puede trabajar sin considerar este ruido.

Relación no lineal

En la hoja de datos del acelerómetro ADXL345 de la *Mongoose 9DOF IMU* se declara que en cualquiera de los tres ejes el error debido a una respuesta no lineal es, en el peor caso, $\pm 0.5\%$ del valor de fondo de escala, por lo tanto se decide considerar para el acelerómetro una respuesta lineal.

No ortogonalidad de los ejes

Debido a defectos de construcción los ejes de sensibilidad del dispositivo pueden no ser ortogonales. Evidentemente si no se considera este aspecto se tendrá un error que puede ser importante en las medidas de aceleración. Se modela esta no idealidad considerando la siguiente relación entre la aceleración medida en cada eje de sensibilidad del acelerómetro y la aceleración medida en un sistema solidario a la plataforma:

$$\mathbf{a}^p = T_a^p \mathbf{a}^a, \quad T_a^p = \begin{pmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{pmatrix} \quad (7.1)$$

Donde:

- α_{ij} es la rotación del i-ésimo eje de sensibilidad del acelerómetro sobre el j-ésimo eje del sistema de la plataforma. Dichas rotaciones se ven representadas en la figura 7.1.
- \mathbf{a}^a es el vector de aceleración medido por el acelerómetro.
- \mathbf{a}^p es el vector de aceleración medido en el sistema solidario a la plataforma.

En la figura 7.1¹ se pueden observar dichos ángulos expresados gráficamente. Estos son constantes ya que su origen es puramente de construcción y se puede asumir que las orientaciones se mantendrán lo suficiente como para que no sea necesario realizar re-calibraciones constantemente.

Asumiendo un modelo lineal, la relación entre la aceleración y la medida realizada se puede expresar matricialmente de la siguiente forma:

$$\tilde{\mathbf{a}}^a = K_a \mathbf{a}^a + \mathbf{b}_a \quad (7.2)$$

donde $\tilde{\mathbf{a}}^a$ es la aceleración medida, K_a es una matriz diagonal que representa el factor de escala para convertir del valor digital a una aceleración correspondiente. \mathbf{b}_a no es otra cosa que un término independiente para corregir la posición del cero. Estos dos parámetros poseen algunas de las fuentes de error que resulta más difícil

¹Figura extraída de [5]

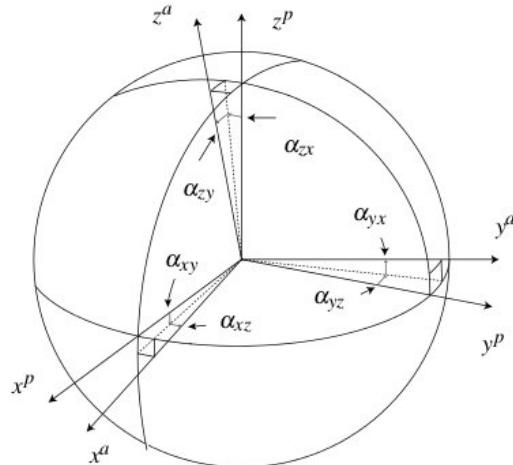


Figura 7.1.: Rotaciones entre los ejes de la plataforma y del acelerómetro

de corregir. Resulta que dichos parámetros varían con la temperatura, por lo tanto el resultado obtenido durante la calibración (temperatura ambiente) puede diferir de los valores obtenidos una vez instalado el acelerómetro dentro del cuadricóptero (temperaturas superiores a los $30^{\circ}C$). A partir de la hoja de datos del fabricante se decide modelar el *offset* mediante una relación lineal con la temperatura. En una primera aproximación no se considera este último factor.

Modelo del acelerómetro

A partir del análisis realizado se decide considerar el siguiente modelo de acelerómetro:

$$\tilde{\mathbf{a}}^a = K_a(T_a^p)^{-1}\mathbf{a}^p + b_a \quad (7.3)$$

7.4. Procedimiento

7.4.1. Caracterización de las no idealidades variables

Previo a la calibración de los parámetros estáticos parece interesante estudiar una serie de algunos minutos a una tasa de muestreo de 100Hz. Estos datos serán útiles para la determinación de dos no idealidades:

- Ruido inherente
- Drift aleatorio

Además se realizará una medida variando la temperatura ambiente a fin de determinar la magnitud del Drift térmico.

7.4.2. Determinación de parámetros estáticos

Como se desprende de la sección anterior, para poder calibrar el acelerómetro se deben determinar 12 parámetros. Por dicho motivo, se precisan obtener al menos 12 ecuaciones diferentes. Típicamente se trabaja con el doble o triple de ecuaciones que de parámetros a determinar. Para mejorar los resultados se trabajará con 27 medidas donde cada una de ellas aporta 3 ecuaciones (una por cada eje), logrando un

7. Calibración del acelerómetro

total de 81 ecuaciones, y se procederá a determinar los parámetros que minimizan el error cuadrático medio entre los valores efectivamente medidos y los valores teóricos.

Preparación

Para realizar una buena calibración es muy importante contar con una figura geométrica en forma de paralelepípedo que pueda solidarizarse a la IMU. De esta forma nos aseguramos que no se introducen errores a la hora de orientar un eje en una dirección particular. Del mismo modo, es ideal disponer de una superficie perpendicular a la vertical y otra paralela, de modo que para rotar 90 grados la IMU, basta con rotar el paralelepípedo ese mismo ángulo, es decir, apoyar otra de las caras del mismo.

El prisma que se utilizó fue el cubo de lapacho que puede verse en la figura 7.2. Los ángulos del mismo son prácticamente rectos difiriendo a lo sumo en medio grado.

Se construyó una plataforma de madera en forma de mesa con la tapa superior móvil, de modo que permite el giro sobre uno de sus lados (eje de giro). Se muestra en la figura 7.3. A su vez, es posible ajustar el dispositivo, de forma de asegurar que dicho eje se encuentre en posición horizontal, por medio de 3 patas de altura regulable situadas en la parte inferior de la mesa. Para realizar las 27 medidas se irá modificando el ángulo de la tapa móvil de la plataforma y será medido con un semicírculo solidarizado a dicha tapa y un péndulo. En cada posición de la plataforma se tomarán medidas colocando el cubo en 3 posiciones distintas.

Medidas a realizar

En reposo el acelerómetro medirá una fuerza igual a g en la dirección vertical y en el sentido radial saliente de la Tierra. En cada uno de los ejes de sensibilidad del acelerómetro se medirá la proyección ortogonal de dicho vector gravedad sobre cada eje. De esta forma, con distintas orientaciones se obtiene una descomposición distinta.

7.4.3. Variación de la medida con la temperatura

Para obtener una serie de datos adecuada se debe variar la temperatura en un rango que incluya la temperatura interna del cuadricóptero en funcionamiento y la temperatura ambiente durante la calibración. Asimismo la variación de la temperatura no puede ser excesivamente rápida. Se procede a calentar la IMU con un secador, una vez alcanzados los 48°C se deja enfriar. Se utilizarán los datos en la zona en la cual la temperatura disminuye.



Figura 7.2.: Cubo de lapacho

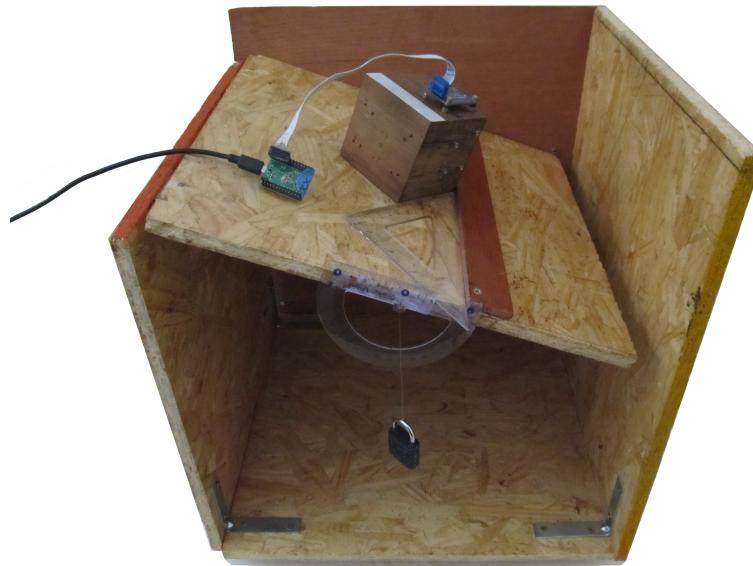


Figura 7.3.: Mesa - Plano inclinado regulable

7.5. Resultados y análisis

7.5.1. Medida estática durante una hora

En la figura 7.4 pueden observarse los datos crudos obtenidos de las aceleraciones registradas en los tres ejes de la IMU durante un período de 3 minutos 45 segundos.

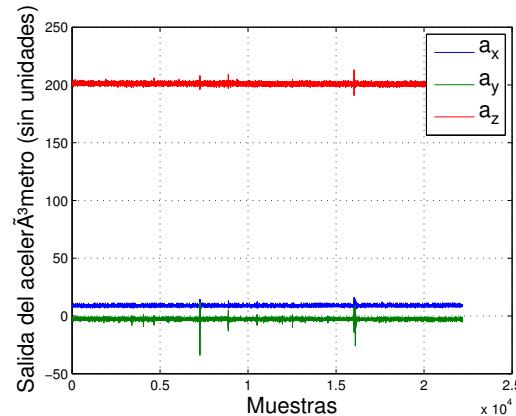


Figura 7.4.: Aceleraciones registradas por la IMU orientada con el eje z verticalmente y en reposo

En la figura se observa lo que se espera a priori de acuerdo a la posición del acelerómetro, dos medidas cercanas al cero y una medida de valor positivo. Esta última corresponde a la medida de la aceleración vertical en el sistema de caída libre. De la observación de esta serie de datos, avalados por características de la tecnología del acelerómetro, podemos afirmar que el ruido observado es asociable a un ruido blanco de media nula y que no existe deriva en las medidas realizadas, como era de esperarse. A partir de dichas constataciones se puede asegurar que es adecuado trabajar con el promedio de las muestras obtenidas en una posición.

7. Calibración del acelerómetro

7.5.2. Medias estáticas para el acelerómetro

Como se aclaró en la sección anterior se trabajará con el promedio de las aceleraciones medidas en una serie de datos. Cada serie de datos es de 20 segundos de duración. Además de variar el ángulo (θ) de la mesa, se rotará el prisma de forma de realizar una medida con cada eje perpendicular al plano de la mesa. Asimismo utilizando una escuadra se lograrán rotaciones de 45° respecto del eje perpendicular a la tapa (ϕ). Los resultados obtenidos se presentan en la tabla 7.1. Se utiliza la siguiente notación para las orientaciones utilizadas:

- Orientación “x”

- Eje “x” perpendicular a la tapa de la mesa y hacia abajo.
- Giro de la tapa de la mesa según el eje “z”.

- Orientación “y”

- Eje “y” perpendicular a la tapa de la mesa y hacia arriba.
- Giro de la tapa de la mesa según el eje “z”.

- Orientación “z”

- Eje “z” perpendicular a la tapa de la mesa y hacia arriba.
- Giro de la tapa de la mesa según el eje “-y”.

El tercer eje se deduce sabiendo que la base es directa. La aceleración medida es un número sin unidades.

Posición		Orientación x		Orientación y Aceleraciones		Orientación z	
$\theta(^{\circ})$	$\phi(^{\circ})$	Medida	Teórica ($m s^{-2}$)	Medida	Teórica ($m s^{-2}$)	Medida	Teórica ($m s^{-2}$)
0	0	-247.3172; 0.1591; -49.4095	-9.81; 0; 0	14.3760; 264.0399; -44.6882	0; 9.81; 0	12.1923; 0.2008; 207.5115	0; 0; 9.81
10	0	-242.5470; 50.0072; -48.7807	-9.6610; 1.7035; 0	15.3408; 259.7820; -91.4934	0; 9.661; -1.7035	62.8272; 0.0766; 203.3657	1.7035; 0; 9.661
10	45	-242.7077; 36.7365; -14.3685	-9.661; 1.2045; 1.2045	50.9558; 259.5086; -77.8004	1.2045; 9.661; -1.2045	48.5749; -35.4617; 203.0832	1.2045; -1.2045; -9.661
20	0	-229.2972; 95.2188; -48.4324	0; 9.2184; 3.3552	15.8545; 249.5544; -133.6116	0; 9.2184; 3.3552	108.2951; 0.0158; 191.7640	3.3552; 0; 9.2184
20	45	-229.6051; 68.0701; 15.6173	-9.2184; 2.3725; 2.3725	82.0560; 248.6061; -109.4629	2.3725; 9.2184; -2.3725	79.4781; -67.2837; 191.4327	2.3725; -2.3725; 9.2184
30	0	-209.4179; 134.6230; -47.5677	-8.4957; 4.905; 0	16.3148; 229.4927; -173.9674	0; 8.4957; -4.905	147.6114; -0.0215; 173.6084	4.905; 0; 8.4957
30	45	-209.7781; 97.1743; 43.9208	-8.4957; 3.4684; 3.4684	111.5630; 228.5963; -134.9143	3.4684; 8.4957; -3.4684	109.1504; -96.7951; 173.0896	3.4684; -3.4684; 8.4957
45	0	-166.6483; 191.3168; -46.3487	-6.9367; 6.9367; 0	17.0755; 186.6639; -226.5195	0; 6.9367; 6.9367	202.2578; -0.1806; 130.7166	6.9367; 0; 6.9367
45	45	-167.2223; 136.3814; 79.8707	6.9367; 4.905; 4.905;	150.2017; 185.5153; -172.5526	4.905; 6.9367; -4.905	147.2068; -135.3122; 130.4525	4.905; -4.905; 6.9367

Cuadro 7.1.: Aceleraciones medidas y aceleraciones teóricas en los tres ejes de la plataforma

Para la calibración se consideran solamente 24 de las 27 medidas realizadas. Esto es útil ya que quedan 3 medidas, que no serán tenidas en cuenta en la calibración, para realizar una validación de la calibración obtenida.

El problema de calibración consiste en encontrar el vector de parámetros θ que mejor ajusta las medidas obtenidas a los valores teóricos de aceleración. El vector θ

se compone de los elementos de la diagonal de la matriz K_a , de los elementos de b_a y de los α_{ij} de la matriz T_a^p . Es decir:

$$\theta = [k_{ax}, k_{ay}, k_{az}, b_{ax}, b_{ay}, b_{az}, \alpha_{yz}, \alpha_{zy}, \alpha_{xz}, \alpha_{zx}, \alpha_{xy}, \alpha_{yx}] \quad (7.4)$$

Como criterio de ajuste se decide minimizar la suma de los cuadrados de la diferencia entre la aceleración teórica y la aceleración medida. Dicho problema se puede expresar matemáticamente como:

$$\theta : \min \left\{ \sum_{i=1}^M \|a_i^p - T_a^p K_a^{-1} (\tilde{\mathbf{a}}_i^a - \mathbf{b}_a) \| \right\} \quad (7.5)$$

donde M es la cantidad de medidas realizadas, a_i^p los valores de aceleración teóricos en cada eje de la plataforma según la posición y orientación de la IMU y $\tilde{\mathbf{a}}_i^a$ los valores obtenidos de la IMU. Para resolver dicho problema se utiliza el algoritmo *lsqnonlin* de *MatLab* que es capaz de encontrar los mínimos locales de la función 7.5 a partir de una semilla.

El rango de aceleraciones manejado en la prueba es de $\pm 16g$. Según la hoja de datos del acelerómetro, se tiene una ganancia de $3,9mg/LSB$. El valor de la semilla de los términos que corresponden a la ganancia debe ser entonces $\frac{1}{g \times 3,9 \times 10^{-3}} = 26,1376$ para los tres ejes. La IMU expresa las aceleraciones obtenidas en *complemento A2*, por lo que una aceleración de 0 equivale a una lectura de 0, por lo tanto el valor de la semilla de los términos que corresponden al offset debe ser 0. Finalmente, los ángulos se suponen pequeños, por lo tanto una semilla válida parece ser:

$$\theta_0 = \left[\frac{1}{g \times 3,9 \times 10^{-3}}, \frac{1}{g \times 3,9 \times 10^{-3}}, \frac{1}{g \times 3,9 \times 10^{-3}}, 0, 0, 0, 0, 0, 0, 0, 0, 0 \right] \quad (7.6)$$

Los parámetros que minimizan la suma de los errores al cuadrado son los siguientes:

$$\theta = [26,7572 \quad 27,0,0599 \quad 25,9838 \quad 16,9374 \quad -0,2446 \quad -47,7863 \quad 0,0005 \quad 0,0025 \quad 0,0149 \quad 0,0023 \quad 0,0000 \quad -0,0077] \quad (7.7)$$

Luego de realizada la minimización se calcula la media del error cometido y la desviación estándar del error.

$$\mu = -2,0779 \times 10^{-10} ms^{-2}$$

$$\sigma = 0,0465 ms^{-2}$$

La media del error cometido es completamente despreciable, por lo tanto podemos afirmar que el 95 % de las medidas de aceleración realizadas tendrán un error inferior $2\sigma = 0,0930 ms^{-2}$.

Para verificar la calibración realizada se utilizan las medidas siguientes:

- $\theta = 0 \quad \phi = 0 \quad$ orientación x
- $\theta = 20 \quad \phi = 0 \quad$ orientación y

7. Calibración del acelerómetro

- $\theta = 10 \quad \phi = 45 \quad$ orientación z

Se presenta para las tres medidas la aceleración teórica, la medida convertida con los parámetros obtenidos y el error en la tabla 7.2.

Medida	Aceleración Medida (ms^{-2})	Aceleración teórica(m^{-2})	Error(ms^{-2})
1	-9.8762; -0.1323; -0.0622	-9.8100; 0.0000;0.0000	-0.0662; -0.1323; -0.0622
2	-0.0534; 9.2382;-3.3743	0.0000;9.2184;3.3552	-0.0534; 0.0198; -0.0191
3	1.2072;-1.3057;9.6649	1.2045;-1.2045;-9.661	0.0027; -0.1012 0.0039

Cuadro 7.2.: Medidas de verificación

En la verificación se obtienen valores aceptables aunque en la mayoría de los casos los errores superan la desviación estándar calculada anteriormente. El error cometido en la calibración del acelerómetro produce inevitablemente un drift en la velocidad y en la posición de la medida del sistema. Este drift será corregido agregando información proveniente de otros sensores, en un filtro de Kalman.

7.5.3. Variación con la temperatura

La temperatura de funcionamiento normal de los sensores es mucho mayor a la temperatura ambiente, debido a la potencia disipada por efecto Joule. Gracias al sensor de temperatura presente en la IMU es posible caracterizar la temperatura de funcionamiento normal de la IMU, que ronda los $35^{\circ}C$. Dado que la calibración fue realizada a una temperatura sensiblemente inferior, y para lograr robustez frente a eventuales cambios de temperatura, se realiza una compensación por temperatura de la lectura otorgada por el acelerómetro.

Para ello se calientan los sensores hasta una temperatura aproximada de $48^{\circ}C$, se lo deja enfriar hasta los $35^{\circ}C$ y se analiza la variación de las medidas con la temperatura. Al realizar dicho experimento se obtienen los resultados mostrados en la figura 7.5.

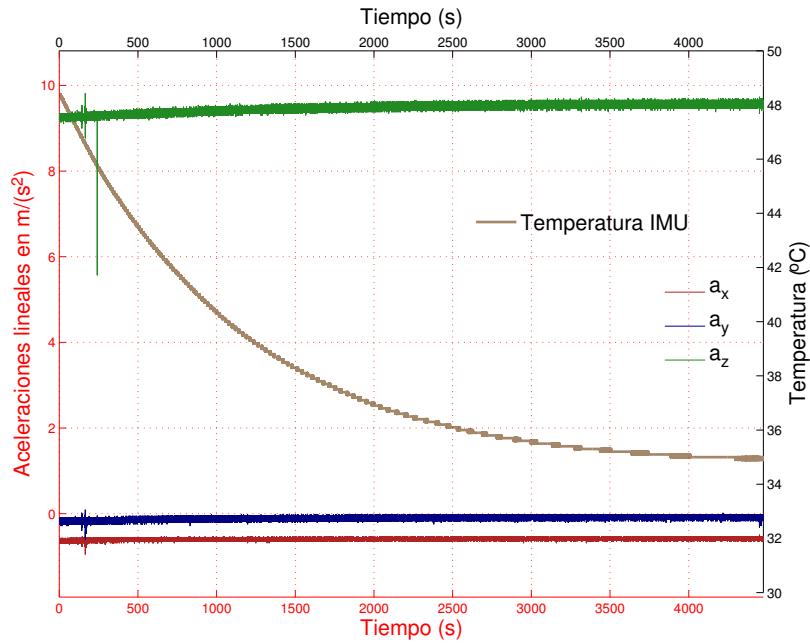


Figura 7.5.: Variación de la medida del acelerómetro con la temperatura

En la misma gráfica se muestran las medidas obtenidas de los sensores y la temperatura en función del tiempo. Los ejes rojos corresponden a la aceleración mientras que los ejes en negro corresponden a la temperatura.

De la hoja de datos del acelerómetro se puede extraer la siguiente información:

0 g Offset vs. Temperature for X-, Y-Axes: $\pm 0,4 \frac{mg}{^{\circ}C}$

0 g Offset vs. Temperature for Z-Axis: $\pm 1,2 \frac{mg}{^{\circ}C}$

la cual indica que la variación con la temperatura de la medida otorgada en z es 3 veces superior que la variación en los otros dos ejes y se puede caracterizar fundamentalmente con un offset que depende de la temperatura. Por ello se propone una compensación por temperatura utilizando un offset solamente para el eje z, de la forma:

$$b = b_a + \alpha(t - t_o)$$

Recordando la ecuación 7.3, se puede actualizar el modelo del acelerómetro al siguiente modelo:

$$\tilde{\mathbf{a}}^a = K_a(T_a^p)^{-1}\mathbf{a}^p + b_a + \begin{pmatrix} 0 \\ 0 \\ \alpha(t - t_o) \end{pmatrix} \quad (7.8)$$

donde t es la temperatura instantánea y t_o la temperatura para la cual el acelerómetro fue calibrado (aproximadamente $27^{\circ}C$).

Los resultados obtenidos luego de la compensación por temperatura se muestran en la figura 7.6.

7. Calibración del acelerómetro

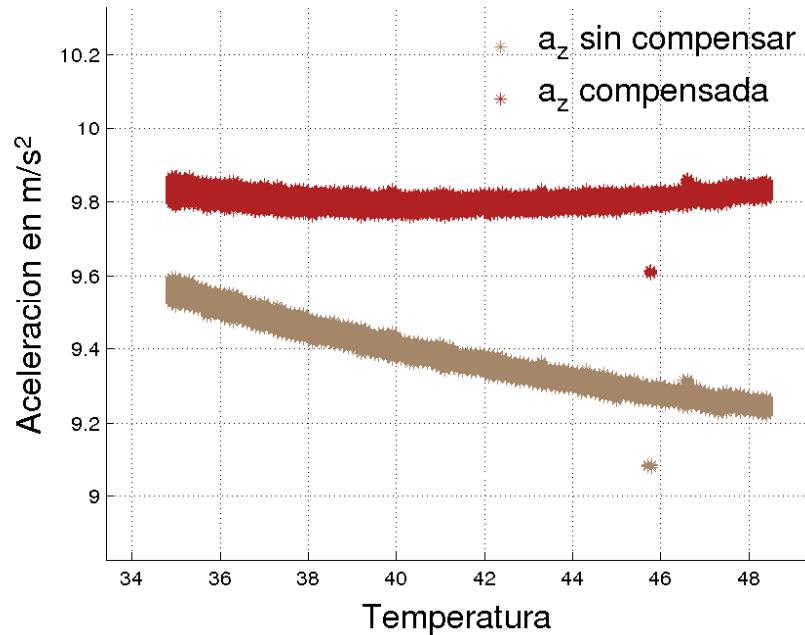


Figura 7.6.: Resultado de la compensación por temperatura

En la prueba mostrada en dicha figura se coloca al acelerómetro en forma horizontal, por lo que debería arrojar una lectura cercana a $9,81\frac{\text{m}}{\text{s}^2}$. En las medidas sin compensación se puede observar el carácter lineal de la dependencia con la temperatura. A su vez puede decirse que el error introducido por la temperatura en las medidas es corregido en buena forma por el modelo propuesto, ya que se obtienen valores de aceleración cercanos a los esperados en todo el rango de temperaturas estudiados.

CAPÍTULO 8

CALIBRACIÓN DEL GIRÓSCOPO

8.1. Objetivos

- Realizar una serie de pruebas con el fin de calibrar el giróscopo de tres ejes de la IMU.
- Estudiar, proponer y validar un modelo para dicha calibración.
- Estimar las fuentes de error del mismo (ruido y bias).

8.2. Materiales

- Mongoose 9Dof IMU de *Ckdevices*
- Beagleboard XM
- Adaptador Wi-Fi
- Tocadiscos
- Cronómetro
- Cubo perfecto de madera
- 2 escuadras de 45°
- 2 escuadras de 30°
- Madera de 4 cm de ancho con lados paralelos

8.3. Marco teórico

Para la calibración del giróscopo se utiliza el mismo modelo que se utilizó para el acelerómetro en la sección 7.

$$\tilde{\mathbf{w}}^a = K_w (T_a^p)^{-1} \mathbf{w}^p + \mathbf{b}_w \quad (8.1)$$

8. Calibración del giróscopo

con K_w matriz diagonal que representa el factor de escala para convertir del valor digital a la velocidad angular correspondiente, \mathbf{b}_w un término independiente para corregir la medida de velocidad angular nula y T_a^p una matriz que corrige la no ortogonalidad de los ejes:

$$T_a^p = \begin{pmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{pmatrix}$$

8.4. Procedimiento

8.4.1. Caracterización de las no idealidades variables

Para obtener más información sobre el *ruido inherente* y el *drift aleatorio* se toman datos durante una hora a una frecuencia de 100 Hz con el dispositivo quieto.

8.4.2. Determinación de parámetros estáticos

Para poder realizar la calibración pertinente es necesario determinar 12 parámetros: las ganancias y bias de los 3 ejes y los 6 ángulos de la matriz T_a^p . Para determinar dichos parámetros es conveniente obtener el doble o triple de medidas que de parámetros. Las medidas a realizar son las siguientes:

Con un total de 9 configuraciones diferentes, donde cada una de ellas aporta 3 medidas (una por cada eje), se utilizarán un total de 27 medidas para determinar los 12 parámetros involucrados.

Preparación



Figura 8.1.: Escuadras

zando las escuadras y la madera de 4 cm de ancho y lados paralelos, como se muestra en la figura 8.1. Apoyando el cubo sobre la escuadra se logran los ángulos de 30° y 45° deseados en las direcciones de giro secundarias.

EGP	EGS	AG
x	z	0
x	z	30
x	z	45
y	x	0
y	x	30
y	x	45
z	y	0
z	y	30
z	y	45

Cuadro 8.1.: Configuraciones utilizadas para calibrar el giróscopo.

Las medidas consisten básicamente en rotar la IMU a la velocidad del tocadiscos en las posiciones listadas en la tabla 8.1, donde **EGP** es el eje de giro principal, **EGS** es el eje de giro secundario, y **AG** es el ángulo de giro.

Con el cubo de madera se resuelven las rotaciones de 90 grados para alinear los tres ejes principales de la IMU con el eje de giro del tocadiscos. Los giros en el eje secundario se realizan utilizando

Si bien la velocidad de giro estándar del tocadiscos es 33.3 rpm, resulta conveniente medir la velocidad del tocadiscos para corroborarla. Se mide la velocidad con el dispositivo IR y se verifica el buen funcionamiento del tocadiscos.

Un esquema completo de la configuración utilizada para la calibración del girósopo se puede ver en la figura 8.2.

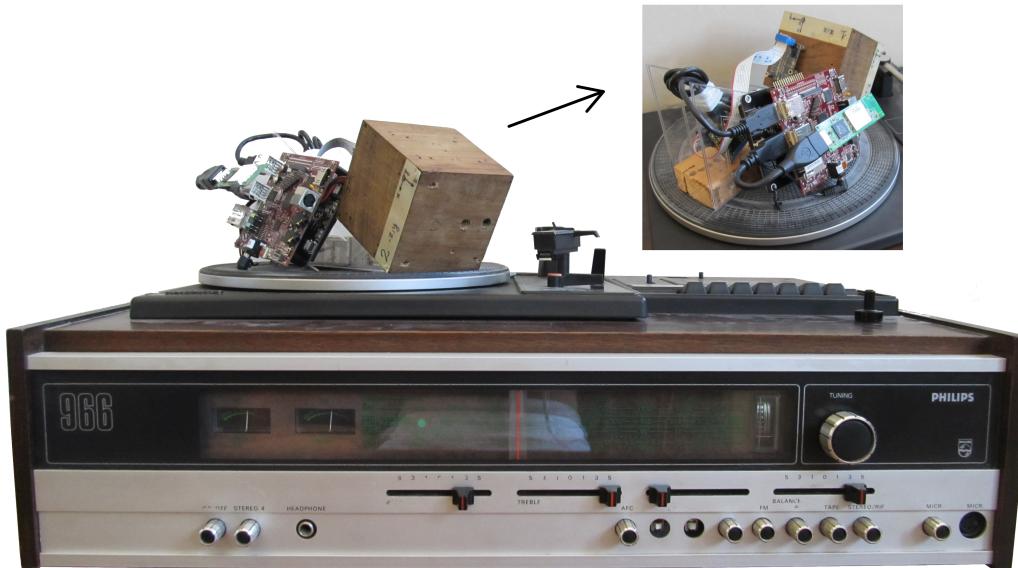


Figura 8.2.: Tocadiscos

8.5. Resultados y análisis

8.5.1. Medias para la determinación de los parámetros estáticos del giróscopo

Se pone el tocadiscos a girar y mediante la conexión Wi-Fi se indica a la *BeagleBoard XM* que empiece y termine de logear los datos. Se trabaja con el promedio de las velocidades angulares obtenidas en series de datos de 20 segundos controlados por cronómetro. La velocidad del tocadiscos permanece constante en todas las pruebas y lo que se varía es el ángulo del plano donde se apoya el cubo, haciendo uso de las escuadras (figura 8.1).

La IMU arroja datos enteros en complemento a dos. La hoja de datos del fabricante asegura que la relación entre bits y *grados/segundo* es: $14,375 \text{ LSB}/(\text{°}/\text{s})$. Como se desea trabajar con velocidades angulares en *rad/s* se tiene que la ganancia será $\approx 823,6$.

Al igual que con la calibración del acelerómetro (capítulo 7), se utiliza la función *lsqnonlin* de *Matlab* para ajustar las medidas obtenidas contra los valores teóricos para cada configuración. De acuerdo a lo anterior, se supone una semilla para las ganancias en los 3 ejes es el vector $G = [823,6; 823,6; 823,6]$. A su vez se supone

8. Calibración del giroscopio

que cada offset de cada eje será cercano a 0 y que los ángulos debidos a la no-ortogonalidad de los ejes es despreciable, se llega a que una buena semilla para la función *lsqnonlin* es la siguiente:

$$\theta_0 = (823,6, 823,6, 823,6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$

Los parámetros que minimizan la suma de los errores al cuadrado son los siguientes:

$$\theta = \begin{bmatrix} 796.6646 & 808.2419 & 803.4482 & -25.3819 & -15.7210 & -1.8871 \\ 0.0093 & 0.0659 & 0.0093 & 0.0128 & -0.0250 & 0.0552 \end{bmatrix} \quad (8.2)$$

Es interesante destacar que el vector θ hallado es similar a la semilla propuesta, por lo que podemos afirmar que el mínimo local encontrado por la función *lsqnonlin* es el deseado.

Una buena medida de los resultados obtenidos se puede obtener al analizar el promedio de los errores cometidos (μ) y la desviación estándar (σ):

$$\mu = 0,01877 \quad (\text{rad/s}) \quad (8.3)$$

$$\sigma = 0,0231 \quad (\text{rad/s}) \quad (8.4)$$

El error promedio corresponde a una velocidad angular de $1,07^\circ/\text{s}$ y 2σ corresponde a una velocidad angular de $2,64^\circ/\text{s}$. Estos valores son suficientemente pequeños en comparación con las velocidades angulares típicas a las cuales se encontrará sometido el cuadricóptero y por ende puede considerarse que la calibración realizada es aceptable.

8.5.2. Variación con la temperatura

Como fue explicado en el capítulo 7, la temperatura de trabajo de la IMU es significativamente mayor que la temperatura para la cual fueron calibrados los sensores. Para aplicar una compensación por temperatura a la medida del giroscopio, se utiliza un método idéntico al utilizado para el acelerómetro: se varía la temperatura desde una temperatura aproximada de 48°C hasta 35°C y se releva la curva de velocidad angular medida contra temperatura.

Al modelo utilizado para el acelerómetro se le agrega un término de offset independiente de la temperatura, que se introduce para tomar en cuenta que la calibración en el tocadiscos puede no haber eliminado completamente el offset.

Recordando la ecuación 8.1, el nuevo modelo para la velocidad angular es el siguiente:

$$\tilde{\mathbf{w}}^a = K_w(T_a^p)^{-1}\mathbf{w}^p + \mathbf{b}_w + \mathbf{b}_1(t - t_0) + \mathbf{b}'_0$$

simplemente definiendo $\mathbf{b}_0 = \mathbf{b}_w + \mathbf{b}'_0$:

$$\tilde{\mathbf{w}}^a = K_w(T_a^p)^{-1}\mathbf{w}^p + \mathbf{b}_0 + \mathbf{b}_1(t - t_0)$$

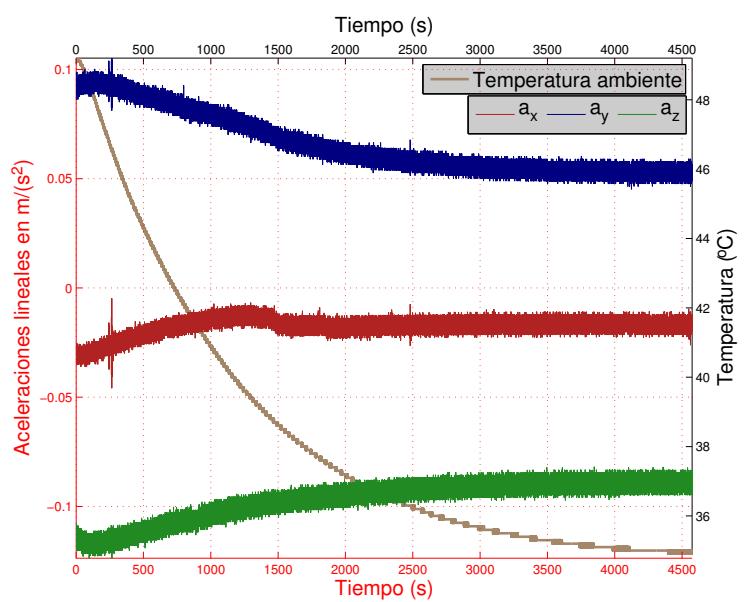


Figura 8.3.: Variación de la velocidad angular con la temperatura

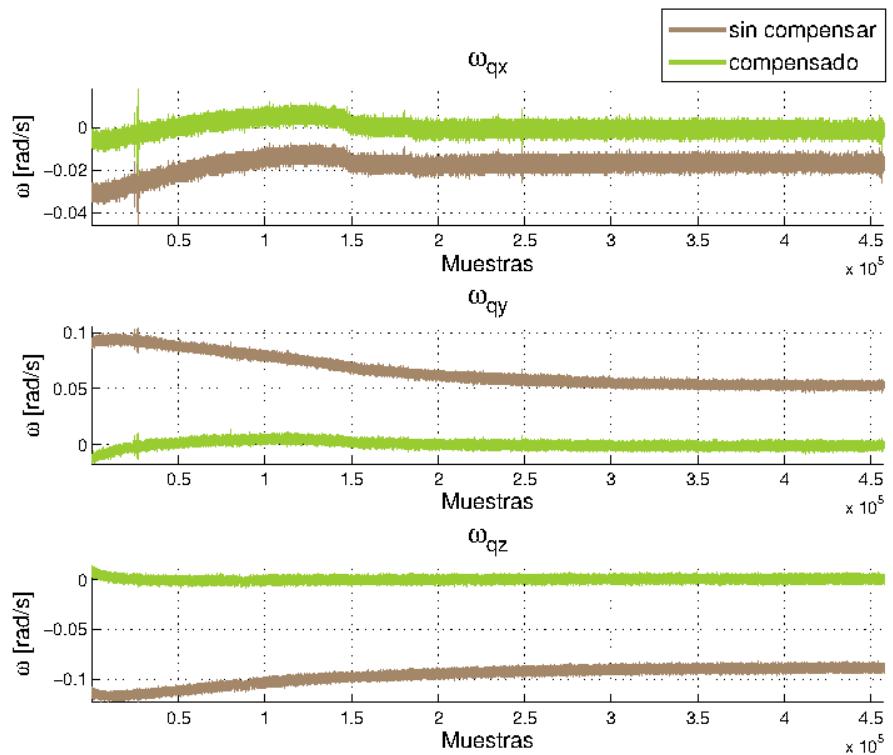


Figura 8.4.: Velocidad angular compensada por temperatura

CAPÍTULO 9

CALIBRACIÓN DEL MAGNETÓMETRO

9.1. Objetivos

El objetivo de estas pruebas es comprender y caracterizar el magnetómetro de 3 ejes Honeywell HMC5583, incorporado para asistir en la determinación de la orientación absoluta del cuadricóptero.

9.2. Procedimiento

El modelo adoptado para relacionar las medidas de campo magnético sin calibrar con las medidas calibradas es idéntico al utilizado a la hora de calibrar el acelerómetro y el giróscopo. Se trata de realizar una transformación lineal de las medidas obtenidas para transformarlas en los valores de campo magnético. Se propone un modelo de la forma:

$$\mathbf{C}^p = K_m(\tilde{\mathbf{C}}^m - \mathbf{b}_m)$$

Las principales distorsiones que sufre un magnetómetro son debidas a los efectos de *hard iron* y *soft iron*. El primero es debido a la presencia de imanes permanentes, mientras que el segundo se debe a la distorsión producida por elementos metálicos como por ejemplo tornillos o conectores. Idealmente el campo magnético terrestre medido en diferentes direcciones es constante, por lo tanto la representación de dichas medidas es una esfera centrada en el origen. Sin embargo, el resultado de los efectos mencionados es que la representación sea una elipsoide (debido al efecto de *soft iron*) centrada en un punto diferente al origen (debido al efecto de *hard iron*). Por esta razón la calibración del sensor, a diferencia de la realizada para el acelerómetro y el giróscopo, debe ser dentro del cuadricóptero con todos los sistemas operando en condiciones normales¹ a fin de poder compensar dicho efecto.

El método de calibración propuesto en [6] consiste en tomar una serie de medidas de campo magnético terrestre en la mayor cantidad de orientaciones posibles. Para

¹Incluso con los motores encendidos. Se quitaron las hélices.

9. Calibración del Magnetómetro

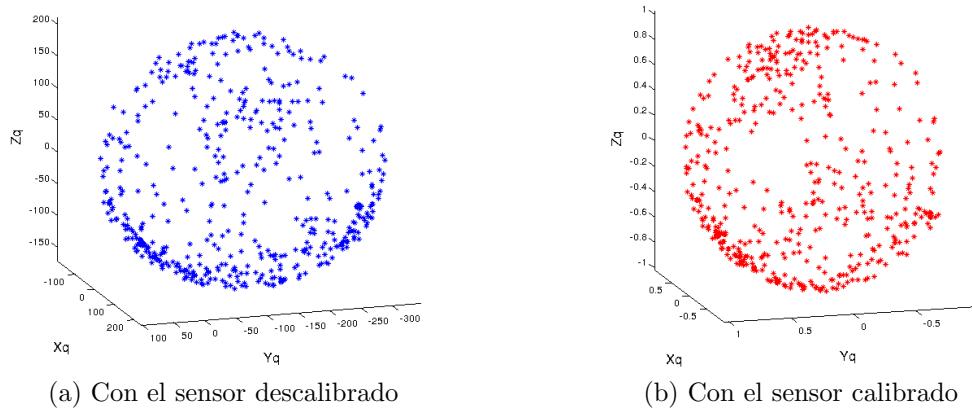


Figura 9.1.: Medias de campo magnético Terrestre

asegurar la calidad de los datos es recomendable tomar medidas distribuidas uniformemente en todas las direcciones. Se utilizó un algoritmo desarrollado por [7] para realizar la calibración, dicho algoritmo no es otra cosa que la implementación de la minimización presente en el método propuesto por [6]. El resultado que se obtiene es la matriz y el vector que hacen que las medidas tomadas aproximen una esfera de centro el origen y radio unitario.

9.3. Resultados y Análisis

Los parámetros obtenidos son los siguientes:

$$\begin{aligned} \blacksquare \quad K_m &= \begin{pmatrix} 4,08 \times 10^{-3} & 2,54 \times 10^{-5} & 5,36 \times 10^{-4} \\ 04,04 \times 10^{-3} - 1,13 \times 10^{-4} & \\ 004,77 \times 10^{-3} & \end{pmatrix} \\ \blacksquare \quad \mathbf{b}_m &= \begin{pmatrix} -63,16 \\ -209,61 \\ -38,62 \end{pmatrix} \end{aligned}$$

En la figura 9.1a se observan graficadas las medidas de campo magnético con el sensor descalibrado en los ejes solidarios a la plataforma. Como era de esperarse, debido a los efectos nombrados anteriormente, con el sensor descalibrado no se obtiene una esfera. En la figura 9.1b tenemos las medidas de campo magnético con el sensor calibrado. La magnitud que medimos luego del proceso de calibración se encuentra normalizada. Dado que utilizaremos dicho sensor para determinar una orientación, nos interesan simplemente las relaciones entre las componentes medidas en cada eje del sensor, por lo tanto trabajar con las medidas normalizadas arroja el mismo resultado que trabajar con las medidas de campo magnético expresadas en Teslas o Gauss.

Para verificar que la calibración es exitosa se debe analizar con mayor detalle las características de la “esfera” obtenida. Con los datos convertidos se procede a realizar

9.3. Resultados y Análisis

una minimización gracias a los mínimos cuadrados para obtener las coordenadas del centro de la esfera y su radio.

Los resultados que se obtienen son:

- Centro de la esfera: $(-3,65 \times 10^{-3}; -9,40 \times 10^{-3}; 1,67 \times 10^{-3})$
- Radio de la esfera: 0,996
- Desviación estándar de la medida del radio: $\sigma = 1,54 \times 10^{-2}$

A partir de dichos resultados se puede concluir que el resultado de la calibración es exitoso, ya que la representación de las medidas es prácticamente una esfera de radio 1 centrada en el origen.

CAPÍTULO 10

GPS

10.1. Objetivos

En este capítulo se analiza la performance del GPS Canmore GT-730F. Se intenta reconstruir un polígono, y se analiza el error al estimar la posición de un punto fijo.

10.2. Materiales

- GPS Canmore GT-730F.
- Laptop.
- Trípode (de fotografía).
- Cinta métrica, pintura y cuerda.

10.3. Procedimiento

En esta sección se caracteriza el error del GPS fundamentalmente en latitud y longitud. Adicionalmente se realiza un análisis del error del GPS en altura.

El experimento que se diseñó consiste en marcar un rectángulo sobre el suelo, utilizando 6 puntos, con la disposición de la figura 10.1. Todas líneas punteadas son de 6m de largo. Resulta en un rectángulo de 6m por 12m.

Los pasos a seguir son los siguientes:

1. Construir el rectángulo sobre un superficie plana.
 - Se utilizó pintura para marcar los vértices del triángulo.
 - Para trazar uno de los lados de 12 metros (puntos 1,2 y 3), se fijó una cuerda de 12 metros (con el punto medio marcado) a un punto, y se la extendió (sin estirarla). El principio (1) y el final (3) de la cuerda son vértices del polígono, y el punto medio (2) es otro de los puntos de interés.

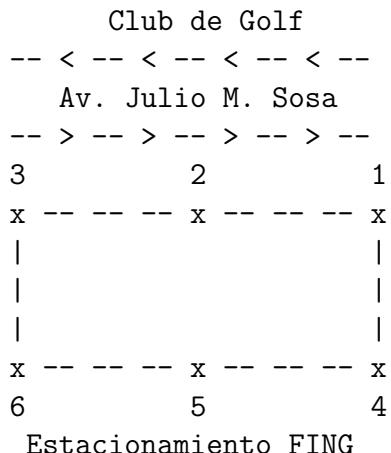


Figura 10.1.: Polígono

10. GPS

- Para construir rectas perpendiculares se utilizó una cuerda de 6m, y otra de 8.5m¹. Uno de los extremos de la cuerda de 6 metros se fijó al 1, y uno de los extremos de la cuerda de 8.5m se fijó a 2. El punto donde ambas se intersectan corresponde a 4. Un procedimiento similar se siguió para determinar la ubicación de 5 y 6.
2. Medir, con un metro, las distancias entre todos los puntos.
 3. Utilizar mínimos cuadrados para minimizar el error entre las distancias esperadas, y las experimentales. Esto puede llevar a trabajar con un polígono que **no** sea un rectángulo, pero el error será menor que el que resultaría de usar los valores teóricos.
 4. Fijar la altura y la orientación del GPS, y tomar medidas en cada uno de los puntos [1,2,3,4,5,6].
 5. Tomar un punto como origen, y comparar la figura que resulta de los datos provenientes del GPS con las medidas tomadas con el metro.

En la figura 10.2a se observa el trípode que sostiene al GPS. Se busca tener el GPS a una altura fija, y separado del piso. Al nivel del piso los rebotes degradan seriamente la performance del GPS. La cuerda que marca el lado del polígono, junto con las patas del trípode, se utilizaron para fijar la orientación del GPS durante el experimento.



(a) (b)
Figura 10.2.: GPS y Atrial

10.3.1. Adquisición de datos

Para tomar datos se conectó el GPS, que envía información mediante USB-serie, al puerto USB de una computadora. En la computadora se utilizó *GPSD*², un software open-source que hace de *daemon*, y se encarga de transformar los datos del GPS a una estructura fácil de manejar en C.

El GPS emite sentencias NMEA a través de un puerto USB serie, a 38400bps. Trabaja con sentencias:

- GPGGA - *Global Positioning System Fix Data*: Información sobre la fecha, posición y datos relevantes sobre el *fix*.
- GPGSV - *Satellites in view*: Información sobre la cantidad de satélites detectados, y la calidad de la señal proveniente de cada uno.
- GPGSA - *DOP and active satellites*: Información sobre DOP³ y satélites activos.

¹ $8,5 \approx \sqrt{6^2 + 6^2} = 8,4852\dots$

²<http://catb.org/gpsd/>

³*Dilution of precision* - Ver E.1.

- GPRMC - *Recommended minimum specific GPS/Transit data*: Sentencia mínima recomendada, trae suficiente información como para poder trabajar con el GPS.
- GPVTG - *Track Made Good and Ground Speed*: Información sobre la dirección y la velocidad (**no** sobre la posición absoluta).

10.3.2. Verificación del polígono

Una vez construido el polígono, es de interés medir todas las diagonales (con la cinta métrica) por dos motivos:

- Verificar que no se cometieron errores.
- Hacer mínimos cuadrados con las medidas, de manera de obtener un polígono, que no tiene porqué ser (y en general no será) un rectángulo, sino algo similar a un rectángulo, más ajustado a la realidad.

D12	D13	D14	D15	D16
603	1205	606	855	1345
D23	D24	D25	D26	D34
603	853	608	853	1344
D35	D36	D45	D46	D56
850	602	602	1202	603

Las medidas tomadas se resumen en la tabla 10.1, donde D12 representa la medida de la recta que une el punto 1 con el punto 2, en cm.

El polígono resultante se observa en la figura 10.3. El error relativo, es decir, el cociente entre las medidas de cada recta **DAB** (distancia entre el punto A y el B) resultante de aplicar MC, y lo esperado en el polígono teórico, de 6m de lado, es:

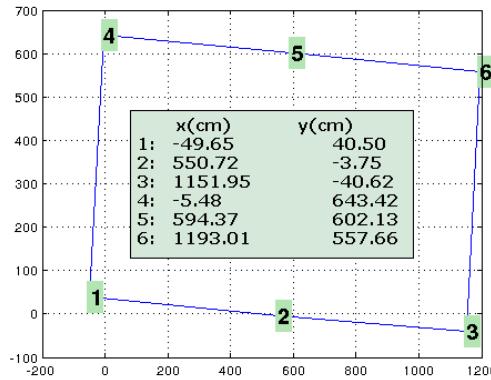


Figura 10.3.: Polígono luego de MC.

- 9,51 % en el peor caso.
- 4,65 % en promedio.

Se puede trabajar con el polígono irregular que fue efectivamente construido en el resto del experimento, pero la interpretación de los resultados se torna poco intuitiva. Considerando que el error es inferior al 10 % en el peor caso, se opta por continuar el experimento considerando que el polígono es un rectángulo de lados 12 y 6 metros.

10.3.3. Punto fijo - 2 minutos

Se tomaron datos durante aproximadamente 2 minutos (≈ 120 muestras) en cada uno de los vértices del polígono, con el objetivo de observar la estabilidad de la información proveniente del GPS.

10. GPS

En las figuras 10.6 y 10.5 se muestra el error en los datos del GPS respecto al valor promedio. Si este careciera de error relativo, todas las muestras coincidirían con el promedio, y estarían ubicadas en el punto [0,0]. En dichas figuras la circunferencia negra es de 2.5m de radio. En la leyenda se muestra qué porcentaje de las muestras se encuentran fuera del círculo. Se Repite el experimento tomando datos durante 10 minutos por punto obteniendo resultados similares. Se orientó el GPS de 3 maneras distintas, siempre alineando el trípode con uno de los lados de 12m del rectángulo:



(a) Orientación #3 (b) Orientación #1 y #4.

Figura 10.4.: Orientaciones del GPS

1. USB hacia la calle, LED hacia el estacionamiento, como en la figura 10.4b.
2. USB hacia la rambla, LED hacia el IIE (no hay figura)
3. Como en la figura 10.4a.
4. Nuevamente, USB hacia la calle, LED hacia el estacionamiento, como en la figura 10.4b.

Punto fijo: Análisis - satélites disponibles

La teoría dice que con 4 satélites debería alcanzar para obtener un *fix 3D*, es decir, estimar la posición sobre la esfera terrestre, y la distancia (altura) a la misma. Se recomienda tener no menos de 6 satélites. Durante el experimento de la figura 10.6c, hubo un período de tiempo en el cual el GPS perdió la señal, y el número de satélites disponibles, que usualmente es 9 o 10, pasó a ser 4. Los datos correspondientes se muestran

en la figura 10.6. El trazo naranja, con un error de hasta 23 metros, corresponde a instantes donde la cantidad de satélites era entre 4 y 5. Luego de volver a 9 o 10 satélites, los datos tienen errores razonables.

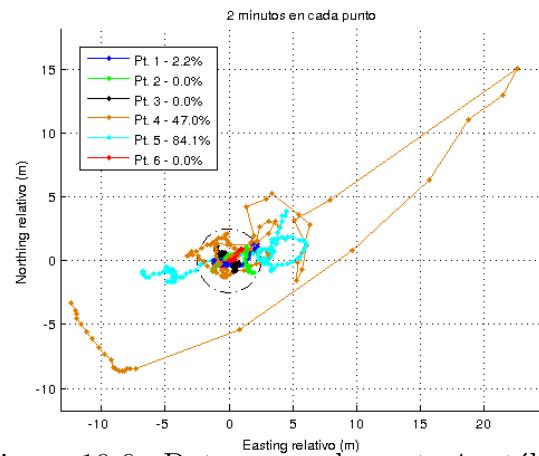
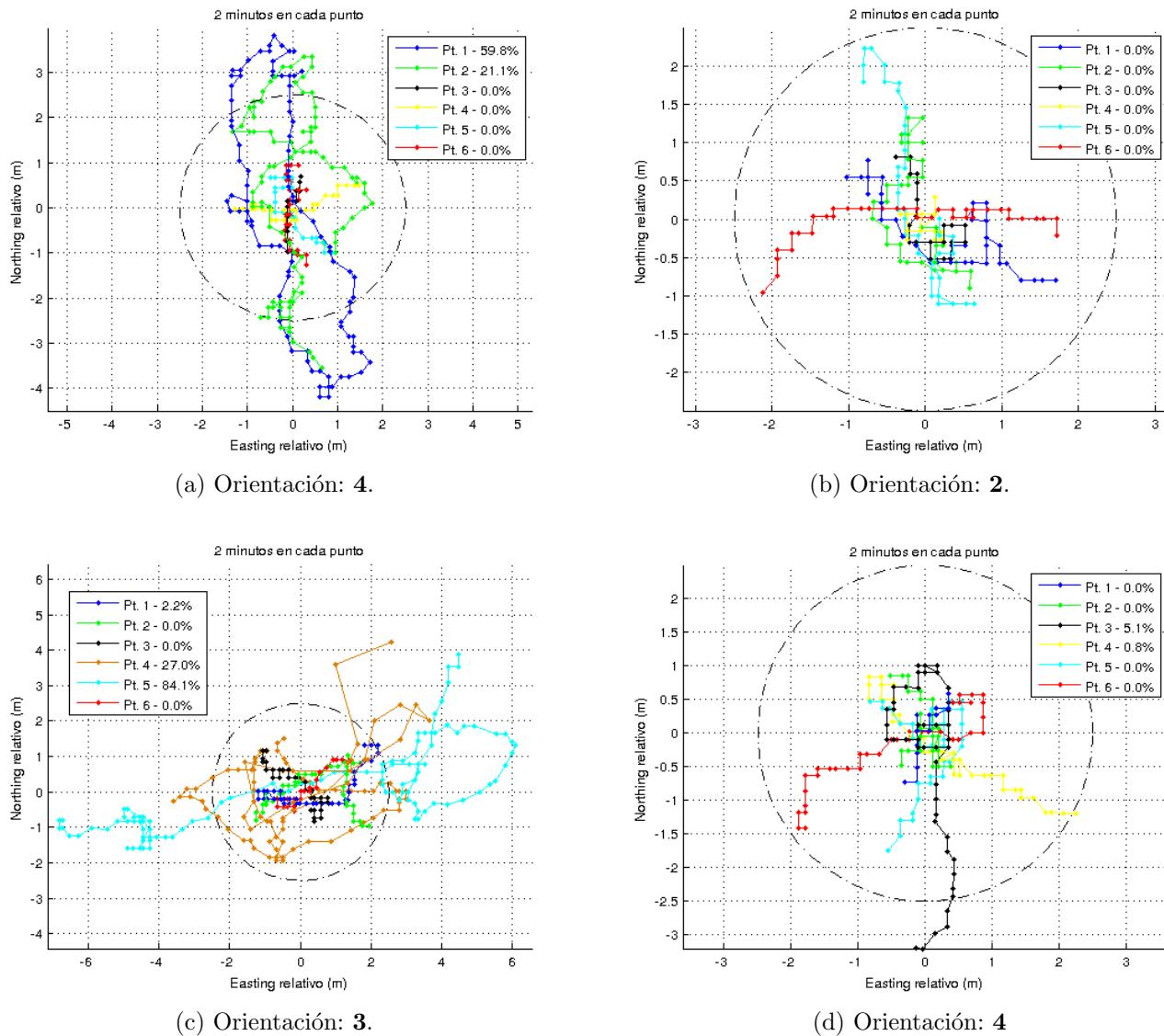


Figura 10.6.: Datos con solamente 4 satélites. Orientación: 3.

En la figura 10.6c se observa el mismo log que en 10.6, pero luego de haber quitado las muestras correspondientes al período donde se deterioró la señal. No se pudo encontrar una explicación para la mala calidad de las muestras correspondientes al punto 5 en la figura 10.6c. La cantidad de satélites disponibles se mantuvo estable en 9 o 10 durante la adquisición de los datos.

Figura 10.5.: Punto fijo - 2 minutos



Punto fijo: Análisis - Orientación

Para evaluar si existe una correlación entre la orientación y las medidas del GPS, se hizo el siguiente experimento:

1. Tomar datos durante 10 minutos con el GPS arriba del trípode, dos patas alineadas con una recta fija.
2. Rotar el trípode 120 grados en sentido horario, de manera que otro lado del triángulo que forman las patas del trípode quede alineado con la recta. Tomar datos durante 10 minutos.
3. Rotar y tomar datos nuevamente.

Los resultados del experimento se observan en la figura.

Orientación - Conclusiones

No se encontró una correlación entre la orientación del GPS y el error en las medidas. El experimento se realizó con cielo abierto, con una buena geometría, en

10. GPS

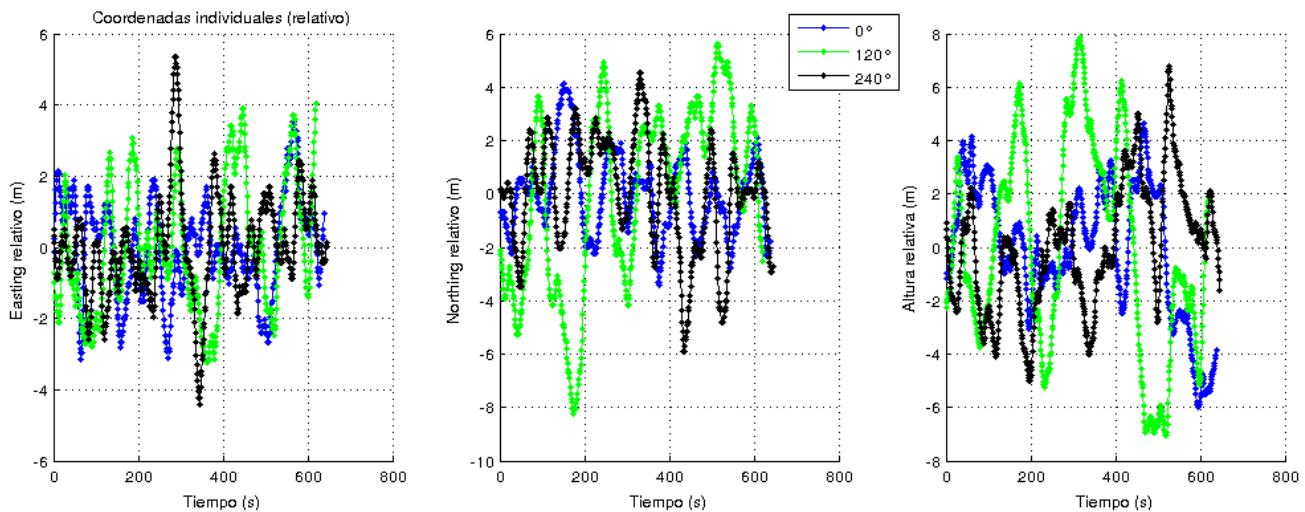


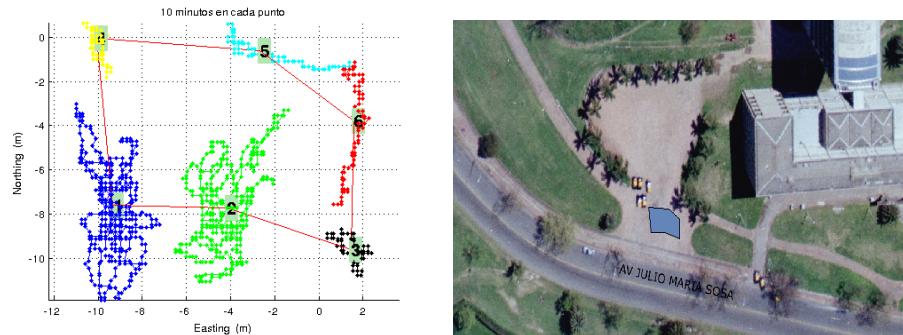
Figura 10.7.: Datos rotando el GPS sobre un punto fijo.

términos de distribución de satélites. Tal vez en situaciones de visibilidad limitada se podría observar una correlación, ya que si la dirección en la que la antena recibe mejor coincide con el lugar donde hay pocos satélites, entonces la cantidad de información sería menor/peor que en otras orientaciones.

Tener visibilidad limitada por obstáculos, o tener una mala geometría deteriora la performance del GPS. En este trabajo no se considerará la situación en la cual se tiene una performance deficiente, o bien se utilizará el GPS a cielo abierto, o se trabajará bajo techo (sin GPS).

10.3.4. Polígono

En la figura 10.8a, la línea en roja representa el polígono resultante de unir el promedio en cada vértice de las muestras tomadas durante un experimento de 10 minutos. En la figura 10.8b se dicho polígono, proyectado sobre una foto satelital⁴.



(a) Polígono formado por los promedios de 10 minutos.
(b) Proyección del polígono sobre una foto satelital.

Figura 10.8.: Polígono

En las siguientes figuras se observan los polígonos formados por los promedios de varias secuencias de 2 minutos por punto.

⁴El mapa y las fotos se obtuvieron de:
<http://sig.montevideo.gub.uy/mapas/mapa-principal>

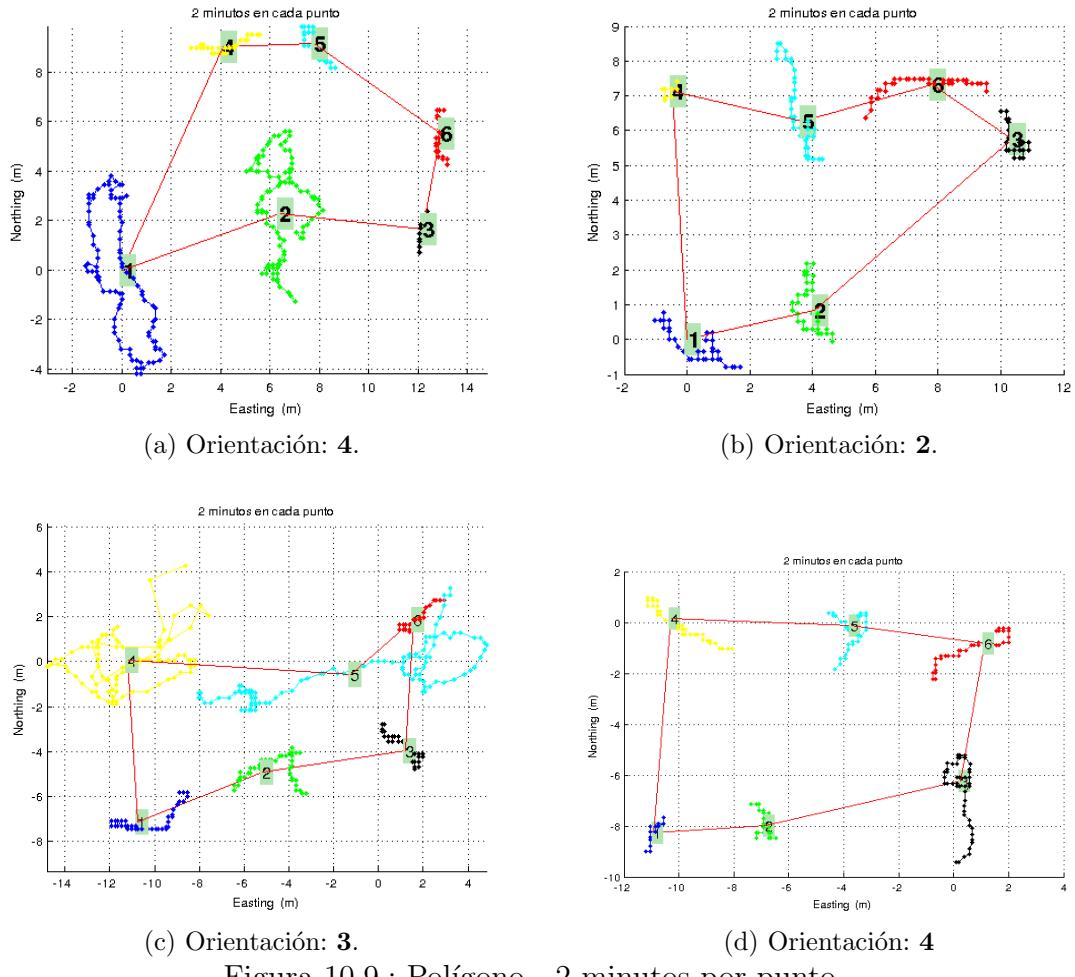


Figura 10.9.: Polígono - 2 minutos por punto

10.3.5. Polígono - Análisis

Dado que no se cuenta con una referencia absoluta, como podría serlo un GPS de alto nivel, no es posible hablar de error absoluto. De cualquier forma, con la información disponible es posible obtener un estimador del error típico en una distancia de X metros, donde X pertenece al conjunto de las medidas de las rectas en juego: $\{6; 8.5; 12; 13.45\}$ metros. Se comparan las distancias entre los vértices del polígono experimental, con las distancias entre las coordenadas resultantes de la aplicación de mínimos cuadrados a las medidas efectuadas con el metro (ver la sección 10.3.2).

El error relativo para cada recta resulta:

$$E = \left| \|\hat{P}_i - \hat{P}_j\| - D_{ij} \right| \quad (10.1)$$

Donde:

- \hat{P}_i es la dupla $\{x, y\}$, coordenadas experimentales del i -ésimo vértice del polígono.
- D_{ij} es la distancia entre el i -ésimo y el j -ésimo vértice, resultado de la aplicación de MC en la sección 10.3.2.

Para cada set de estimaciones de la posición de los vértices del polígono⁵, se utilizó la fórmula 10.1, separando los datos según el largo esperado de la recta⁶.

⁵Un set de valores $\hat{P}_1^1, \hat{P}_2^1, \hat{P}_3^1, \hat{P}_4^1, \hat{P}_5^1, \hat{P}_6^1$

⁶Valores posibles: $\{6; 8.5; 12; 13.45\}$

10. GPS

Resulta un set de valores del “*error al estimar la distancia entre dos puntos que deberían estar a X metros*”, para cada valor de X.

Los resultados se resumen en la tabla 10.2.

Largo Teo. (cm)	μ (cm)	σ (cm)	$\frac{\mu+2\sigma}{\text{Largo Teo.}}$	# Muestras
600	372.15	169.39	118 %	35
848	493.6	165.79	97 %	20
1200	429.67	212.81	71 %	10
1341	614.73	166.71	70 %	10

Cuadro 10.2.

En la tabla 10.2 se observa, cómo es de esperarse, que el error relativo disminuye al intentar estimar distancias mayores. Asumiendo que el error en la estimación de la posición de cada punto A, B es similar, entonces para cometer un error de P % en la estimación de una distancia X (medida con el metro) entre A y B, el GPS debe equivocarse en $\frac{P}{100}X\frac{1}{2}$ en la estimación de la posición de A y B.

Resulta que el error en la estimación de la distancia entre dos puntos es aproximadamente 4.5m. Esto es coherente con los resultados sobre la estimación de un punto fijo. Si las medidas para la posición de un punto fijo en general caen dentro de un círculo de 2.5m de radio, entonces, el error en el peor caso en la distancia es de 5m.

Nuevamente, no se cuenta con información sobre la posición real, pero parece razonable asumir que la ubicación real es cercana al promedio de los datos provenientes del GPS. Evidencia a favor de esto se observa en la figura 10.8b. En la sección 10.8 se profundiza en este aspecto.

10.4. Caminata por el borde del polígono

Para simular una situación más parecida a la que se tendrá con el GPS montado sobre el cuadricóptero, se colocó el trípode con el GPS en un mochila, y se procedió a recorrer el polígono con con la mochila puesta.

Los errores en la medida de un punto fijo hacen pensar que sería posible que la trayectoria determinada por el GPS luego de este experimento fuese similar a un rectángulo.

En las figuras siguientes se observan los datos tomados durante 3 caminatas siguiendo el contorno del polígono, siguiendo la secuencia 1-2-3-6-5-4-1⁷. Se empezó a loguear datos en el punto 1, y se terminó nuevamente en el punto 1.

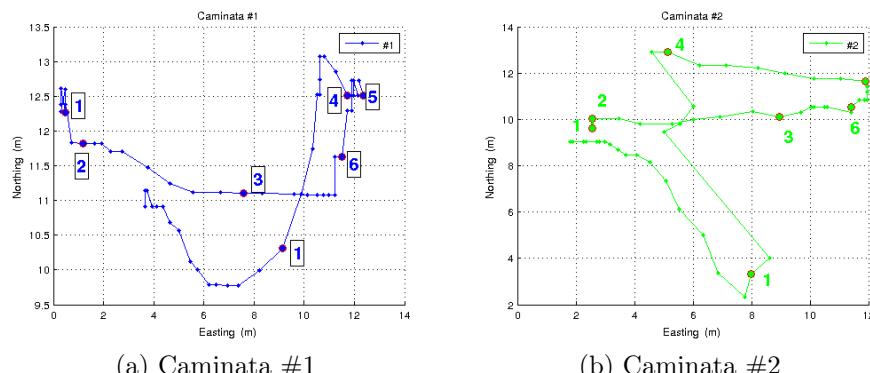


Figura 10.10.: Caminatas por el borde del polígono

⁷Notar que **no** se recorre en el mismo orden en el que se numeran los puntos del rectángulo

10.4.1. Caminata - Análisis

Las gráficas de las caminatas son muy desalentadoras. Se observa un error mayor al esperado a partir de los experimentos de punto fijo. Recorrer el polígono no es comparable a tomar muestras, quieto, en cada vértice.

En algunos de los experimentos la posición parece tender a estabilizarse una vez que se llega al destino final, el punto 1. Ahí se continuó logueando datos por unos 20 segundos. En la figura 10.10b la posición final parece tener un drift que se aleja de la posición inicial, en lugar de acercarse. No se encontró una justificación para este comportamiento.

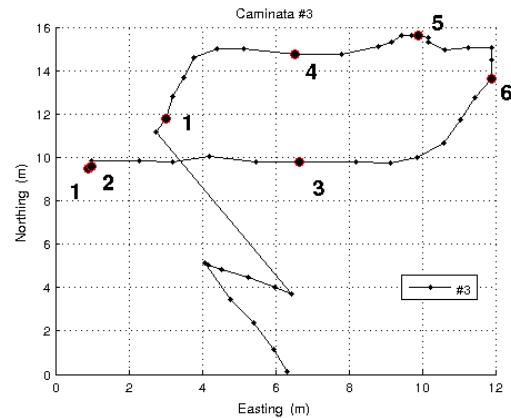


Figura 10.11.: Caminata #3

10.5. Error en altura

Para determinar el error la información sobre la altura que provee el GPS, se diseñó un experimento, que consiste en tomar medidas en una perpendicular a la esfera terrestre, a 4 alturas diferentes: 0m, 1m, 2m y 3m respecto al suelo.

Se mantuvo el GPS quieto en cada uno de los niveles, y se tomaron muestras durante aproximadamente 60 segundos. El objetivo de esta etapa era verificar si era viable el experimento.

Se colocó una escalera en el medio del estacionamiento de atrás de la Facultad de Ingeniería, se ató un piolín con marcas cada 1 metro, y una plomada en la punta para mantenerlo tenso y vertical.

En la figura 10.12 se observan los resultados del experimento. Es de esperarse que el error sea mayor al estar apoyado sobre el suelo, ya que los rebotes (*multipath*) pueden deteriorar el sistema. El error a 1m de altura es mayor al que se obtuvo con el GPS en el suelo, una posible explicación para esto sería que el GPS estaba muy cerca de la escalera metálica, lo cual podría introducir una cantidad significativa de rebotes.

Los resultados de este experimento llevan a pensar que el GPS da información estable si se encuentra a al menos 2m del suelo. Esto no parece ser un problema, ya que el cuadricóptero volará a alturas superiores.

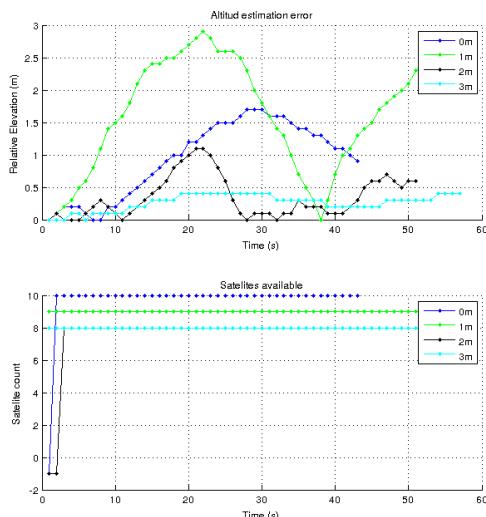


Figura 10.12.: Variación de la altura determinada por el GPS a distintas alturas.

10.6. Tiempo de *warmup*

El tiempo que demora el GPS en adquirir un *fix* depende de si estaba apagado, o si estaba previamente en funcionamiento.

- **Frío:** Aproximadamente 40 segundos.
- **Caliente:** Entre 3 y 5 segundos.

Esto implicaría que al prender el cuadricóptero, habría que esperar unos 40 segundos para obtener señal del GPS, y que si por algún motivo se pierde la conexión con el GPS, habrá que esperar entre 3 y 5 segundo antes de contar con datos útiles nuevamente.

10.7. Tasa de muestreo

El GPS bajo funcionamiento normal trabaja a aproximadamente 1Hz. Se observó un patrón de 3 muestras a 1Hz, luego una 4ta muestra pegada a la última de las 3, luego nuevamente a 1Hz.

Se observó que estando quieto, a veces repite información, y la etiqueta como “válida”. Se observó este comportamiento durante los experimentos de punto fijo, llegando a pasar 30 segundos registrando exactamente la misma información.

Se compararon los datos crudos provenientes del GPS con los que devuelve el GPSD, y se concluyó que el GPSD **no** es responsable de la repetición de datos, es decir, no es un problema de software, sino que el GPS es quien envía datos repetidos.

Al loguear trayectorias de movimiento permanente, como un recorrido en bicicleta de 15 minutos, no se observaron datos repetidos durante más de 4 segundos. No parece razonable que el cálculo de la posición dé *exactamente* lo mismo durante varias medidas sucesivas, probablemente se trate de un problema interno del GPS.

10.8. Posición Absoluta

Se realizaron varios experimentos para analizar si la ubicación que daba el GPS se correspondía con la realidad, ya que algunos GPSs a veces dan un error constante de varios metros, y de ser el caso, se podría considerar este offset.

Sobre un mapa de Montevideo se graficaron trayectorias conocidas, con el objetivo de verificar que los datos provenientes del GPS eran “razonables”.

No se observó un corrimiento de los datos evidente y constante, por lo que se asume el GPS **no** introduce un offset.

En la figura 10.13 se observa una trayectoria realizada en bicicleta, con el GPS colocado sobre el trípode, asomando de una mochila. El recorrido fue hacia el sur-oeste, sobre el cordón nor-oeste. Se observa que la posición dada por el GPS marca un camino similar al realizado, pero a veces llega a cruzar al cordón opuesto (esto no sucedió durante el experimento).

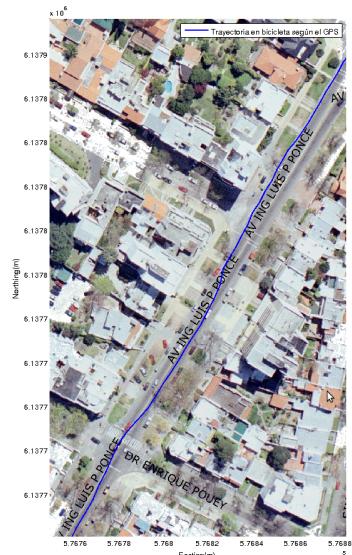


Figura 10.13.: Recorrido en bicicleta.

10.9. Conclusiones

Los siguientes items servirán de guía para la utilización del GPS como instrumento de navegación:

- **Error en las medidas:** Las medidas del GPS tienen un error típico menor a 2.5m para un punto fijo, situación irreal, ya que el cuadricóptero necesita al GPS para poder quedarse quieto. Para las caminatas el error es de alrededor de **5m**, siempre que se disponga de una buena geometría y de alrededor de 9-10 satélites a la vista. Es de esperarse que la performance se reduzca al utilizar el GPS montado sobre el cuadríptero, donde estará sujeto a interferencia electromagnética de los motores, vibraciones, etc.
- **Visibilidad:** Es importante asegurar buena visibilidad, la performance del GPS se reduce drásticamente si pierde satélites.
- **Resolución:** El GPS **no** es adecuado para tareas de navegación con restricciones menores a 10 metros, pero sí para distancias de varias decenas de metros.
- **Tasa de muestreo:** El GPS normalmente muestrea a **1Hz**. Cabe destacar que debe ignorarse información repetida (muestras sucesivas que sea idénticas).
- **Error en altura:** El GPS **no** es adecuado para determinar la elevación durante el aterrizaje/despegue, ya que a menos de 2m del piso, la estimación de la elevación es muy mala. Será necesario otro sensor (sensor de presión, IR, etc) para asistir durante el despegue/aterrizaje.
- **Tiempo de fix:** En frío demora alrededor de 40 segundos, en caliente entre 3 y 5 segundos.

CAPÍTULO 11

BARÓMETRO

Como instrumento para determinar la altura del cuadricóptero se pensaba utilizar un GPS, pero la frecuencia de muestreo del GPS y el error en los datos provenientes del mismo lo hacen insuficiente. Se opta por incorporar un barómetro que por sí solo también lo hacen insuficiente, dichos sensores padecen de *otros* problemas. Ambos sensores permiten obtener una buena estimación de la altura.

El sensor adquirido es un barómetro digital BOSCH BMP085. Permite medir presión absoluta y, conociendo la presión y la temperatura a nivel del mar, permite calcular la altura absoluta.

Tiene varios modos de funcionamiento, habiendo un compromiso entre el consumo, la tasa de actualización y la resolución.

11.1. Objetivos

El objetivo de estas pruebas es determinar como ha de usarse el barómetro BOSCH BMP085. Se analizan las siguientes situaciones:

- Reposo: Se analizan períodos de distinta duración, para cumplir con distintos objetivos:
 - Decenas de minutos: Caracterizar el drift y el tiempo de *warm-up*.
 - Decenas de segundos: Caracterización del ruido de las medidas.
- Altura relativa: 3 experimentos variando el rango a analizar: Puntos que distan decenas de centímetros entre sí, 1m entre sí, y aproximadamente 5m entre sí.

11.2. Materiales

- Laptop.
- Tanza.
- Cinta métrica.
- Cubo de lapacho.
- IMU “Mongoose” de CKDevices (cuenta con un BMP085).

11.3. Consideraciones previas

El sensor de presión devuelve la presión absoluta. Si se conoce la presión y la temperatura a nivel del mar, es posible calcular la elevación. A nivel de la tropósfera, la capa más baja de la atmósfera, se puede calcular la altura a partir de la presión atmosférica mediante la siguiente fórmula[8]:

$$p = p_0 \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R \cdot L}} \quad (11.1)$$

Donde:

- p_0 : Presión atmosférica estándar a nivel del mar - 101325 Pa.
- L : Gradiente de temperatura¹ - 0.0065 K/m
- T_0 : Temperatura estándar a nivel del mar - 288.15 K
- g : Constante de gravitación terrestre - 9.80665 m/s²
- M : Masa molar del aire seco - 0.0289644 kg/mol
- R : Constante universal de los gases - 8.31447 J/(mol·K)

11.4. Procedimiento

- **Drift y *warm-up*:** En los datos tomados durante varios minutos, se observa lo que parecería ser un drift en las medidas obtenidas del barómetro. Se toman muestras durante 1 hora, con el barómetro quieto, comenzando con el circuito en frío².
- **Caracterización del ruido:** Es de interés caracterizar el ruido en los datos provenientes del barómetro. Si se trata de un proceso estacionario y de ruido blanco entonces se puede reducir el error promediando. Se analizan datos tomados durante diversos períodos de tiempo: 5 minutos, 2 minutos, 20 segundos y 15 segundos. Sobre estos datos se calcula la autocorrelación de las muestras, y se compara el ruido con los valores dados por la hoja de datos. La hoja de datos especifica ruido RMS típico para los distintos modos de funcionamiento.

¹Tasa de incremento de la temperatura con la altura (es negativa).

²Habiendo estado apagado durante, por lo menos, los 30 minutos previos a la prueba.

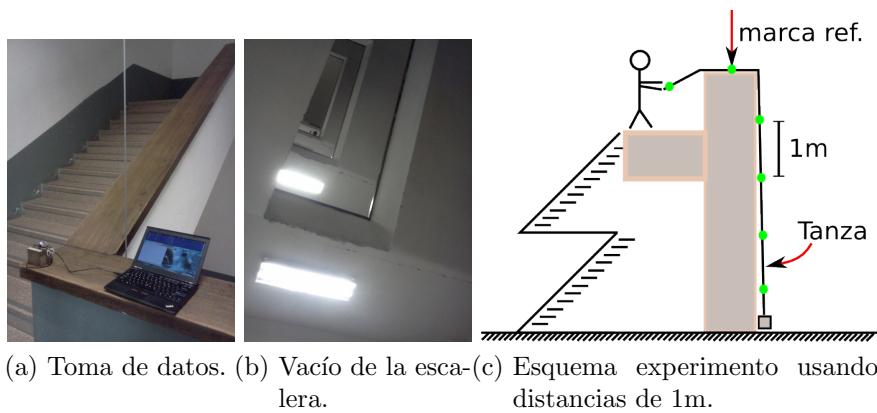


Figura 11.1.

- **Medidas de altura absoluta:** Para analizar la performance del barómetro como instrumento para determinar la elevación absoluta, se toman puntos de altura conocida, y se comparan las lecturas contra los datos conocidos.
- **Distancias de varios metros:** Se despliega una cinta métrica en el vacío del centro de una escalera de 4 tramos. La cinta extendida cubre aproximadamente 25 metros. Se comparan las medidas dadas por el barómetro con las que se obtienen de la cinta.

Se utiliza una cinta de agrimensor, graduada cada 10 centímetros, para medir la distancia de un piso a otro. Luego se toman 1 minuto de datos en cada piso, recorriendo la escalera de un extremo a otro. Se repite este experimento 3 veces (se sube, se baja y se vuelve a subir) y luego se realiza un experimento similar, pero recorriendo la escalera de manera casi continua (Pausa por piso menor a 2 segundos).

- **Distancias de un metro:** Se realizan marcas cada un metro, en seis puntos distintos de una tanza. Se ata el cubo de lapacho en un extremo de la tanza, con la *Mongoose* atornillada a él. Se lo deja descender desde la escalera del *IMERL*³. Se suelta tanza hasta que el cubo se apoya contra el piso, de forma que la primer marca en la cuerda se encuentre junto a quien está realizando las medidas. Se hace una pequeña marca sobre la baranda, en el punto que coincide con la primer marca de la cuerda. Se mide la presión durante un minuto. Luego se sube el cubo hasta que la marca en la baranda coincida con la segunda marca de la cuerda ($\Delta h = 1m$), etc. Al comienzo y al final del experimento se mide la presión a nivel del suelo, para tomar en cuenta cambios meteorológicos.
- **Distancias de decenas de centímetros:** Se ubica el cubo de lapacho siempre en la misma orientación sobre cada uno de los estantes de una estantería, se mide distancia al piso y se toman datos de la presión durante 5 segundos.

11.5. Análisis y resultados

11.5.1. Drift y *warm-up*

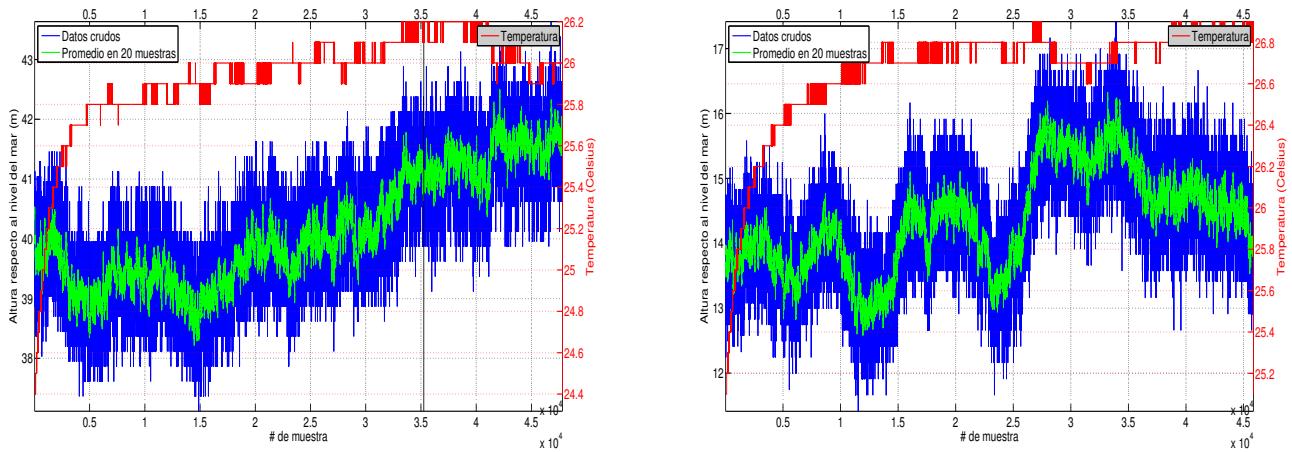
Para observar si hay un drift y/o tiempo de *warm-up* significativo, se procede a tomar muestras durante 1 hora, habiendo estado el dispositivo desenchufado durante

³Instituto de Matemática y Estadística Rafael Laguardia

11. Barómetro

por lo menos media hora (el circuito arranca en frío).

Se repite este experimento varias veces. En las figuras 11.2b y 11.2a se observan los resultados característicos.



(a) Muestras durante una hora, experimento #1.

(b) Muestras durante una hora, experimento #4.

Figura 11.2.

Se observa que existe un tiempo de *warm-up* bastante extenso, pero con un rango pequeño. La temperatura tarda entre 15 y 45 minutos en estabilizarse, y en el proceso varía menos de 2°C. Las muestras del barómetro no parecen estar directamente correlacionadas con la temperatura.

El barómetro no parece ser adecuado para medir la elevación absoluta. En los datos de las figuras 11.2a y 11.2b se observan variaciones de hasta 3 metros en la estimación de la altura en un período de 1 hora. Sin embargo, el error en el corto plazo es significativamente menor que en el largo plazo, parece viable el uso del barómetro como estimador de la altura en el corto plazo. Analizaremos este aspecto en secciones posteriores.

11.5.2. Caracterización del ruido

En los logs de la sección 11.5.1, se observa que:

- El comportamiento del ruido en intervalos extensos no es estacionario.
- La mayoría de los saltos en la medida de la altura son de aproximadamente 25cm. Esto no es coherente con las especificaciones, ya que dan una resolución de 1Pa, que corresponden a una variación en altura de aproximadamente 8cm.

Estacionariedad

Analizando el ruido en intervalos más pequeños, donde el proceso se puede considerar estacionario, el comportamiento del ruido es muy similar al del ruido blanco.

En la figura 11.6 se observa la autocorrelación de las muestras tomadas con el barómetro en intervalos de tiempo de 5 minutos, 2 minutos, 20 segundos y 15 segundos.

Cabe destacar que el ruido se puede considerar estacionario si se usan intervalos de tiempo menores a 15 segundos.

11.5. Análisis y resultados

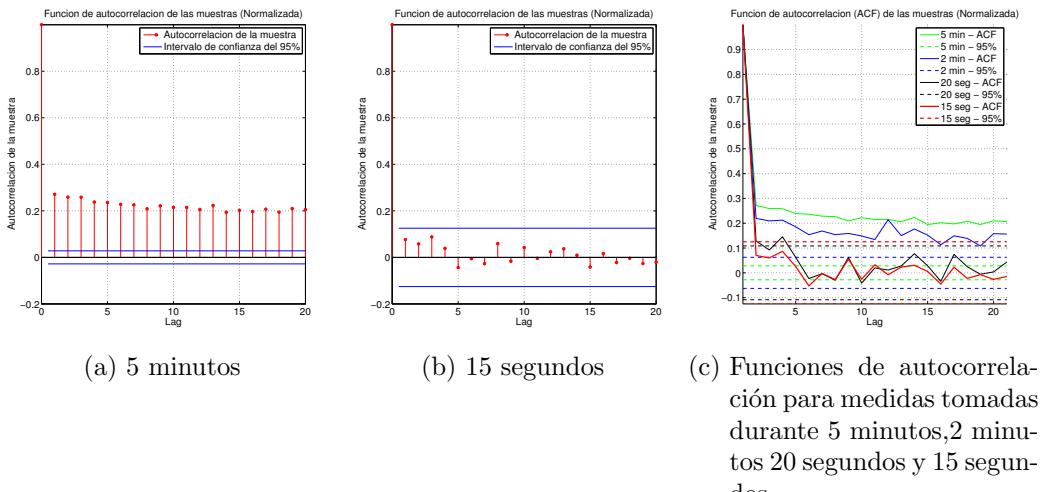


Figura 11.3.: Autocorrelación de las muestras del barómetro.

Modos de funcionamiento

El barómetro puede dar más resolución y menos ruido, a cambio de un incremento en el tiempo entre datos nuevos, y en el consumo de energía. Esto lo logra promediando datos internamente. En la tabla 11.3 se compara el ruido RMS en los distintos modos, con lo que se lograría usándolo en el modo básico (sin promediar internamente) y promediando en el microprocesador.

La resolución más atractiva parece ser la número 2, en la cual se promedian 4 datos.

Modo	Ruido RMS (m)		
	μp	Sensor	specs
0	0.52	0.52	0.50
1	0.37	0.45	0.40
2	0.25	0.37	0.30
3	0.63	0.35	0.25

Cuadro 11.1.

Resolución del instrumento

En la figura 11.5b se observa un histograma de las diferencias entre muestras sucesivas de los datos del barómetro. Se observa que la mayor parte de los datos presentan saltos de 25cm. Se esperaba una discretización en niveles de aproximadamente 8cm, correspondientes a variaciones de presión de 1Pa (la resolución del instrumento). Aunque es difícil de observar, hay muestras en todos los bins.

11.5.3. Altura absoluta

Las figuras 11.2a y 11.2b se corresponden a datos tomados a la misma altura, pero en días distintos. Cabe destacar que la diferencia entre ambas curvas es de aproximadamente 25 metros. Se descarta la posibilidad de utilizar el barómetro para determinar la altura absoluta, ya que estando el barómetro a una altura fija, la elevación varía mucho de un día a otro, lo cual refleja una fuerte dependencia con factores externos, probablemente climáticos, que no interesa considerar.

11.5.4. Distancias de varios metros

En las siguientes figuras se observan los resultados del experimento. Las líneas verticales (en negro) representan el comienzo y el fin de la toma de datos en cada

11. Barómetro

piso. No se hace una medida de la altura absoluta, se miden las distancias entre piso y piso (con la cinta métrica), y se usa como origen de coordenadas la altura dada por el promedio de algunas muestras al comienzo del experimento, ya sea en el piso de más arriba, o en el piso de más abajo (bajadas y subidas respectivamente). La coordenada Y de los segmentos en rojo representan la distancia entre el origen y el piso actual. Dicha distancia se mide con la cinta métrica.

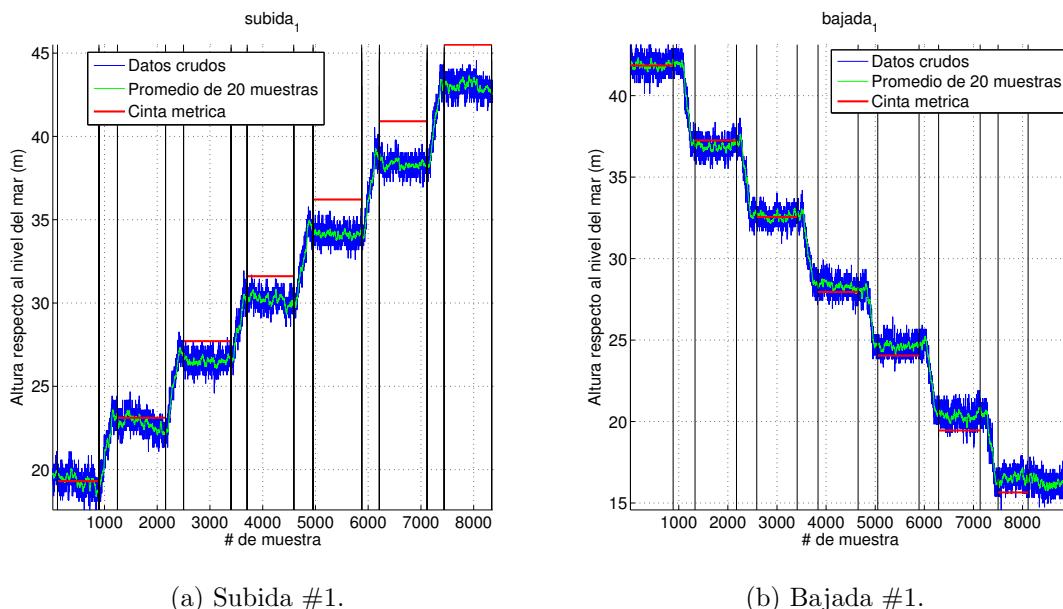


Figura 11.4.: Subida/Bajada #1, 2 minutos de datos por piso.

En la figura 11.4a se observa que para cuando se termina de recorrer la escalera, el error acumulado llega a los 3 metros. En la 11.4b se observan resultados mejores que en la figura 11.4a. Se repite la subida, y se obtienen resultados similares a los de la figura 11.4. Cabe destacar que la performance durante la bajada rápida (figura 11.5a) es muy similar a la que se obtuvo tomando muestras durante 1 minuto.

En la figura 11.6a se observa la evolución del error acumulado para dos subidas⁴, una bajada y una bajada rápida.

El error en la determinación de la distancia entre un piso y el siguiente se observa en la figura 11.6b. Dicho error es relativo a la altura de cada piso, que es de aproximadamente 4.3m. Analizando los números con cuidado, se llega a que el error es siempre menor al 10 %.

De las figuras 11.5a y 11.6a se desprende que utilizando promedios de 20 muestras⁵, se obtiene un error de aproximadamente 0.5m. El comportamiento del barómetro es aceptable, y se corresponde con las especificaciones de la hoja de datos.

11.5.5. Distancias de un metro

Las figuras 11.7a y 11.7b presentan las medidas obtenidas al realizar las medidas estáticas a alturas que difieren en 1 metro. La figura 11.9 muestra las medidas

⁴No se incluye una gráfica de la segunda subida, ya que es similar, en cuanto al error, a la de la primera bajada.

⁵Esta es la cantidad de muestras que se promediaron para determinar la altura cada vez en cada piso en el experimento “rápido”.

obtenidas durante el proceso de subir y bajar el barómetro. En las tres figuras se observan los datos obtenidos y la media móvil considerando 20 muestras.

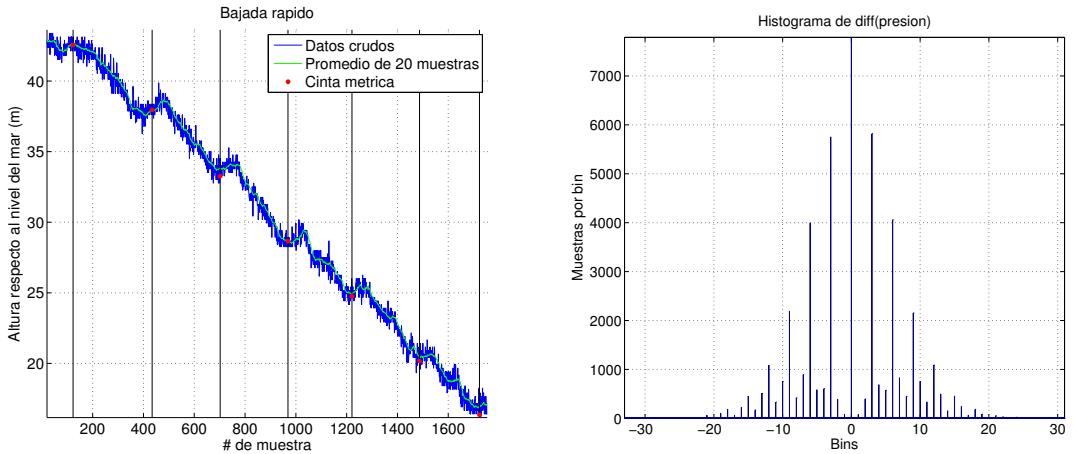


Figura 11.5.

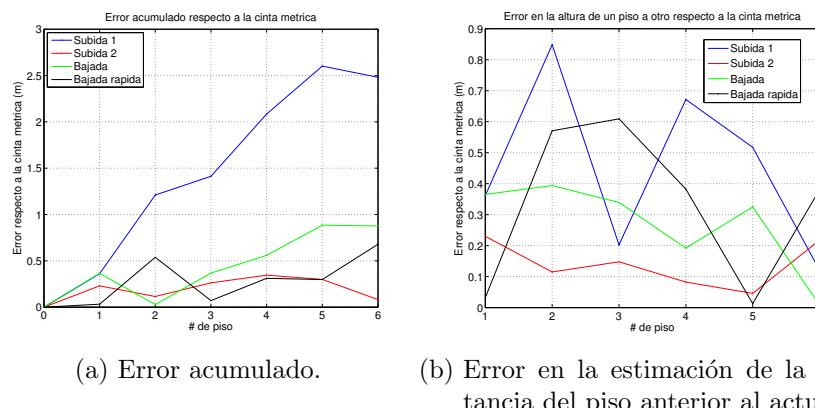


Figura 11.6.: Evolución del error en distancias de varios metros.

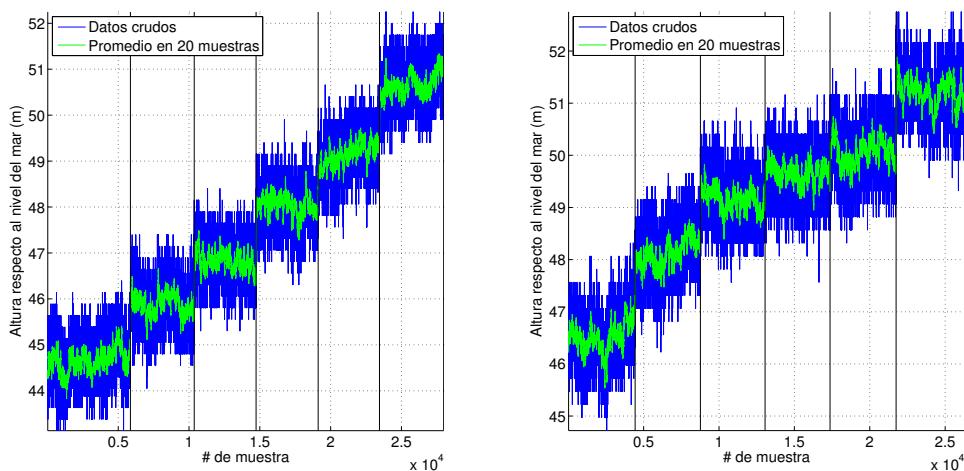


Figura 11.7.: Distancias de 1 metro.

11. Barómetro

En la tabla 11.2 se muestran los valores de altura obtenidos en las distintas posiciones en las dos series de datos. La altura absoluta no es correcta, ya que no se usó la presión real a nivel del mar en la ecuación 11.1.

	Altura medida con el barómetro (m)	
Posición	Serie 1	Serie 2
1	44.66	46.52
2	45.87	48.12
3	46.82	49.15
4	48.04	49.66
5	49.15	50.05
6	50.66	51.21

Cuadro 11.2.

La altura absoluta registrada por el barómetro cambia considerablemente de una serie a la otra. En la figura 11.7b las medidas están, en promedio, un metro más arriba que en la 11.7a. Los puntos que se consideraron fueron los mismos y las medidas fueron realizadas un día de tormenta. La presión atmosférica es muy cambiante en esos días. Eso puede explicar dicha diferencia. Otra explicación puede ser el drift observado en secciones previas.

En la figura 11.8 se muestra la evolución de los errores relativos y acumulado.

Del análisis de cada serie de manera independiente:

■ Serie 1:

- Error promedio: $0,20m$
- Desviación estándar: $0,20m$

■ Serie 2:

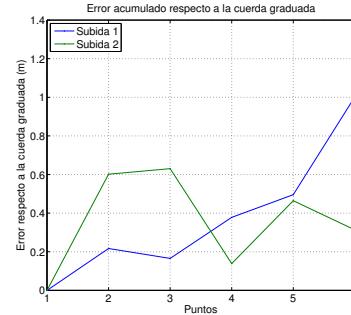
- Error promedio: $-0,07m$
- Desviación estándar: $0,49m$

Si se trabaja con las dos series de datos, se obtienen:

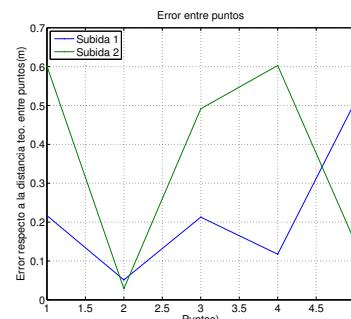
- Error promedio: $0,07m$
- Desviación estándar: $0,38m$

Pese a tener pocas muestras, a partir de los datos anteriores se puede suponer que el 95,5 % de las veces se tendrán errores inferiores a $0,76m$ en diferencias de un metro. El error se mantiene por debajo de $1m$ al usar pocas muestras (4, 8, 10, etc) para cada punto. Este error depende del drift del barómetro, y se puede reducir incorporando, de manera periódica, información externa (GPS) sobre la altura absoluta.

En la figura 11.9 se observa la variación en la altura absoluta. Si bien cada tramo comienza y termina a una altura que cae dentro del error esperado, hay tramos del principio y del final del experimento (barómetro apoyado en el suelo) entre los que se observa una diferencia en altura de $1.5m$. Esto no cae dentro del error esperado. El problema se ve luego de transcurridos varios minutos, tiempo mucho mayor al sugerido en la sección 11.5.2. Puede ser causado por el drift y/o por cambios en las condiciones atmosféricas.



(a) Error acumulado.



(b) Error en la distancia entre puntos.

Figura 11.8.

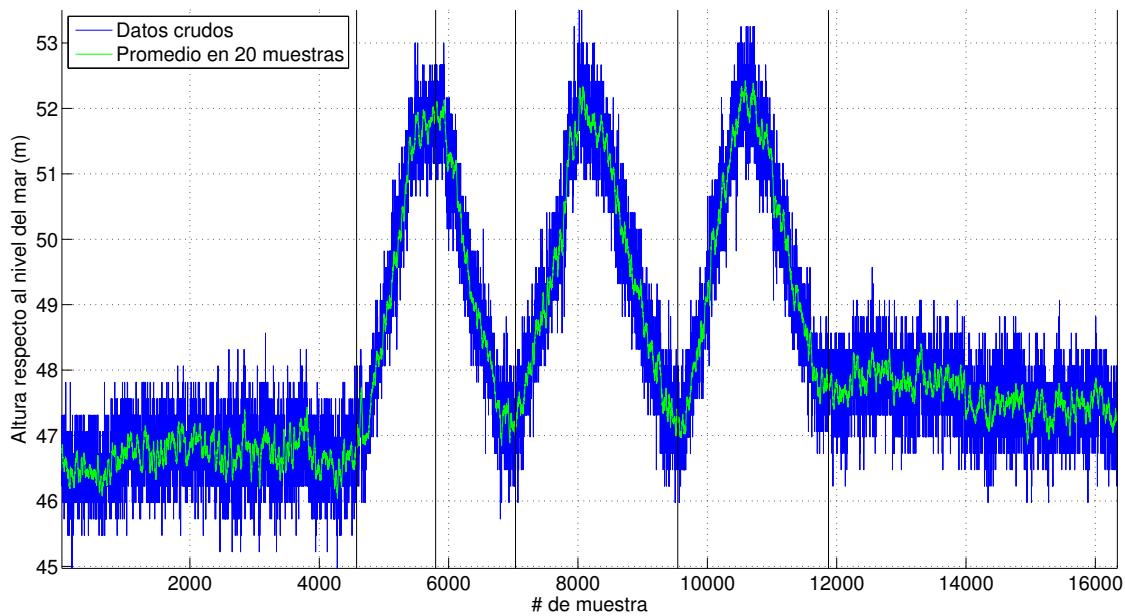


Figura 11.9.: Medidas de altura subiendo y bajando.

11.5.6. Distancias de decenas de centímetros

5 segundos por estante

Se toman datos durante 5 segundos en casa escalón. Repitiendo el análisis realizado en secciones anteriores, se obtiene:

- Error promedio: 2.0 cm
- Desviación estándar: 24.4 cm

3m en aproximadamente 7 segundos

	μ (cm)	σ (cm)
\searrow	15.1	25.9
\nearrow	4.4	58.9
$\searrow + \nearrow$	10.7	49.0

Cuadro 11.3.

Como prueba extra, se recorre 4 veces la estantería sin detenerse cada estante, completando el trayecto de bajar y volver a subir en aproximadamente 7 segundos. Se calcula la distancia recorrida al bajar, la distancia recorrida al subir, y la total, y se compara dichas distancias contra los valores medidos con la cinta métrica. Los resultados se muestran en la tabla 11.3. Solamente se hicieron 4 experimentos, por lo que los datos estadísticos son solamente para dar una idea de la performance frente a movimientos bruscos.

Cabe destacar que el error se mantiene por debajo de los 1.2m en el peor caso. La figura 11.10 muestra gráficamente los resultados del experimento.

11. Barómetro

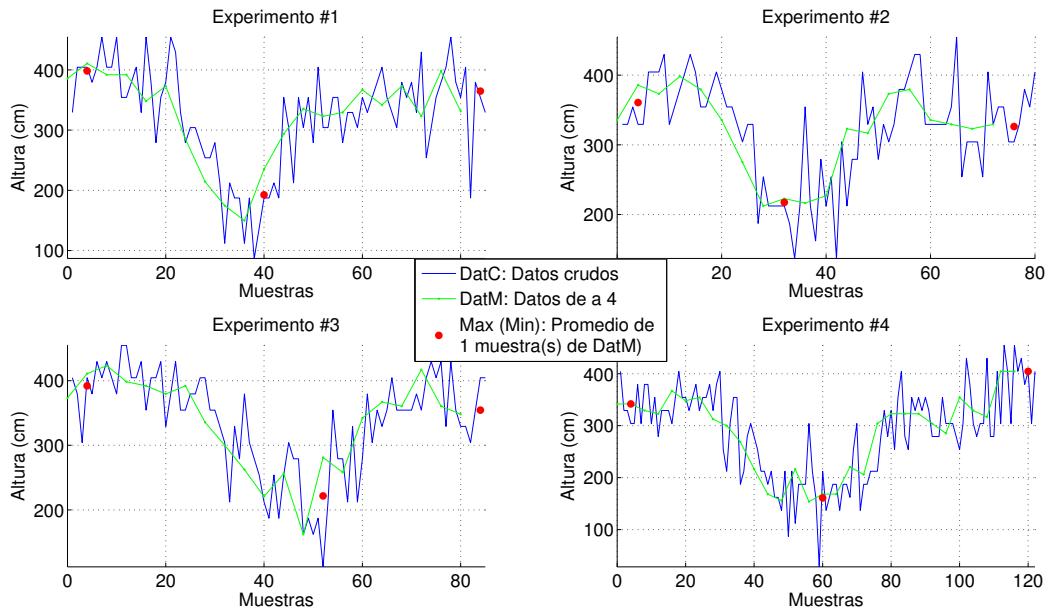


Figura 11.10.: 3m metros en aproximadamente 7 segundos.

11.6. Conclusiones

- **Tiempo de *warm-up*:** El circuito tiene un tiempo de *warm-up* de aproximadamente 45 minutos, durante el cual la temperatura sube aproximadamente 1.5°C. De las especificaciones de los sensores se deduce que una variación de temperatura tan pequeña no debería afectar la performance. De cualquier forma es de interés analizar experimentalmente el efecto de la variación de la temperatura sobre las lecturas del resto de los sensores de la IMU.
- **Drift y Altura absoluta:** El drift y la sensibilidad a factores externos (clima, etc) hacen que el barómetro **no** sea adecuado para determinar la altura absoluta. No es posible obtener una mejor estimación de la altura tomando promedios durante períodos de tiempo mayores, ya que el drift introduciría errores en dicho estimador. Se observa un drift de hasta 0.5m en 1 minuto.
- **Altura relativa:** Estando en reposo, en períodos de tiempo menores a 1 minuto es posible conocer variaciones de altura con un error menor a 0.5m.
- **Rol del barómetro:** Se concluye que el barómetro sirve para determinar variaciones de altura, suavizando la trayectoria, mientras se espera una altura absoluta proveniente del GPS. El barómetro y el GPS se complementan. El barómetro ha de utilizarse con los siguientes parámetros:
 - Error promedio: 0.5m
 - Desviación estándar: 0.25m
 - Modo: 0, promediando 4 muestras en la IMU.
 - Se observa un drift de hasta 0.5m por minuto. Se debe utilizar el GPS para corregirlo.

CAPÍTULO 12

FILTRO DE KALMAN

Se desea implementar un algoritmo de estimación de estados que, corriendo en tiempo real, mantenga una estimación del estado mejor que la que se obtendría utilizando solamente las medidas de los sensores, agregando robustez a las medidas obtenidas, y logrando estimar variables de estado que no se miden directamente de los sensores.

El filtro de Kalman es un algoritmo que usa una serie de medidas ruidosas observadas a lo largo del tiempo y produce una estimación de alguna variable desconocida que resulta ser más precisa que la observación llana de la medida. Opera recursivamente sobre el flujo de la entrada ruidosa y arroja la estimación estadísticamente óptima del estado.

Consiste básicamente en 2 etapas:

- Predicción
- Actualización

En la figura 12.1 se muestra un diagrama de flujo de la operación de un filtro de Kalman. Se parte de una estimación a priori del estado, la cual es utilizada como semilla para las posteriores iteraciones del filtro. En la etapa de **predicción**, el filtro produce una estimación del estado de las variables, junto con sus incertidumbres. Se hallan las variables $P_{k|k-1}$ y $\hat{x}_{k|k-1}$, correspondientes a la predicción de la matriz de covarianza estimada y la predicción del estado estimado, respectivamente. La covarianza, en la teoría de la probabilidad, es una medida de cuán juntas cambian 2 variables aleatorias. Si presentan un comportamiento similar tendrán una covarianza elevada y positiva, si el comportamiento es opuesto la covarianza será negativa y elevada en valor absoluto. Si las variables aleatorias no presentan relación, la covarianza será 0. Es una magnitud que no es sencilla de interpretar pero cobra vital importancia a la hora de entender el comportamiento del filtro y resulta ser la herramienta más clara para regular la influencia de la predicción y de la corrección, como se verá más adelante.

Una vez que llega la siguiente medida de los sensores, necesariamente contaminada con ruido, las estimaciones son actualizadas en la etapa de **actualización**, mediante la utilización de un *promedio ponderado*, dándole mayor peso a las estimaciones

12. Filtro de Kalman

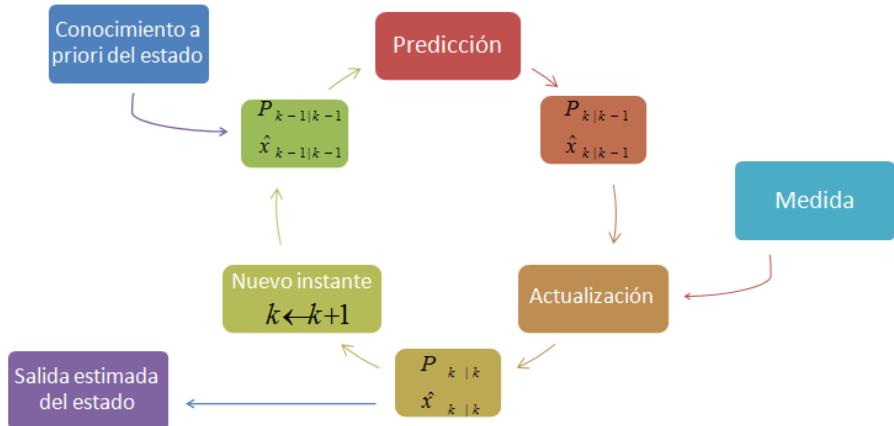


Figura 12.1.: Diagrama de flujo del filtro de Kalman

con menor incertidumbre en cada paso, obteniendo la estimación del estado para el instante k : $P_{k|k}$ y $\hat{x}_{k|k}$.

Por otro lado, las mayores restricciones que impone el filtro de Kalman son que el sistema dinámico subyacente debe ser lineal y que los ruidos presentes deben ser blancos y Gaussianos.

El sistema dinámico que queremos estimar, como se puede ver en el capítulo 3, es claramente **no lineal**, por lo que se debe buscar alguna alternativa al filtro de Kalman clásico. Al revisar la literatura existente, basados en [9] y [10], se decide implementar un filtro de *Kalman extendido* para la estimación del vector de estados.

12.1. Filtro de Kalman Extendido (EKF)

El filtro de *Kalman extendido (EKF)* es la versión no lineal del filtro de Kalman. Linealiza en torno a una estimación de la media y la covarianza. En caso de conocer con exactitud el modelo de transición de estados, el EKF se considera el estándar en la estimación no lineal aplicada a sistemas de navegación.

12.1.1. Modelo matemático

El sistema dinámico sigue el modelo:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

donde \mathbf{w}_k y \mathbf{v}_k son los ruidos de proceso y observación respectivamente. Se asume que son gaussianos y de media nula. Sus matrices de covarianza son \mathbf{Q}_k y \mathbf{R}_k respectivamente.

La función f es utilizada para hallar el estado predicho a partir del estado previo, y de forma análoga la función h se utiliza para hallar la medida predicha a partir del estado. Dicho de otro modo, la función f guarda información sobre la evolución

12.1. Filtro de Kalman Extendido (EKF)

del estado, mientras que la función h representa la transformación entre el vector de estados y la observación ideal (sin ruido). Dada la no linealidad del sistema, las funciones f y h no pueden ser aplicadas directamente a la covarianza. En su lugar se computa su **Jacobiano**, una matriz de derivadas parciales. Es importante destacar que el filtro de Kalman Extendido no tiene propiedades de optimalidad y su precisión dependerá en gran medida de la precisión de la linealización. Dado que se realiza una linealización dinámica, no hay manera de conocer su performance de antemano (por mayor detalles referirse a [11]).

Para la predicción del estado se utilizan las ecuaciones del modelo físico del cuadricóptero presentadas en el capítulo 3.

Las ecuaciones que gobiernan el comportamiento del filtro de Kalman Extendido son:

- Predicción:
 - Estimación de la predicción del estado

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, u_{k-1})$$

- Estimación de la predicción de la covarianza

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$$

- Actualización
 - Residuo de medida

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$$

- Residuo de covarianza

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k$$

- Ganancia de Kalman

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}$$

- Estimación actualizada del estado

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

- Estimación actualizada de la covarianza.

$$\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Las matrices de transición de estados y observación son los siguientes jacobianos:

$$\mathbf{F}_{k-1} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$$

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}$$

12.1.2. Esquema general del estimador de estados

Los datos (en algunos casos redundantes) obtenidos de los diferentes sensores son combinados usando un filtro de Kalman Extendido para determinar el vector de estados, detallado en 3:

$$X = \{x, y, z, \psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}, \omega_{qx}, \omega_{qy}, \omega_{qz}\} \quad (12.1)$$

De la caracterización de los motores, capítulo 6, se obtienen las ecuaciones que rigen la fuerza y el torque de *drag* de cada motor, funciones que dependen solamente de su velocidad de giro. Por lo tanto, basta con controlar la velocidad de giro de los motores para poder determinar todas las entradas al sistema. Se considerará entonces como entrada al sistema las velocidades angulares w_1, w_2, w_3, w_4 de los 4 motores.

El esquema general del integración de los sensores se puede ver en la figura 12.2, el cual se pasa a describir a continuación.

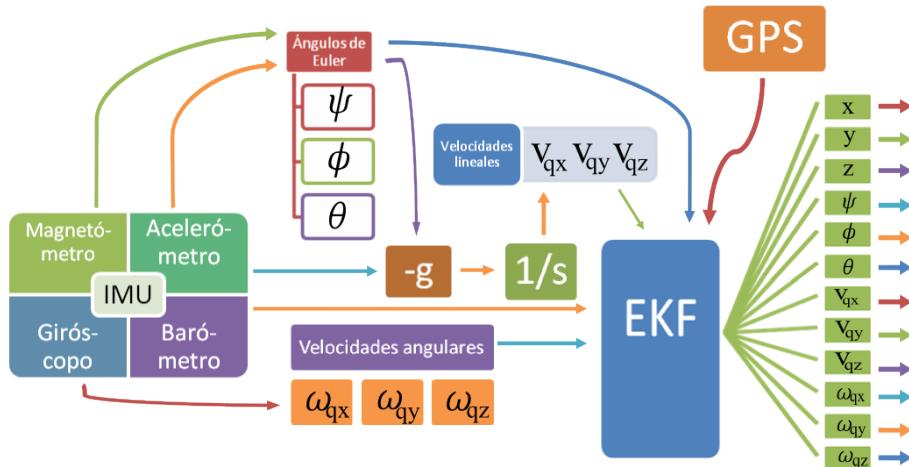


Figura 12.2.: Esquema general de integración de sensores

Como la velocidad con la que llegan las muestras frescas del GPS no es la misma que la velocidad de actualización de las muestras de la IMU, se utilizan 2 filtros de Kalman distintos, dependiendo si hay muestra nueva del GPS. Los 2 filtros de Kalman son en realidad muy similares. Cuando se tiene un dato fresco del GPS ($Kalman_{GPS}$), se utiliza como realimentación para la estimación de la posición (x, y y z), y cuando no se tiene dato nuevo de GPS ($Kalman_{IMU}$), la estimación de la posición se realiza utilizando solamente la etapa de predicción.

$Kalman_{IMU}$ se encarga de estimar los 12 estados del vector X considerando solamente las medidas obtenidas del sensor inercial (IMU): acelerómetros, giróscopos, magnetómetro, barómetro y termómetro. Para la estimación de los ángulos se utiliza una combinación entre las medidas obtenidas del acelerómetro y del magnetómetro, para la estimación de las velocidades angulares se utiliza directamente la medida arrojada por el giróscopo y para las velocidades lineales se trabaja con las medidas arrojadas por el acelerómetro. A su vez, al estar considerado el modelo físico dentro del filtro, se utilizan las estimaciones de todas las variables para determinar las predicciones del resto. Para la estimación de la posición en “x” y en “y” no se tiene

12.1. Filtro de Kalman Extendido (EKF)

ninguna medida de corrección, por lo que quedará determinado únicamente por la predicción del modelo físico, lo cual resulta en una acumulación significativa de error.

$Kalman_{GPS}$ se encarga de la integración de los datos recogidos por el GPS, logrando una corrección en las posiciones y velocidades lineales gracias a la realimentación de la posición entregada por el GPS. En este caso se resuelve el problema de acumulación de error del filtro inercial.

La separación en 2 filtros distintos otorga una interesante flexibilidad. Es posible utilizar una estimación del estado estando a puertas cerradas, sin la necesidad de señal de GPS, a costas de una peor estimación de las posiciones y velocidades lineales en “x” e “y”. Además, como ya se mencionó, la frecuencia de muestreo del GPS es mucho más baja que la del resto de los sensores, por lo cual se hace imprescindible la utilización de un filtro meramente inercial. Los datos de la IMU llegan cada $10ms$, mientras que los del GPS cada $1s$, entonces mientras no se tiene un dato nuevo de GPS, se utiliza la estimación inercial de las posiciones y cuando llega un dato nuevo de GPS es utilizado como corrección, dándole mucho más peso que a la estimación inercial, para evitar el *drift* mencionado.

Orientación

En estado estático, los ángulos de *pitch* y *roll* pueden ser determinados proyectando el vector medido de aceleración gravitacional por el acelerómetro. A su vez, los cambios entre el sensor magnético y el vector geomagnético medido, representan al ángulo *yaw*. Por lo tanto, el conjunto del acelerómetro y magnetómetro es capaz de determinar la orientación del cuadricóptero mediante las medidas del vector de aceleración gravitatoria y el vector geomagnético. Dado que el magnetómetro digital no introduce acumulación de error, este sistema ha sido muy utilizado para la determinación estática de la posición, como se puede ver en [12]. Una observación no menor es que en el pasaje de estado estático al vuelo, lo antes dicho sobre los ángulos *pitch* y *roll* sigue siendo válido ya que las aceleraciones que puedan aparecer en vuelo resultan despreciables frente a la aceleración de la gravedad.

Aceleración

Para poder obtener la transformación entre el vector de estados y la observación es necesario entender qué aceleración medirá el acelerómetro. Como se explica en 7, dicho dispositivo referencia su medida a un sistema en caída libre, por lo que por ejemplo al estar quieto se deberá medir la aceleración gravitatoria (g).

Como se puede ver en el esquema mostrado en la figura 12.2, el acelerómetro es utilizado para estimar las variables de estado v_{qx} , v_{qy} y v_{qz} , correspondientes a las velocidades lineales medidas en el sistema del cuadricóptero. No se debe confundir la derivada de estas variables con la aceleración medida por el acelerómetro. Estas corresponden a la velocidad expresada en el sistema del cuadricóptero, por lo tanto para obtener la aceleración se utiliza la fórmula de la derivada de un vector en un sistema móvil. Esta derivada incluye términos que dependen de la velocidad angular del sistema. Por otro lado se debe tener en cuenta que el sistema de referencia del acelerómetro es un sistema en caída libre, por ello es que se incluye el bloque “- g ” en el diagrama de la figura 12.2.

12.2. Resultados: Kalman inercial

12.2.1. Simulación de vuelo

Hovering

La trayectoria de *Hovering* consiste en que el cuadricóptero permanezca suspendido en el aire en el lugar. Para simular esta situación se setea como entrada una velocidad angular en los motores calculada para equilibrar la fuerza del peso de cuadricóptero. Se presentan a continuación algunos resultados de la implementación del filtro de Kalman como estimador del vector de estados.

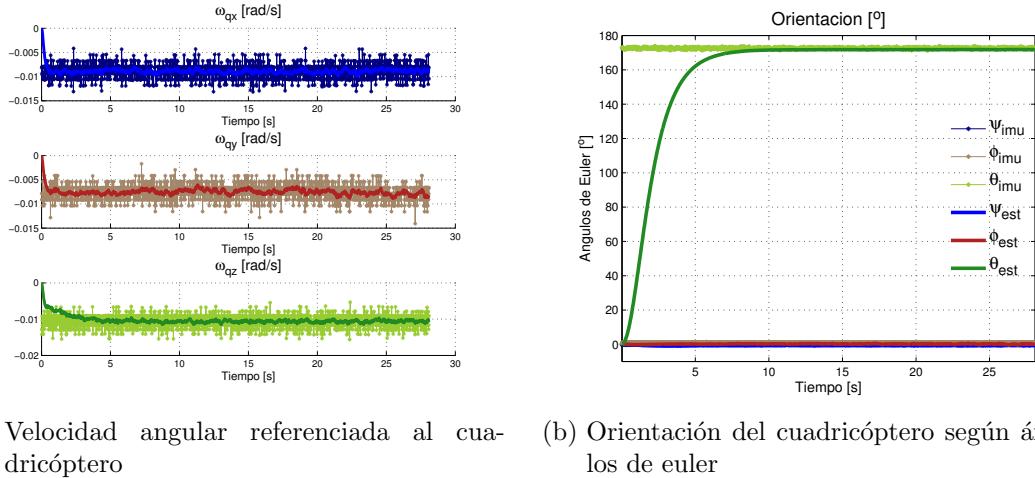


Figura 12.3.

En la figura 12.3a se muestra la estimación de las velocidades angulares, junto con las medidas del giróscopo. Se puede ver claramente que la estimación del estado sigue a las medidas del giróscopo, suavizándolas y logrando robustez frente a ruidos inherentes al sensor.

Por otro lado en la figura 12.3b se muestra tanto la orientación estimada por el filtro de Kalman, como la deducida directamente a partir de las medidas de los acelerómetros y magnetómetros de la IMU. Nuevamente se puede apreciar el buen comportamiento del filtro, suavizando notoriamente las medidas otorgadas por los sensores. Además resulta interesante analizar el transitorio de la estimación del estado, es decir, el tiempo que el filtro tarda en estimar al estado adecuadamente. En este caso el estimador tarda aproximadamente 7 segundos en pasar de $\theta = 0$ hasta $\theta = 170$, lo cual sin duda es un tiempo totalmente inaceptable en vuelo. Este es uno de los casos más extremos ya que el ángulo θ es cercano a 180 grados. De todas maneras es un problema fácil de solucionar, se puede tomar algunas muestras de la IMU y usarlas como semilla para el filtro, y/o agregar una etapa de *calentamiento* del filtro antes de que el cuadricóptero levante vuelo. A su vez, ajustando las matrices de covarianza \mathbf{Q} y \mathbf{R} , presentadas anteriormente, se puede lograr muy fácilmente que la estimación se acerque mucho más rápido a la medida del sensor.

Del resto de las variables de estado, x , y , z , v_{qx} , v_{qy} y v_{qz} , la única que tiene realimentación, considerando una estimación de estado puramente inercial, es z . El

resto quedan determinadas únicamente por el modelo físico, en el cual se basa la etapa de predicción del filtro de Kalman. Dicho de otro modo, la etapa de actualización/corrección del filtro no surte efecto sobre dichas variables de estado.

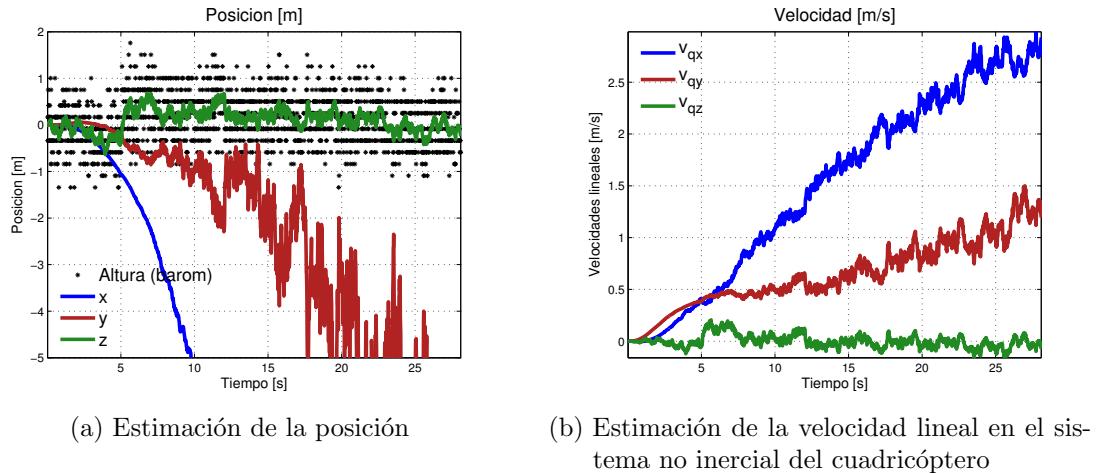


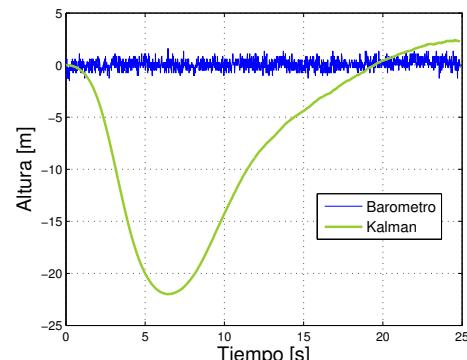
Figura 12.4.

Como se puede ver en la figura 12.4, la estimación de los estados que no tienen realimentación directa resulta claramente peor que el resto. De hecho, lo que ocurre por ejemplo en las velocidades lineales según x e y , es que la observación (dato obtenido del sensor) es la derivada de la variable de estado, y por ello es que se observa el carácter creciente (en valor absoluto) en la estimación del estado, causado por la integración en el tiempo del valor de aceleración otorgado por la IMU. La posición según z , en cambio, tiene realimentación directa del barómetro. En la figura 12.4a se muestra en negro los datos obtenidos de dicho sensor, y sobre ellos la estimación de la altura z .

Situación imposible

Se simula una situación donde todos los motores se setean a una velocidad w_{hover} , como en el caso anterior, pero en esta oportunidad el cuadricóptero se dispone con un ángulo de $\varphi = 45^\circ$. Claramente es una situación de vuelo imposible ya que con esa velocidad en los motores el cuadricóptero no podría mantener el reposo y perdería altura. Se puede analizar la estimación de la orientación obtenida por el filtro en la figura 12.5.

Lo que sucede en esta situación es que la predicción de altura dice que el cuadricóptero descenderá, pero el barómetro indica altura constante. El desempeño del filtro en este caso estará determinado por la relación entre el peso que se le da a la observación y el que se le da a la predicción. En este caso el peso de la predicción


 Figura 12.5.: Altura - $\phi = 45^\circ$

12. Filtro de Kalman

es lo suficientemente grande como para que el filtro estime un estado distinto al que otorgan los sensores.

12.2.2. Vuelo real

Si bien las simulaciones resultan importantes para analizar el comportamiento del filtro en diferentes situaciones, tienen limitaciones importantes al intentar simular situaciones de vuelo reales, ya que las predicciones no serán correctas debido a la influencia de fuerzas externas no consideradas en el modelo físico (y por lo tanto, tampoco en la predicción), como estar apoyado sobre una mesa, etc. Por este motivo, es de vital importancia analizar el comportamiento del filtro sobre una situación real de vuelo.

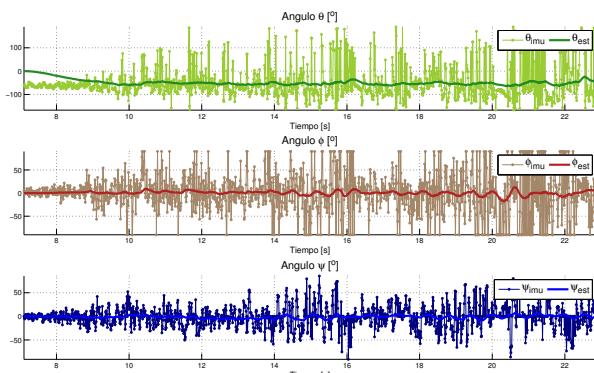
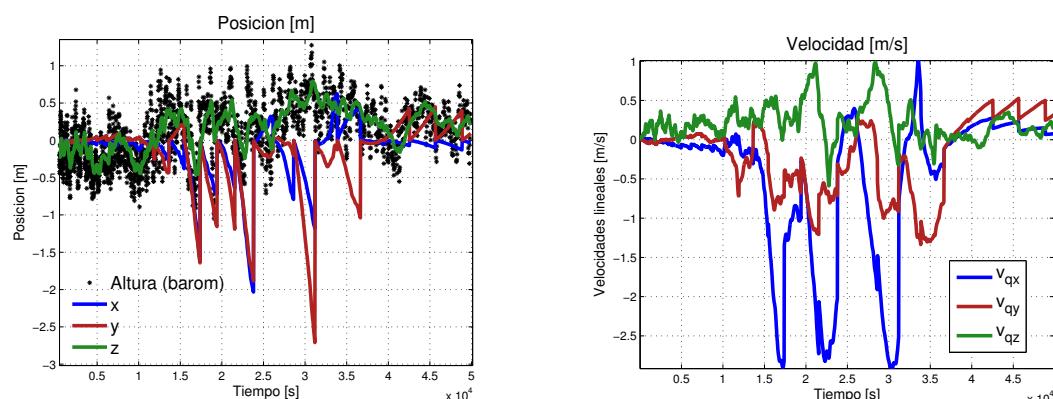


Figura 12.6.: Estimación en vuelo real

En la figura 12.6 se muestra los resultados de la estimación de la orientación del cuadricóptero. Se observa claramente un aumento del ruido de las medidas respecto a los casos simulados, llegando a picos de hasta 50 grados. De todas maneras se puede corroborar el buen funcionamiento del filtro.

12.3. Resultados Kalman inercial + GPS

Se realiza una simulación de trayectoria de *Hovering*, introduciendo el filtro EKF que integra los datos del GPS. Éstos son simulados como si el cuadricóptero no se moviera (que es lo que efectivamente ocurre), de modo que se entrega el valor 0 a todas las posiciones



(a) GPS - Estimación de la posición

(b) GPS - Estimación de la velocidad lineal en el sistema no inercial del cuadricóptero

Figura 12.7.: Posición y velocidad al introducir datos de GPS

12.3. Resultados Kalman inercial + GPS

Como se puede ver en la figura 12.7, se actualiza la posición en “x” e “y” cada 1 segundo gracias al GPS, corrigiendo el *drift* observado en la figura 12.4a. A su vez, las velocidades lineales se acotan notoriamente gracias a la influencia de la realimentación de la posición en el filtro de Kalman.

Se obtienen resultados muy satisfactorios para la estimación de estados, logrando suavizar el ruido presente en los sensores, que es mayormente mecánico (y magnético en el magnetómetro), sin retrasar la señal. A su vez la información de todos los sensores es combinada para mejorar la estimación de cada una de las variables de estado, logrando resultados ampliamente mejores que lo que se lograrían utilizando la información de cada sensor por separado.

Parte V.

Desarrollo del controlador

CAPÍTULO 13

LINEALIZACIÓN Y PUNTOS DE OPERACIÓN

Independientemente del sistema bajo estudio, a la hora de elegir la técnica de control a utilizar se plantean diversas posibilidades. Parece razonable intentar resolver el problema planteado utilizando las técnicas más sencillas de las que se disponen, al menos en una primera aproximación. En caso de que dicha solución no fuese satisfactoria se puede optar por una técnica con un mayor grado de complejidad.

Las técnicas de control más sencillas y con las que se tiene mayor experiencia se basan en el estudio de sistemas lineales invariantes en el tiempo (SLIT). El MVE obtenido en el capítulo 3 es no lineal, por lo que se propone resolver el problema del control del cuadricóptero aproximando el sistema por un sistema lineal invariante en el tiempo. En una primera instancia se centrará el análisis en determinar bajo qué condiciones es posible aproximar el sistema por un SLIT.

13.1. Concepto general

Dado un sistema que se rige por la siguiente evolución de su vector de estados:

$$\dot{x}(t) = f(x(t), u(t)) \quad (13.1)$$

Donde $x(t)$ es el vector de estados del sistema y $u(t)$ el vector de las entradas. Suponiendo, en una primera instancia, que tanto $x(t)$ como $u(t)$ son de dimensión uno y considerando además el punto de operación definido por $x^*(t)$ y $u^*(t)$, un desarrollo de Taylor de orden uno en torno al punto de operación resulta en:

$$x(t) = f(x(t), u(t)) = f(x^*(t), u^*(t)) + \frac{\partial f}{\partial x} \Big|_{u=u^*}^{x=x^*} (x(t) - x^*(t)) + \frac{\partial f}{\partial u} \Big|_{u=u^*}^{x=x^*} (u(t) - u^*) \quad (13.2)$$

Definiendo $\tilde{x}(t) = x(t) - x^*(t)$ y $\tilde{u}(t) = u(t) - u^*(t)$ se obtiene que:

$$\dot{\tilde{x}}(t) = \frac{\partial f}{\partial x} \Big|_{u=u^*}^{x=x^*} \tilde{x}(t) + \frac{\partial f}{\partial u} \Big|_{u=u^*}^{x=x^*} \tilde{u}(t) \quad (13.3)$$

13. Linealización y puntos de operación

Este mismo concepto puede generalizarse al caso en el cual el vector de estados y el vector de entradas tienen dimension n y m respectivamente.

$$\dot{\tilde{X}}(t) = A(t)\tilde{X}(t) + B(t)\tilde{U}(t) \quad (13.4)$$

Donde $A(t)$ es una matriz de $n \times n$ y $B(t)$ es de $n \times m$, tales que $a_{ij} = \frac{\partial f_i}{\partial X_j}|_{U=U^*}^{X=X^*}$ y $b_{ij} = \frac{\partial f_i}{\partial U_j}|_{U=U^*}^{X=X^*}$.

En el caso en que todos los coeficientes de las matrices A y B son constantes es posible afirmar que el sistema es lineal e invariante en el tiempo.

13.2. Puntos de operación

Se buscan trayectorias tales que la linealización del sistema en torno a ellas resulte en un sistema lineal invariante en el tiempo.

Para lograr este cometido, se debe cumplir que todos los elementos de las matrices D.2 y D.3 del anexo D sean constantes. Dichas matrices corresponden a la linealización del MVE en torno a una trayectoria genérica. Se desprende del análisis de dichas matrices que los puntos de operación que cumplen esta condición quedan restrictos a un subconjunto tal que:

$$\dot{\psi} = \dot{\phi} = \dot{\theta} = \dot{v}_{qx} = \dot{v}_{qy} = \dot{v}_{qz} = \dot{\omega}_{qx} = \dot{\omega}_{qy} = \dot{\omega}_{qz} = \dot{\omega}_1 = \dot{\omega}_2 = \dot{\omega}_3 = \dot{\omega}_4 = 0 \quad (13.5)$$

Luego se verá que, realizando un cambio de variable como se explica en [13], se puede ampliar el conjunto de trayectorias posibles, pudiendo agregar aquellas para las cuales $\dot{\theta} \neq 0$. Estas consideraciones llevan a concluir que el conjunto de trayectorias permitidas¹ es aquel en el cual tanto la velocidad del centro de masa, como la velocidad angular son constantes en el tiempo. Las trayectorias que cumplen con estas condiciones son tres:

- Hovering
- Vuelo en linea recta a velocidad constante
- Vuelo en círculo a velocidad constante

En los tres casos se tienen que cumplir las restricciones siguientes:

$$\begin{pmatrix} \dot{v}_{qx} \\ \dot{v}_{qy} \\ \dot{v}_{qz} \end{pmatrix} = 0 \quad \begin{pmatrix} \dot{\omega}_{qx} \\ \dot{\omega}_{qy} \\ \dot{\omega}_{qz} \end{pmatrix} = 0 \quad (13.6)$$

Cada una de las trayectorias posibles tiene además sus particularidades, a continuación se determinan las restricciones específicas de cada una de ellas.

¹por la restricción de trabajar con un sistema lineal invariante en el tiempo.

13.2.1. Hovering

En el caso del reposo mecánico no solo deben cumplirse las condiciones establecidas en 13.6 sino que además las velocidades del centro de masa y las velocidades angulares deben ser iguales a cero:

$$\begin{pmatrix} v_{qx} \\ v_{qy} \\ v_{qz} \end{pmatrix} = 0 \quad \begin{pmatrix} \omega_{qx} \\ \omega_{qy} \\ \omega_{qz} \end{pmatrix} = 0 \quad (13.7)$$

Dado que la velocidad y velocidad angular del sistema fueron definidas como cero, solo quedan por determinar 7 variables: los ángulos de Euler y las velocidades angulares de los motores.

Al imponer las condiciones de 13.7, las ecuaciones 3.18 y 3.21 del MVE se cumplen trivialmente. La restricción establecida en 13.6 consta de seis ecuaciones. Estas no dependen de θ , obteniendo entonces un sistema de seis ecuaciones y seis incógnitas que debe ser resuelto a fin de determinar las condiciones que permiten el hovering. Ninguna de las seis ecuaciones depende la posición lo cual, sumado a la independencia de la condición de hovering respecto de θ , permite confirmar un resultado evidente: **el hovering puede lograrse en cualquier punto del espacio con cualquier ángulo de Yaw**

Del análisis de las ecuaciones 13.6 se obtiene que:

$$\varphi = 0 \quad \psi = 0 \quad \omega_1 = \omega_2 = \omega_3 = \omega_4 = \omega_{hover} \quad (13.8)$$

13.2.2. Vuelo en linea recta a velocidad constante

Para lograr el vuelo en linea recta a velocidad constante se tienen que cumplir las condiciones de 13.6, al igual que en el caso anterior. La particularidad del vuelo en linea recta es que la velocidad del centro de masa es una constante distinta de cero.

$$\begin{pmatrix} v_{qx} \\ v_{qy} \\ v_{qz} \end{pmatrix} = cte \quad \begin{pmatrix} \omega_{qx} \\ \omega_{qy} \\ \omega_{qz} \end{pmatrix} = 0 \quad (13.9)$$

Al imponer que $\vec{\omega}_q = 0$ la ecuación 3.18 se verifica trivialmente y las ecuaciones 3.25 y 3.36 quedan idénticas al caso del sistema en reposo. Por lo tanto, al igual que para el hovering, se tendrá que:

$$\varphi = 0 \quad ; \quad \psi = 0 \quad ; \quad \omega_1 = \omega_2 = \omega_3 = \omega_4 = \omega_{hover} \quad (13.10)$$

Hasta aquí no se ha considerado la ecuación que relaciona las velocidades lineales en el sistema inercial con las velocidades lineales expresadas en el sistema del cuadricóptero. Teniendo presente el resultado obtenido sobre φ y ψ dichas ecuaciones toman la forma:

$$\begin{aligned} \dot{x} &= v_{qx} \cos \theta - v_{qy} \sin \theta \\ \dot{y} &= v_{qy} \sin \theta + v_{qx} \cos \theta \\ \dot{z} &= v_{qz} \end{aligned} \quad (13.11)$$

Se deduce entonces que el vuelo en línea recta queda determinado al fijar las restantes 4 variables del sistema: v_{qx} , v_{qy} , v_{qz} y θ . Estos parámetros serán determinados por el generador de trayectorias con el objetivo de realizar una recta en particular con una orientación determinada.

13.2.3. Vuelo en círculos

Con las ecuaciones desarrolladas hasta ahora no resulta posible realizar movimientos con velocidades angulares distintas de cero sin que el sistema resultante sea variante en el tiempo. Si se impone que la velocidad angular del sistema sea distinta a cero se tiene que al menos una de las derivadas de los ángulos de Euler es distinta de cero y por lo tanto se tendrá al menos un ángulo de Euler variante en el tiempo. Para poder obtener un movimiento circular se realiza el cambio de variable en el MVE en lo que respecta a las ecuaciones de la derivada de la posición propuesto en [13].

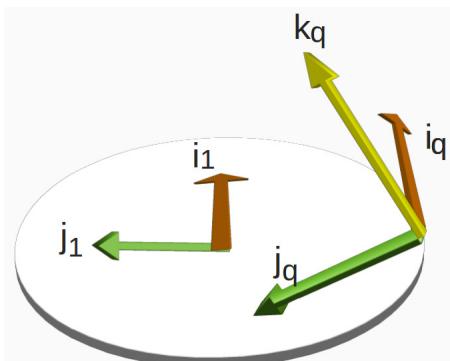


Figura 13.1.: Vuelo en círculo

El cambio de variable en cuestión consiste en expresar la posición (exclusivamente para los movimientos circulares) en la base del cuadricóptero. Suponiendo que el cuadricóptero se encuentra realizando un movimiento tal que su proyección sobre el plano horizontal ($z = 0$) es circular, se puede describir la posición en todo momento tomando como origen el centro de dicho círculo. Como se observa en la figura 13.1 la posición en dicho plano se puede expresar como $-R\vec{j}_1^2$. Donde R , es el radio del círculo y \vec{j}_1 es un vector de la base que se obtiene al realizar la primera rotación de Euler definida en 3.12. Multiplicando el vector $-R\vec{j}_1$ por la matriz cambio de base $H_2^q H_1^2$ definida en 3.12 se obtiene la posición expresada en el sistema S_q . Como \vec{j}_1 es invariante frente a la rotación H_1^2 , alcanza con multiplicar dicho vector por H_2^q obteniendo así:

$$\vec{r}_q = -H_2^q \vec{j}_1 = -R \begin{pmatrix} 0 \\ \cos \psi \\ -\sin \psi \end{pmatrix} \quad (13.12)$$

Al expresar la posición con este cambio de variables se obtiene la independencia de la misma respecto de dos de los tres ángulos de Euler: φ y θ . En lo que respecta al ángulo φ la independencia de esta variable no aporta absolutamente nada ya que para que el movimiento circular a velocidad constante sea posible dicho ángulo debe ser cero. La razón es que si el ángulo fuese diferente a cero se tendría una componente de la fuerza en la dirección tangencial al círculo, y por lo tanto una aceleración en dicha dirección.

La independencia respecto de θ en la posición es un paso fundamental para lograr el vuelo en círculos ya que las únicas ecuaciones del MVE (3.53) que dependen de esta variable son las que refieren a la derivada de la posición. Por lo tanto, será posible desarrollar un nuevo MVE independiente del ángulo de Yaw.

²Esto es debido a que se elige que el vector \vec{i}_q sea tangente a los círculos descriptos. El razonamiento es análogo si se desea que el vector \vec{j}_q sea tangente al círculo, en este caso debe expresarse la posición del sistema como $R\vec{i}_1$

Interesa ahora derivar la posición para terminar de completar el modelo del sistema. Dado que la posición se encuentra expresada en el sistema del cuadricóptero, se tiene que:

$$\vec{v}_q = \frac{d\vec{r}_q}{dt} = \frac{d'\vec{r}_q}{dt} + \vec{\omega}_q \times \vec{r}_q \quad (13.13)$$

y por lo tanto:

$$\begin{pmatrix} \dot{x}_q \\ \dot{y}_q \\ \dot{z}_q \end{pmatrix} = \begin{pmatrix} v_{q_x} - \omega_{q_y} z_q + \omega_{q_z} y_q \\ v_{q_y} - \omega_{q_z} x_q + \omega_{q_x} z_q \\ v_{q_z} - \omega_{q_x} y_q + \omega_{q_y} x_q \end{pmatrix} \quad (13.14)$$

Este modelo permite realizar el control basado en técnicas de control lineal para trayectorias circulares en planos tales que $z = cte$.

Es fundamental aclarar que las ecuaciones derivadas hasta aquí serán utilizadas exclusivamente para la linealización del sistema y no para el control. Para esto último se continuará utilizando el modelo de 3.53. El movimiento circular queda determinado exclusivamente por dos parámetros: la velocidad (V_I) y la velocidad angular ($\dot{\theta}$), estos serán considerados conocidos ya que es trabajo del generador de rutas o del usuario determinar que movimiento se desea realizar. Se desea determinar once parámetros: siete de las doce variables de estado (No interesa fijar ni ángulo de Yaw ni la posición y ya fue determinado que $\varphi = 0$) y las cuatro velocidades angulares de los motores. Se deben tener en cuenta las restricciones establecidas en 13.6 sobre las derivadas de las velocidades del centro de masa y de las velocidades angulares. En lo que respecta a las derivadas de los ángulos de Euler se debe cumplir que $\dot{\varphi} = \dot{\psi} = 0$, además $\dot{\theta}$ debe ser igual al valor impuesto. De $\dot{\psi} = 0$ se deduce trivialmente que $\omega_{q_x} = 0$

Para cada instante se tendrá un ángulo de Yaw distinto, por lo tanto la velocidad horizontal a cada instante tendrá una dirección distinta, y su módulo será constante. Sin pérdida de generalidad podemos considerar la situación evaluando $\theta = 0$

$$\begin{pmatrix} v_{q_x} + v_{q_y} \sin \psi \\ v_{q_y} \cos \psi - v_{q_z} \sin \psi \\ v_{q_y} \sin \psi + v_{q_z} \cos \psi \end{pmatrix} = \begin{pmatrix} V_I \\ 0 \\ 0 \end{pmatrix} \quad (13.15)$$

De la imposición de dicha condición surge:

$$v_{q_y} = -\tan^2 \psi \quad v_{q_y} \quad (13.16)$$

La solución de dicha ecuación es: $v_{q_y} = 0$, y por lo tanto $v_{q_z} = 0$ y $v_{q_x} = V_I$.

La solución trivial $\psi = 0$ no es de interés para el vuelo en círculos ya que, si se impone dicha condición es imposible producir una fuerza en algún sentido distinto que según la vertical, no pudiéndose realizar una fuerza centrípeta necesaria para lograr un movimiento circular.

Resta por determinar tres de las variables de estado (ψ , ω_{q_y} y ω_{q_z}) y las velocidades angulares de los motores, para lo cual se utilizarán las siete ecuaciones restantes del modelo físico. El sistema no será resuelto en forma analítica, sino en forma numérica a la hora de planificar la trayectoria.

13. Linealización y puntos de operación

Hasta aquí se han definido tres tipos de trayectorias, desde las cuales es posible definir y entender el comportamiento del cuadricóptero desde el “mundo” del control lineal, lo que significa un paso fundamental en camino de lograr controlar el cuadricóptero. Las trayectorias definidas ofrecen una amplia gama de posibilidades, ya que concatenando las mismas se pueden obtener prácticamente cualquier movimiento, a excepción de maniobras que involucren variaciones en más de un ángulo de Euler a la vez. El subconjunto de trayectorias con el que se puede trabajar es considerado ampliamente satisfactorio y por ende se da por concluído el análisis sobre los puntos de operaciones y la linealización del sistema.

CAPÍTULO 14

DISEÑO DEL CONTROLADOR

Dentro del “mundo” del control lineal, existen diversas técnicas que permiten alcanzar el objetivo trazado de lograr que el cuadricóptero siga alguna de las trayectorias específicas en el capítulo 13. Algunas de las técnicas más utilizadas se centran en el diseño de controladores PID¹ o LQR². Ambas técnicas presentan ventajas y desventajas. En el trabajo realizado en [3] se propone el control de un cuadricóptero utilizando un controlador PID. La gran mayoría de controladores en aplicaciones industriales son de este tipo, la principal ventaja que presentan es que se trata de un diseño que tiene una estructura simple y es adecuado para la gran mayoría de procesadores ya que el costo computacional del mismo es prácticamente nulo. El concepto principal es que la señal de entrada a la planta es una función del error entre el estado deseado y el estado estimado.

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (14.1)$$

donde $e(t) = X_d - \hat{X}$, siendo X_d el vector de estados deseado y \hat{X} el estado estimado.

En [3] se limitan las trayectorias a trayectorias de hovering. En ese supuesto, se realizan algunas aproximaciones que permiten reducir el sistema físico a las siguientes ecuaciones

$$\begin{pmatrix} \ddot{z} \\ \ddot{\psi} \\ \ddot{\varphi} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} g + (\cos \varphi \cos \psi) \frac{U_1}{M} \\ \frac{U_2}{I_{xx}} \\ \frac{U_3}{I_{yy}} \\ \frac{U_4}{I_{zz}} \end{pmatrix} \quad (14.2)$$

Donde U_1, U_2, U_3 y U_4 son combinaciones lineales de los cuadrados de las velocidades angulares de cada motor. En este caso se puede tratar cada variable por separado, ya que cada variable es afectada por una sola entrada, siendo relativamente sencillo determinar las constantes K_p, K_I y K_d en función de donde se desea ubicar los polos del sistema realimentado. Con esta estrategia se pierde la posibilidad de controlar las otras 8 variables de estado, limitándose entonces a un cuadricóptero

¹Proporcional, integral y derivativo

²Linear quadratic regulator

14. Diseño del controlador

que puede realizar exclusivamente movimientos en la dirección vertical y giros en torno a su eje vertical.

Si se intenta controlar el sistema de interés en este trabajo utilizando esta técnica, surge el problema de determinar al menos una matriz de realimentación (si se trabaja solamente con un controlador proporcional) de 12×4 , es decir que se deben determinar los 48 elementos de la matriz de forma de lograr que la respuesta del sistema sea la deseada. Esta tarea no resulta sencilla, ya que es extremadamente difícil comprender exactamente la influencia de cada parámetro de la matriz de realimentación en la ubicación de los polos en el sistema realimentado incluso para asegurar algo elemental y necesario como la estabilidad del sistema.

Por dicho motivo se opta por explorar el camino propuesto por otros trabajos como [14], donde la técnica elegida para realizar el control del cuadricóptero es LQR.

14.1. Conceptos generales sobre LQR

Sea el sistema realimentado de la figura 14.1, con $X(t) \in \mathcal{M}_{n \times 1}$ el vector de estados del sistema y $U(t) \in \mathcal{M}_{m \times 1}$ las entradas. r es el setpoint a partir del cual, multiplicando por dos matrices adecuadas pueden obtenerse los valores deseados de entrada U^* y de las variables de estado X^* . El objetivo planteado es el de obtener una matriz de realimentación K para el sistema utilizando la técnica de LQR.

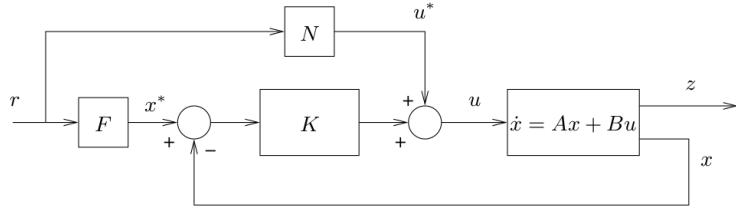


Figura 14.1.: Sistema realimentado

El problema de encontrar un regulador óptimo de horizonte infinito se puede plantear de la siguiente forma: Se trata de encontrar la matriz de transferencia $C(s)$ que minimice la siguiente función de costo:

$$J_{LQR} = \int_0^\infty X'(t)QX(t) + U'(t)RU(t)dt \quad (14.3)$$

Donde Q y R son matrices simétricas definidas positivas de dimensiones $n \times n$ y $m \times m$ respectivamente.

El primer término de la integral corresponde a la energía de los estados controlados y el segundo a la energía de la señal de control. En función de como se escogen las matrices Q y R , se obtienen resultados distintos. Por ejemplo si la norma de Q es pequeña la forma más efectiva de reducir J_{LQR} es utilizar señales de control de norma pequeña a expensas de tener grandes variaciones en los estados controlados. Si bien existen diversos métodos para determinar las matrices Q y R , gran parte del trabajo es iterativo y se realiza a ensayo y error.

14.2. Consideraciones particulares respecto del sistema a controlar

En la versión de realimentación de estados del problema LQR se asume que se disponen de medidas de todas las variables del vector de estados. En este caso, el controlador óptimo LQR es una matriz de ganancia K tal que:

$$U(t) - U^*(t) = -K(X(t) - X^*(t)) \quad (14.4)$$

Donde $K \in \mathcal{M}_{m \times n}$. Para el caso bajo análisis se tiene que:

$$K = R^{-1}B^TP \quad (14.5)$$

Donde P es la solución a la ecuación algebraica de Riccati.

La propiedad escencial del controlador LQR es que la respuesta del sistema en lazo cerrado es asintóticamente estable, es decir que la parte real de los valores propios de la matriz $A - BK$ es negativa, mientras se cumplan las siguientes condiciones:

- El sistema es controlable
- El sistema es observable

14.2. Consideraciones particulares respecto del sistema a controlar

Como se explicó en capítulos anteriores, se tiene una nueva estimación del vector de estados cada un cierto tiempo T^3 , lo cual nos permite realizar acciones de control con esta misma tasa de muestreo.

Al tener un tiempo de respuesta rápido (en relación a las constantes del sistema) se podría resolver el problema utilizando la técnica de control LQR pensando en un sistema continuo, por más que el sistema sea discreto. Los algoritmos de determinación de la solución a la ecuación algebraica de Riccati más sencillos que se encontraron corresponden a la solución de la formulación discreta del problema planteado en 14.3. Por otra parte, dicha formulación corresponde al caso en el cual se trabaja con un horizonte infinito, es decir se busca alcanzar un determinado punto de operación y el setpoint no será modificado. Estrictamente, esta no es la situación que se presenta dado que es de interés concatenar diversas trayectorias, por ende el punto de operación sí debe ser modificado, sin embargo se puede suponer que cada trayectoria se realizará por un tiempo ampliamente superior al tiempo del transitorio entre dos trayectorias. Por dicho motivo, aproximar el problema a un problema de horizonte infinito es acertado ya que simplifica enormemente la formulación del problema.

14.2.1. Discretización del sistema

Como se explicó en el capítulo 13 se trabaja con tres tipos de trayectorias: hovering, vuelos en linea recta y círculos. En cada uno de estos casos se tiene un sistema lineal de la forma

$$\dot{X} = AX + BU \quad (14.6)$$

³El período elegido en la implementación final es de 10ms.

14. Diseño del controlador

La forma que toma el sistema continuo al ser convertido a tiempo discreto es:

$$X_{k+1} = \Phi X_k + \Gamma U_k \quad (14.7)$$

donde:

$$\Phi = e^{AT_s} \quad \Gamma = \int_0^{T_s} e^{As} ds B \quad (14.8)$$

Estas relaciones surgen de discretizar el sistema considerando muestreadores de orden cero. Por más detalles de este proceso puede consultarse [15].

El problema de encontrar un regulador óptimo también puede ser planteado en un sistema discreto si reescribimos la ecuación 14.3.

$$J_{DLQR} = \sum_0^{\infty} X_k^T Q X_k + U_k^T R U_k \quad (14.9)$$

En el trabajo realizado en [14] se propone un algoritmo iterativo para obtener la matriz de realimentación que minimiza J_{DLQR} . Se comparó este algoritmo con la función *lqr* de *MatLab* obteniéndose resultados similares con ambos. Por dicho motivo se optó por reproducir exactamente el algoritmo de [14].

14.2.2. Agregado de integradores

El esquema de la figura 14.1 funciona a la perfección en el caso en que la caracterización del sistema sea perfecta. Los valores de setpoint fueron determinados analíticamente en función de los parámetros determinados, de forma que por ejemplo, para la trayectoria de hovering la velocidad angular de los motores es aquella que produce una fuerza igual al peso del sistema. Errores en la caracterización del sistema llevan a que el punto (o trayectoria) de equilibrio no sea alcanzable. Bajo la hipótesis que el valor de ω^* para la trayectoria de hovering es inferior al realmente necesario para mantener al sistema en reposo y que inicialmente $X = X^*$ la fuerza de empuje no es suficiente para mantener al sistema en equilibrio, por lo tanto la posición vertical disminuye. La realimentación proporcional produce un aumento en la velocidad angular de los motores hasta que para cierto valor de altura la entrada al sistema es aquella que produce el equilibrio. En este caso se obtiene un nuevo punto de equilibrio distinto al deseado. En el ejemplo considerado se obtiene una altura inferior a la deseada.

Realizando diversas pruebas se ha comprobado que según el nivel de batería disponible un mismo comando I^2C corresponde a distintos valores de velocidad angular. Para resolver este problema se plantea el camino de medir el voltaje en la batería durante el vuelo, sin embargo este camino no parece práctico ya que se debería caracterizar la relación entre velocidad angular y comando I^2C para todo el rango de voltajes en que la batería puede operar. Este camino no parece pertinente, quedando incluso sujeto a la exactitud con la cual se realizan las medidas.

Surge entonces la necesidad de agregar al controlador cierta robustez frente a errores de caracterización del sistema o frente a variaciones del mismo, por ejemplo

la tensión de la batería o un cambio en la masa del sistema⁴. Esto último es fundamental si se planea utilizar el cuadricóptero por ejemplo para transportar alguna carga útil.

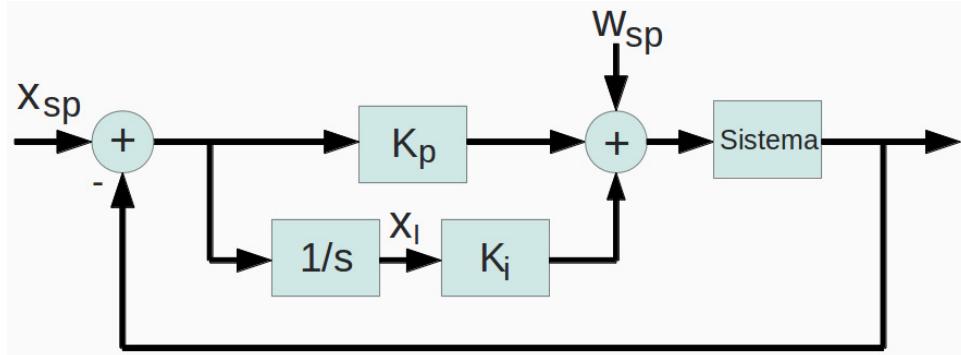


Figura 14.2.: Diagrama de bloques final utilizado

El camino a explorar es el de agregar un bloque integrador. El esquema del controlador es ahora el de la figura 14.2. Para explicar la utilidad del mismo, se continúa razonando con el ejemplo de una trayectoria de hovering en la cual ω^* está subestimado. En dichas condiciones se obtiene un nuevo punto de equilibrio con altura inferior a la deseada. Si se realimenta la integral de la diferencia de altura se tiene una entrada que aumenta con el tiempo hasta que la diferencia de altura sea cero. En este punto el término correspondiente a la integral de la diferencia de altura se mantiene constante en un valor positivo y el término correspondiente a la realimentación proporcional es cero, compensando el error en el modelado.

El agregado de un bloque integrador se puede entender como una ampliación del vector de estados. La salida del bloque integrador verifica que $\dot{X}_I = X$. Donde X_I corresponde a los estados integrados. La dinámica del sistema puede ser escrita de la siguiente forma:

$$\begin{pmatrix} \dot{X} \\ \dot{X}_I \end{pmatrix} = \begin{pmatrix} A & 0 \\ Id & 0 \end{pmatrix} \begin{pmatrix} X \\ X_I \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} U \quad (14.10)$$

En la ecuación 14.10 lo que se obtiene es un nuevo sistema lineal de mayor dimensión. Lo que se busca es encontrar una matriz de realimentación para el sistema 14.10 minimizando la función de costos definida en 14.9.

14.3. Controlabilidad y observabilidad

Los conceptos de controlabilidad y observabilidad son fundamentales para asegurar la estabilidad del sistema en lazo cerrado controlado con la técnica de LQR.

Sea el sistema lineal representado por la ecuación 14.11:

$$\begin{aligned} \dot{X} &= AX + BU \\ Y &= CX + DU \end{aligned} \quad (14.11)$$

⁴Para realizar pruebas se disponen de baterías de diverso tamaño (y peso), no parece práctico modificar constantemente la masa del sistema según la batería con la que se realiza una determinada prueba.

14. Diseño del controlador

donde $A \in \mathcal{M}_{n \times n}$, $B \in \mathcal{M}_{n \times m}$, $C \in \mathcal{M}_{p \times n}$ y $D \in \mathcal{M}_{p \times m}$.

La matriz de controlabilidad $S \in \mathcal{M}_{n \times nm}$ se define como:

$$S = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (14.12)$$

El sistema es controlable si y solo si el rango de S es n.

La matriz de observabilidad $V \in \mathcal{M}_{pn \times n}$ se define como:

$$V = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (14.13)$$

El sistema es observable si y solo si el rango de V es n.

El sistema bajo estudio es originalmente no lineal. En el capítulo 13 se estudió en torno a que trayectorias se puede aproximar el problema de controlar el cuadricóptero por el problema de controlar un sistema lineal invariante en el tiempo. Si bien no todas las trayectorias son posibles, el conjunto de trayectorias es infinito. Notar que por ejemplo dos vuelos en línea recta a dos velocidades distintas son, en realidad, dos trayectorias distintas. Evidentemente, es imposible evaluar las condiciones de controlabilidad y observabilidad para todos los casos. Sin embargo, se puede evaluar en algunas trayectorias y confiar en que para las trayectorias no testeadas el resultado sea el mismo.

En una primer aproximación se analizaron diversas trayectorias para el sistema lineal sin los estados agregados correspondientes a la inclusión de integradores. Las matrices de controlabilidad y observabilidad para todas las trayectorias testeadas son de rango 12. Esto implica que los controladores que surgen de la técnica de control LQR son estables. Sin embargo es de interés, por los motivos presentados en la sección anterior, poder agregar al controlador una dependencia del pasado (integradores), esto implica cuestionarse que estados integrados pueden ser incluidos en el sistema manteniendo la condición de controlabilidad. La condición de observabilidad se cumple trivialmente ya que siempre se tiene una estimación de las integrales de los estados.

Los únicos estados que pueden integrarse sin perder la controlabilidad del sistema son las tres posiciones (x, y, z) y el ángulo de Yaw (θ). En resumen, el vector de estados controlados es el siguiente:

$$X = (x, y, z, \psi, \varphi, \theta, v_{q_x}, v_{q_y}, v_{q_z}, \omega_{q_x}, \omega_{q_y}, \omega_{q_z}, x_I, y_I, z_I, \theta_I)^T \quad (14.14)$$

Observabilidad y controlabilidad pueden ser testeadas ejecutando en *MatLab*:

- `rank(obsv(A,C));`
- `rank(ctrb(A,B));`

Las funciones obsv y ctrb computan las matrices de observabilidad y controlabilidad respectivamente. Si el rango de las mismas es igual a n el sistema es observable y controlable. En la documentación de *MatLab* se aclara que la función obsv no es siempre adecuada para realizar este test, se recomienda la función obsvf. Esta función devuelve una descomposición en estados observables e inobservables, la transformación necesaria para obtener dicha descomposición y un vector k que contiene la cantidad de estados observables identificados en cada iteración. La suma de dicho vector resulta en la cantidad de estados observables. Se debe ejecutar en *MatLab*:

- `[ABAR,BBAR,CBAR,T,k] = obsvf(A,B,C);`
- `sum(k);`

14.4. Matriz de realimentación

Como se explicó anteriormente, la matriz de realimentación es obtenida numéricamente gracias al algoritmo presentado en [14]. Dicha matriz depende del sistema lineal con el cual se representa cada trayectoria. Por lo tanto para cada trayectoria se debe calcular una nueva matriz de realimentación. Lo fundamental es entonces determinar adecuadamente las matrices Q y R presentadas en 14.9 de forma de obtener trayectorias adecuadas. Estas matrices fueron determinadas en forma iterativa, modificando los parámetros hasta obtener un comportamiento satisfactorio en cuanto a tiempos de respuesta y robustez.

Se entiende que los parámetros fundamentales a controlar son los ángulos de Euler, si el control sobre estos no se hace en forma rápida la performance del sistema puede verse afectada. Las condiciones de vuelo determinadas dependen de que los ángulos de Pitch y de Roll se encuentren perfectamente ajustados. A modo de ejemplo si estos ángulos no son nulos se tiene un sistema que acelera indefinidamente en alguna dirección. Por lo tanto parece importante darle un peso superior al control de estas variables de estado respecto a las restantes. Las matrices de realimentación se construyen a partir de una matriz (Q) diagonal, cuyos elementos son:

$$\text{diag}(Q) = \{1, 1, 1, 10^3, 10^3, 10^3, 1, 1, 1, 1, 1, 1, 1, 1, 1\} \quad (14.15)$$

$$R = \begin{pmatrix} 0,1 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0,1 \end{pmatrix} \quad (14.16)$$

CAPÍTULO 15

SIMULACIONES REALIZADAS SOBRE EL CONTROLADOR

El objetivo de la presente sección es el de analizar el desempeño del controlador. En primer lugar se realiza un estudio en lo que respecta a la respuesta al escalón para las tres trayectorias definidas en el capítulo 13, luego se analiza el comportamiento del sistema frente a medidas con ruido simulado para finalmente estudiar el desempeño del controlador frente a errores y no idealidades en el modelado del sistema.

En lo que respecta al análisis de la respuesta del controlador ante la presencia de ruido en las medidas se utilizará la función del simulador desarrollado que permite agregar a cada estado un ruido que cumple con el siguiente modelo:

$$\eta_i = A_i \cos(\omega t) + \varepsilon(\mu_i, \sigma_i) \quad (15.1)$$

Donde $\varepsilon(\mu_i, \sigma_i)$ es un ruido gaussiano de media μ_i y de desviación estándar σ_i .

15.1. Respuesta al escalón

Esta sección se centra en el análisis de las respuestas al escalón de los tres tipos de trayectorias. En esta sección no se consideran medidas con ruido, es decir que se conoce el estado a la perfección.

15.1.1. Trayectoria de hovering

Desplazamientos en la dirección vertical

Se consideran condiciones iniciales nulas. Se fija como setpoint:

- Prueba 1: $x_s = 0m; y_s = 0m; z_s = 1m; \theta = 0^\circ$
- Prueba 2: $x_s = 0m; y_s = 0m; z_s = 10m; \theta = 0^\circ$

No se aprecian variaciones en ninguna de las variables de estado excepto en la posición vertical (z) y en la velocidad v_{q_z} . La respuesta al escalón para las pruebas uno y dos se muestran en la figura 15.1.

15. Simulaciones realizadas sobre el controlador

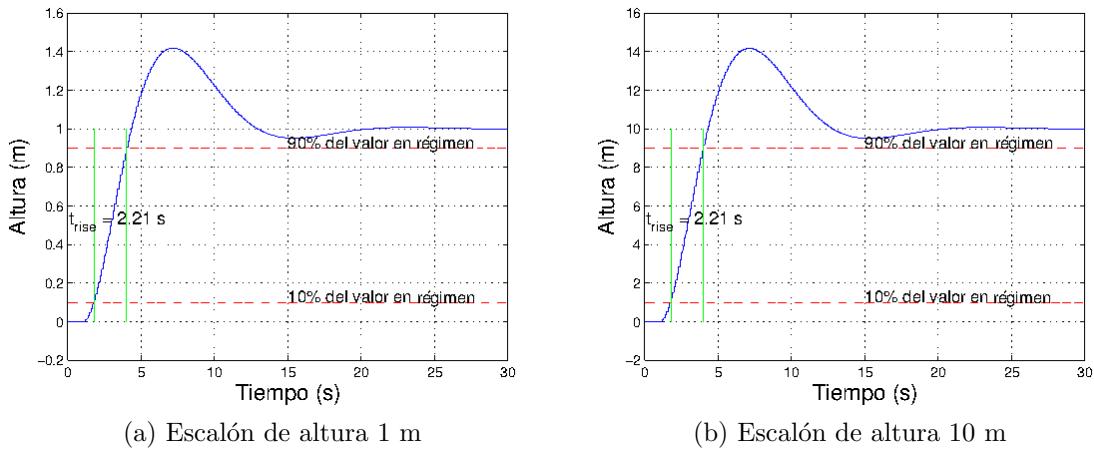


Figura 15.1.: Respuesta al escalón en la altura

El tiempo de *rise* de ambas respuestas al escalón es considerado ampliamente satisfactorio. El sobretiro obtenido es superior a lo deseable, siendo aproximadamente un 40 % del valor en régimen. El mismo puede reducirse otorgando un peso menor en la matriz 14.15 a los términos asociados al estado integrado de la altura. Esta modificación produce a la vez que el controlador se comporte considerablemente peor frente a variaciones en el sistema. Esto será analizado con mayor detalle en la sección 15.3. Existe un compromiso entre los fenómenos descriptos, la matriz de realimentación escogida busca contemplarlos a ambos determinando un valor considerado como óptimo. Otra forma de reducir el sobretiro es acotar tanto el paso de integración como la integral en sí. Este ajuste se realizará en la práctica, quedando por fuera de las simulaciones.

En la figura 15.2 se observa la velocidad angular de uno de los motores para el escalón de altura 10m. Se observa que en los instantes iniciales la velocidad angular de uno de los motores alcanza un valor cercano al máximo (387rad/s), por lo tanto no se puede asegurar el perfecto funcionamiento para escalones de mayor altura. Por otra parte se observa en ambas pruebas que una vez alcanzado el valor objetivo se mantiene perfectamente.

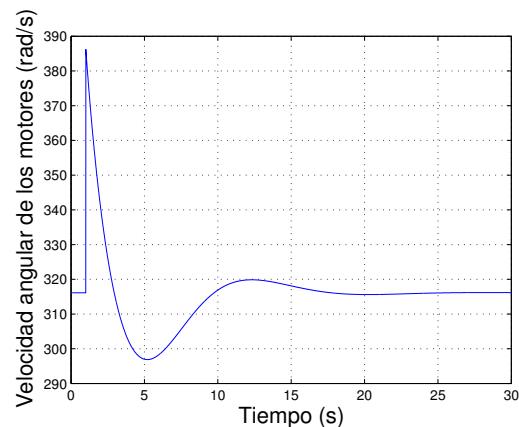


Figura 15.2.: Velocidad angular del motor 1 durante un escalón de altura 10m

Desplazamientos en la dirección horizontal

Partiendo de condiciones iniciales nulas excepto la altura donde se considera $z = 3m$, se fija el siguiente setpoint:

- $x_s = 5m; \quad y_s = 0m; \quad z_s = 3m; \quad \theta = 0^\circ$

En la figura 15.3 se observan las características más importantes de la trayectoria. Para lograr un desplazamiento en alguna dirección horizontal el cuadricóptero debe realizar al menos un giro según los ángulos de pitch o roll. Esto se observa claramente en la figura 15.3a. Este giro produce una variación en la altura. Para el caso considerado, la máxima variación es cercana a los $40cm$, valor que se considera aceptable. Para mayores escalones en la dirección horizontal se observan variaciones de altura no deseables. Por otra parte en lo que respecta a la respuesta al escalón en la dirección \vec{i}_I se tiene un tiempo de *rise* y un sobretiro muy similar que en el caso del escalón vertical.

En resumen, el controlador diseñado se comporta adecuadamente para la condición de hovering. Cabe recordar que la linealización realizada para dicha trayectoria supone el reposo del sistema y los desplazamientos no se encuentran a priori contemplados en dicha trayectoria. El controlador diseñado permite algunos desplazamientos con un buen tiempo de respuesta. Para desplazamientos de mayores distancias se debe trabajar con rectas o círculos.

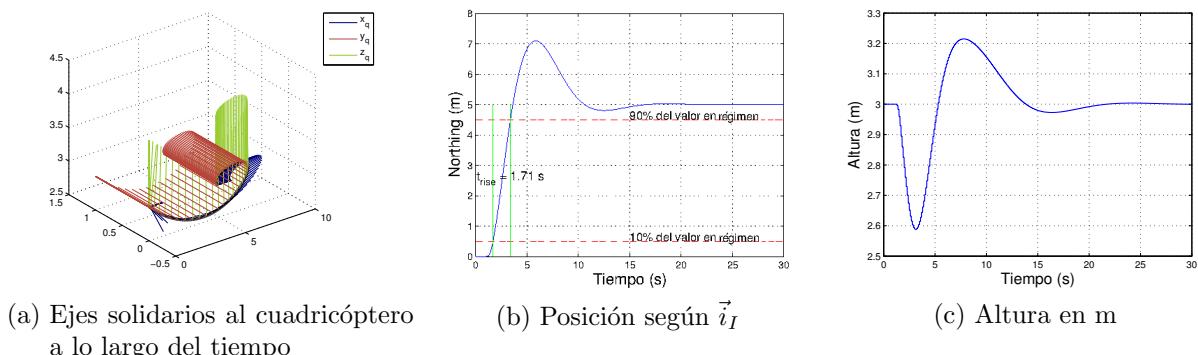


Figura 15.3.: Escalón de 5m en la dirección horizontal

15.1.2. Vuelo en linea recta

Partiendo de condiciones nulas se impone un escalón de $5m/s$ según \vec{i}_q . Para alcanzar una velocidad superior a la inicial en la dirección \vec{i}_q es necesario que el cuadricóptero modifique su ángulo de pitch. Esto produce que la fuerza neta en la dirección vertical sea inicialmente inferior a la necesaria para mantener el equilibrio, produciendo así un descenso en la altura. Para que el resultado obtenido sea aceptable dicha modificación no puede ser significativa. Del mismo modo se busca que el valor de velocidad en régimen se alcance en un tiempo razonable, así como se busca que el sistema sea capaz de seguir en posición a la integral de la velocidad de setpoint. En la figura 15.5 se observa la altura del sistema, la velocidad en la dirección \vec{i}_q y la posición del cuadricóptero según \vec{i}_I .

15. Simulaciones realizadas sobre el controlador

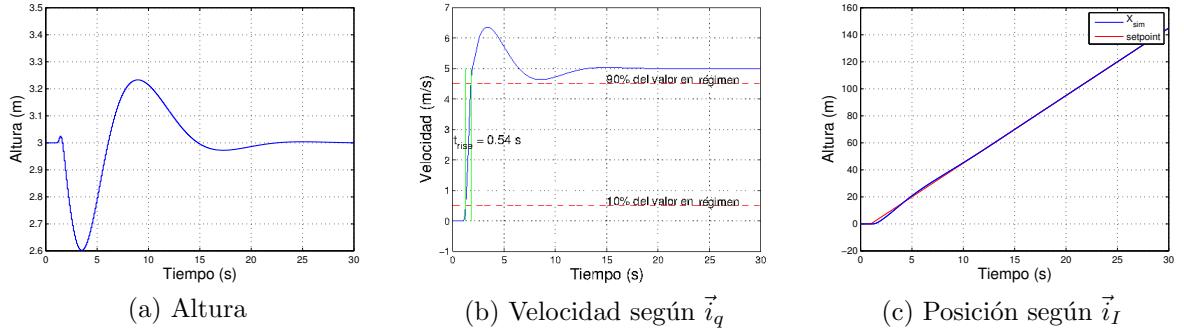


Figura 15.4.: Escalón de $5m/s$ en la dirección \vec{i}_q

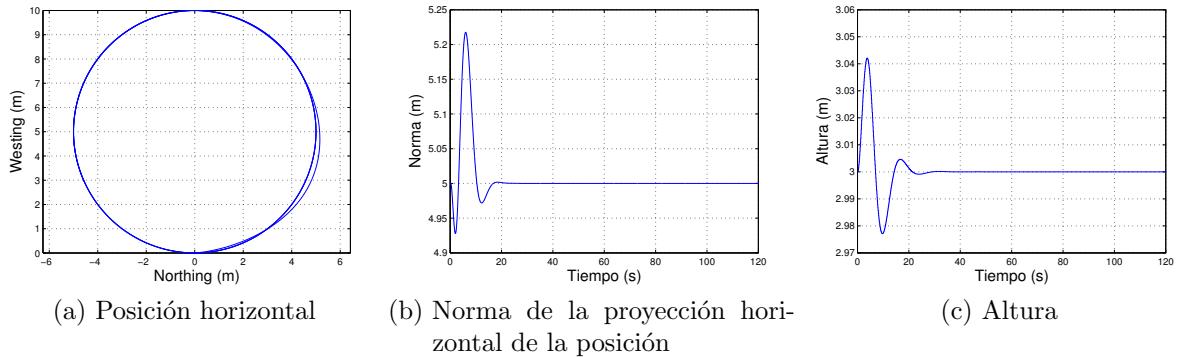


Figura 15.5.: Escalón de $5m/s$ en la dirección \vec{i}_q

La respuesta del sistema es rápida, alcanzando el 90 % del valor de régimen en $0,54s$, asímismo como puede observarse en 15.4c la posición se sigue de forma muy acertada. Para poder seguir la posición es necesaria la presencia de un sobretiro en la velocidad. Finalmente cabe observar que la máxima variación en la altura es cercana a los $40cm$, valor que se considera aceptable.

15.1.3. Vuelo en círculos

Para verificar el correcto comportamiento del controlador en trayectorias circulares se realiza una simulación partiendo del reposo, con $x = y = 0m$ y $z = 3m$. Se desea realizar un círculo a velocidad angular $0,2rad/s$ y con velocidad tangencial de $1m/s$. Estas condiciones dan origen a un círculo de radio $5m$ ya que $V = \omega R$. En la figura 15.5a puede observarse el resultado arrojado por la simulación para la proyección en el plano horizontal de la posición, mientras que en la figura 15.5c se observa la altura del sistema. Los resultados arrojados por la simulación son ampliamente satisfactorios puesto que la trayectoria descripta es un círculo perfecto de radio $5m$ una vez que se alcanza el régimen. En la figura 15.5b se presenta la norma de la proyección horizontal de la posición. Se observa que el tiempo en que se alcanza el valor de régimen es inferior a los 20 segundos, teniendo un error máximo de la norma del orden del 5 %. Adicionalmente en la figura 15.5c se observa que la altura no se modifica más de $4cm$.

Los resultados observados hasta el momento aseguran el buen comportamiento

del sistema en situaciones ideales en las cuales se conoce el estado exactamente (no hay ruido de medida ni errores) y donde el sistema se comporta exactamente de acuerdo al modelo realizado.

15.2. Robustez frente a la presencia de ruido de medición

Hasta aquí se han evaluado las características del controlador diseñado en situaciones en las cuales se conoce exactamente el vector de estados, evidentemente la situación de vuelo real no se corresponde con esta idealidad. Las medidas realizadas tienen ruido intrínseco, muy superior a los considerados en las pruebas de calibración y caracterización de los sensores. El aumento de este ruido se debe principalmente a las vibraciones mecánicas que introducen los motores.

Resulta fundamental conocer en forma aproximada el comportamiento de los ruidos de medida. Con dicho fin se realizó una prueba de vuelo en la cual el cuadricóptero estaba sujeto por arriba y por debajo quedando imposibilitado de realizar movimientos. Se comandaron los motores a la velocidad de hovering. En la presente sección se presentan los parámetros de la ecuación 15.1 que mejor ajustan el ruido. Si bien se podría haber realizado dicho ajuste utilizando mínimos cuadrados, al menos para la parte del ruido no aleatorio, se optó por realizar dicho ajuste en forma iterativa.

Dado que los ángulos de pitch y de roll se obtienen directamente con los acelerómetros, los ruidos asociados a ambas medidas son idénticas. Por dicho motivo consideraremos un solo ruido replicado en ambas variables. El simulador ofrece la posibilidad de agregar ruido en todas las variables, sin embargo no se realizará un análisis para cada una de las variables afectada por el ruido. Serán presentados los resultados en lo que respecta a los ángulos de Euler dada la importancia que tienen los mismo en el global del sistema, se analizará además como afecta el ruido de medida a una velocidad angular y a la altura.

Finalmente cabe aclarar que se espera que los ruidos obtenidos sean independientes del tipo de trayectoria realizada. Se analiza la performance del controlador ante medidas ruidosas exclusivamente para la trayectoria de hovering.

15.2.1. Hovering

De acuerdo al ruido presente en las medidas de los ángulos de pitch y de roll obtenidas de los sensores en situación de vuelo, que es mayormente mecánico debido a las perturbaciones de los motores, se escogen los siguientes parámetros:

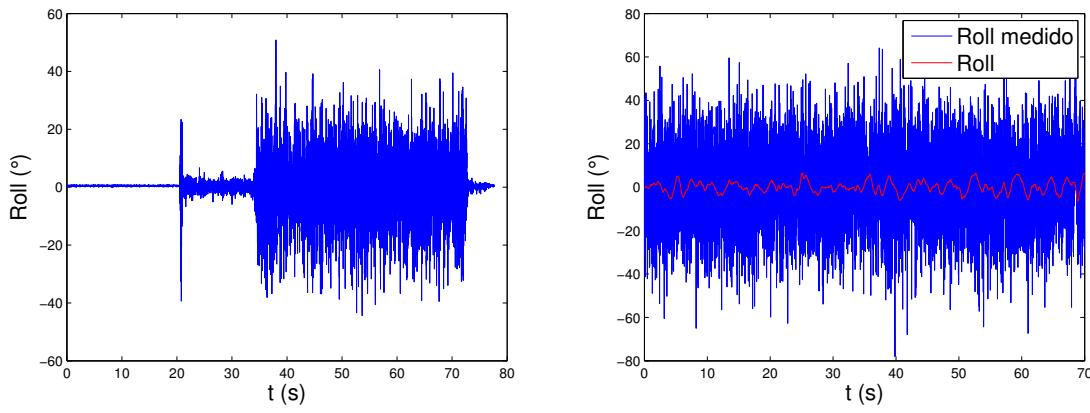
- $A_{roll} = 0,05^\circ$
- $\omega_{i,roll} = 2\pi 0,03 rad/s$
- $\sigma_{roll} = 6,05^\circ$

15. Simulaciones realizadas sobre el controlador

- $\mu_{roll} = 0,27^\circ$

En la figura 15.6 pueden compararse los ruidos obtenidos en la “situación de vuelo” descripta anteriormente y el ruido en la simulación. Asimismo, en la figura 15.6b puede observarse como, a pesar de la gran presencia de ruido, la variable de estado de interés se mantiene controlada muy cercana al valor deseado. El valor máximo y mínimo de roll alcanzados son $6,48^\circ$ y $-6,41^\circ$.

En la figura 15.6a se observan tres amplitudes de ruido bien marcadas. La primera corresponde a las medidas con los motores apagados, la segunda a partir del segundo 20.5 donde los motores operan a velocidad mínima (109 rev/s) y la tercera a partir del segundo 34 donde la velocidad angular de los motores es igual a la velocidad de hovering. El análisis se centra exclusivamente en esta tercer etapa.



(a) Medida del ángulo de Roll en “situación de vuelo” (b) Ruido de Roll simulado y Roll efectivo

Figura 15.6.: Ruidos de Roll

El ángulo de Yaw se determina además utilizando la lectura del magnetómetro, por dicho motivo se separa el análisis de su ruido de los restantes ángulos de Euler. En este caso los parámetros de ruido utilizados son:

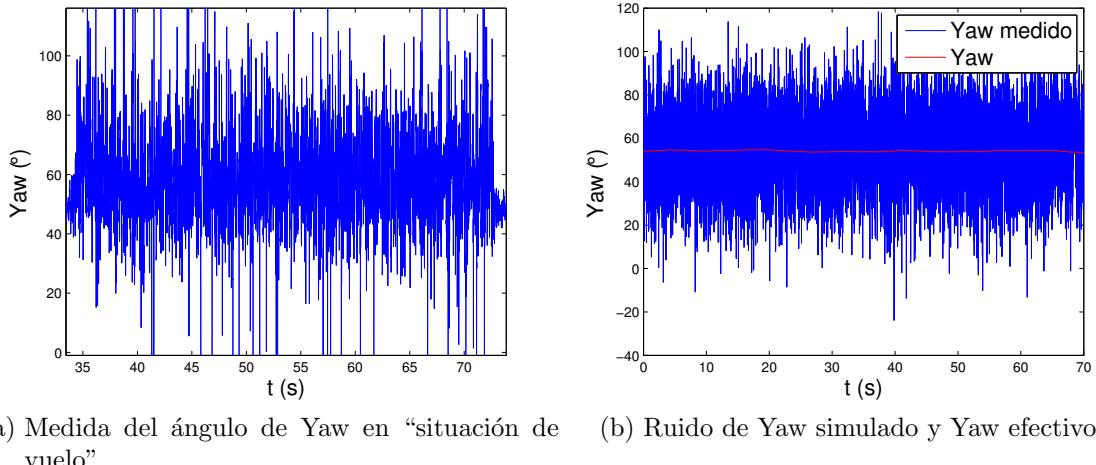
- $A_{yaw} = 0,23^\circ$
- $\omega_{yaw} = 2\pi 0,07 \text{ rad/s}$
- $\sigma_{yaw} = 2^\circ$
- $\mu_{yaw} = 0$

En la figura 15.7 puede observarse como, a pesar del ruido en la medida el controlador se mantiene robusto apartándose del valor objetivo $0,96^\circ$ en el peor de los casos. El resultado en lo que respecta al control de la velocidad angular del sistema es similar. En la figura 15.8 pueden observarse las gráficas de los ruidos medido y simulado, además de la velocidad angular simulada. Se trabaja en este caso con la velocidad angular ω_{qx} . Los parámetros de ruido escogidos son:

- $A_{\omega_{qx}} = 0,03^\circ/\text{s}$

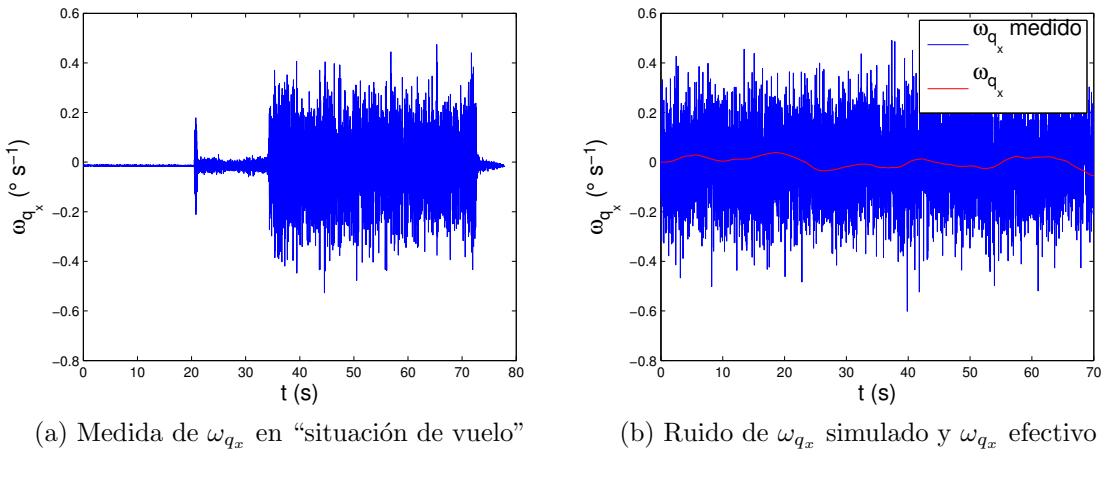
15.2. Robustez frente a la presencia de ruido de medición

- $\omega_{\omega_{qx}} = 2\pi 0,03 \text{ rad/s}$
- $\sigma_{\omega_{qx}} = 0,64^\circ/\text{s}$
- $\mu_{\omega_{qx}} = 0$



(a) Medida del ángulo de Yaw en “situación de vuelo” (b) Ruido de Yaw simulado y Yaw efectivo

Figura 15.7.: Ruidos de Yaw



(a) Medida de ω_{qx} en “situación de vuelo”

(b) Ruido de ω_{qx} simulado y ω_{qx} efectivo

Figura 15.8.: Ruidos de ω_{qx}

Los ruidos observados hasta el momento son preponderadamente blancos. Por dicho motivo es prácticamente imposible observar la respuesta del control más allá de afirmar que efectivamente no nos alejamos sustancialmente de la posición deseada. En el caso del ruido en la altura la situación es distinta ya que quien juega el papel más importante en el ruido es una oscilación de baja frecuencia. Los parámetros elegidos para representar dicho ruido son:

- $A_z = 1m$
- $\omega_z = 2\pi 0,04 \text{ rad/s}$
- $\sigma_z = 0,34m$
- $\mu_z = 0,94m$

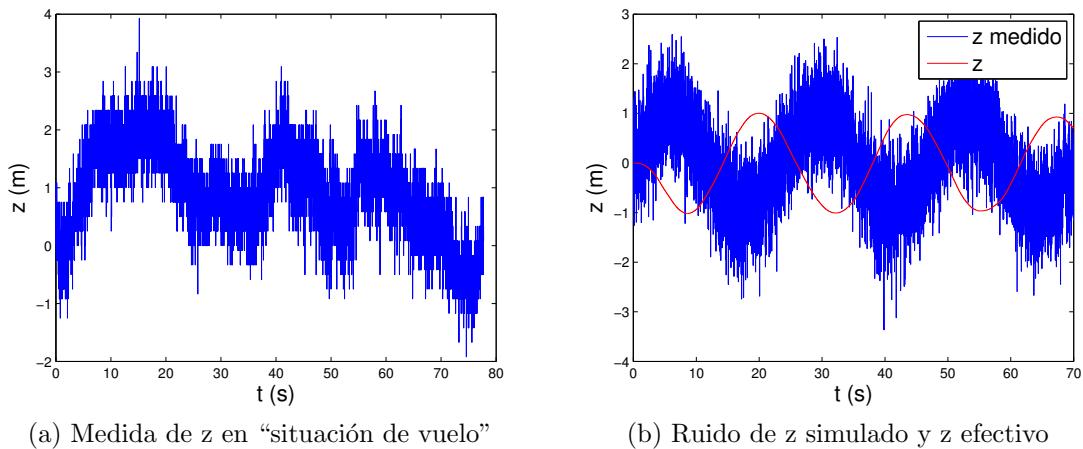


Figura 15.9.: Ruidos de z

En la figura 15.9b se aprecia claramente la acción del control ya que para medidas que superan el valor deseado de *setpoint* el control actúa en sentido contrario, sucede lo mismo para las medidas que son inferiores al *setpoint*.

Los ruidos que se han manejado hasta aquí corresponden a las medidas directas de los sensores. Al realizar la estimación del estado con filtro de Kalman (ver capítulo 12) se obtendrá un ruido sensiblemente inferior al obtenido directamente de los sensores. Por lo tanto podemos asumir que el sistema se comportará aún mejor de lo que se evaluó en esta sección.

En lo que respecta al ruido de la velocidad no se puede trabajar simplemente con las medidas de los sensores ya que no se tiene ninguna medida directa de la velocidad. Se podría integrar la medida de aceleración obtenida de los acelerómetros para obtener valores de velocidad, pero el error introducido en la integración implican una deriva en la velocidad inaceptable, por lo que no tiene sentido en pensar esta magnitud con un ruido asociado, excepto que se trabaje con las estimación del vector de estados.

15.3. Robustez frente a variaciones en el modelo

En este punto se analiza el comportamiento del sistema realimentado frente a algunas variaciones en el mismo. Al igual que en la sección anterior se trabaja con la trayectoria de hovering. Se estudian los dos casos que tienen más importancia práctica: la variación de la masa y el hecho de que los motores no son perfectamente idénticos, causando que sus respuestas al mismo comando i^2c no sea la misma.

Variación de la masa

Con condiciones iniciales nulas excepto por la altura que se fija a 3m, se aumenta la masa del sistema en el orden del 5% (100g). En la figura 15.10 se observa la variación de la altura por el agregado de masa.

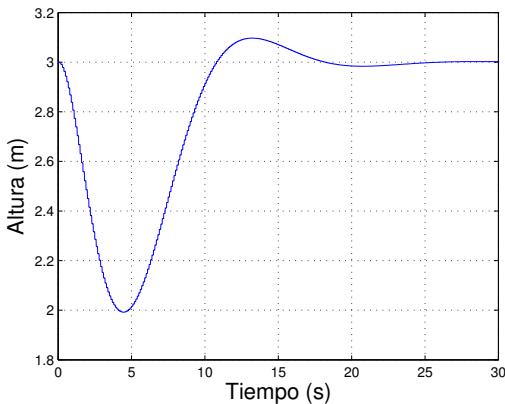


Figura 15.10.: Variación de la altura al agregar una masa adicional de 100g

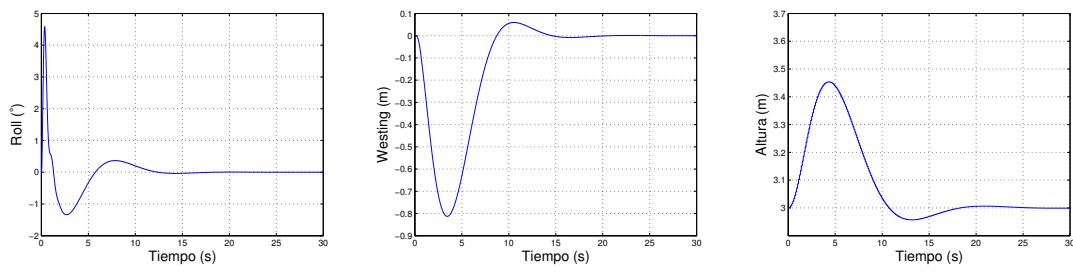


Figura 15.11.: Diferencia de $15,2\text{rad/s}$ entre las velocidades angulares de los motores 2 y 4.

La respuesta frente a la variación de la masa es buena ya que en aproximadamente de diez segundos se vuelve a obtener la altura inicial. Adicionalmente la variación de la altura es de un metro.

Diferencias entre las velocidades angulares de los motores

En las mismas condiciones que en la situación anterior se impone que la velocidad angular del motor 2 cumpla:

$$\omega_2 = \omega_4 + 15,2\text{rad/s} \quad (15.2)$$

La diferencia de $15,2\text{rad/s}$ corresponde a la máxima diferencia observada entre las velocidades angulares de los motores. A priori se espera que esta diferencia produzca un giro positivo en el ángulo de Roll y un desplazamiento en la dirección $-\vec{j}_I$. Se espera también un desplazamiento en la dirección vertical debido a que la fuerza neta es mayor, ya que continuamos asumiendo que $\omega_1 = \omega_3 = \omega_4 = \omega_{\text{hover}}$.

En la figura 15.11 se observa la variación de estas tres variables en función del tiempo. La inclusión del bloque integrador es fundamental para poder controlar el sistema frente a esta variación en las velocidades angulares de los motores. Sin dicho bloque el sistema se estabiliza, pero en una posición diferente de la deseada.

15. Simulaciones realizadas sobre el controlador

En resumen el controlador diseñado se presenta como satisfactorio ya que permite seguir las trayectorias deseadas con buenos tiempos de respuesta. Por otra parte, su respuesta frente a medidas extremadamente ruidosas (en la práctica se trabaja con estados mejor estimados) es muy buena, ya que mantiene al sistema en torno al valor objetivo, aunque evidentemente presenta algunas variaciones debidas al error en la medida, que son aceptables. Finalmente cabe señalar la robustez del mismo frente a variaciones en el sistema como ser la masa y una diferencia entre las velocidades angulares de los motores.

El sobretiro en la respuesta al escalón parece superior al deseado y es un punto que debe mejorarse en el futuro si se desea un control más preciso del sistema. En esta primer aproximación al problema los sobretiros presentados son aceptables. La mayor dificultad que se encontró al intentar reducir el sobretiro es que el sistema presenta mayores variaciones frente a variaciones de masa o de las velocidades angulares de los motores. Una posibilidad es la de imponer valores mayores en la diagonal de la matriz R definida en 14.16. Sin embargo esta solución implica que el sistema tenga tiempos de respuesta más lentos.

La matriz de realimentación obtenida mediante simulaciones no será necesariamente la que se utilizará en la práctica, pueden existir otros factores no considerados que lleven a modificar alguno de los parámetros hasta aquí elegidos. Sin embargo el trabajo sobre el simulador ofrece la posibilidad de tener un punto de partida más cercano a la solución definitiva para el control autónomo del cuadricóptero.

CAPÍTULO 16

PRUEBAS DEL CONTROLADOR

El controlador diseñado se comporta adecuadamente en lo que respecta a las simulaciones, sin embargo debido a que la caracterización del sistema puede contener errores se procede a realizar algunas pruebas sobre los subsistemas que componen al sistema global como paso intermedio antes de realizar una prueba de vuelo real. Estas pruebas son de utilidad para verificar el correcto funcionamiento del controlador diseñado y/o para realizar los ajustes que sean necesarios en el mismo.

16.1. Control de subsistemas Roll y Pitch

Para lograr que el cuadricóptero se mantenga horizontal, es fundamental que el control sobre los ángulos de Pitch y de Roll se comporten de acuerdo a lo esperado. A modo de ejemplo, es imposible lograr el equilibrio mecánico si dichos ángulos difieren de cero. Por dicha razón, previo a realizar pruebas sobre el sistema completo es necesario asegurarnos que los subsistemas de Roll y Pitch funcionan correctamente. De acuerdo al modelo físico del sistema desarrollado en 3 ni el Roll ni el Pitch son subsistemas independientes entre sí, además ambos dependen de la velocidad angular segú \vec{k}_q . Sin embargo, dichos ángulos toman valores cercanos a cero en las trayectorias de interés, caso en el cual se puede realizar la aproximación de que ambos sistemas son independientes.

A continuación se explicará solamente el funcionamiento para el ángulo de Roll, ya que para Pitch el procedimiento es completamente análogo.

A partir de esta consideración se procede a fijar al cuadricóptero sobre dos guías como se muestra en la figura 16.1, de forma de eliminar todos los grados de libertad del sistema excepto el ángulo de Roll (ψ) y la velocidad angular correspondiente al eje de rotación de este ángulo. Se realizan dos pruebas: la primera consiste en que el sistema alcance la posición de equili-



Figura 16.1.: Dispositivo de prueba de Roll

16. Pruebas del controlador

brio ($\psi = 0$), la segunda consiste en alejar al sistema del equilibrio y lograr que vuelva al punto de equilibrio.

El controlador posee dos términos proporcionales: uno para el ángulo ψ y el otro para la velocidad angular ω_{qx} . Además se considera un término integral asociado a la integral de ψ . El modelo de este subsistema es el siguiente:

$$\begin{pmatrix} \dot{\psi} \\ \dot{\omega}_{qx} \\ \dot{\psi}_I \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{MgL'}{I_{xx}} & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi \\ \omega_{qx} \\ \psi_I \end{pmatrix} + b \begin{pmatrix} 0 & 0 \\ 1 & -1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_2 \\ \omega_4 \end{pmatrix} \quad (16.1)$$

donde b y $\frac{MgL'}{I_{xx}}$ se calculan a partir de las caracterizaciones realizadas anteriormente. La matriz Q difiere de la utilizada en las simulaciones ya que el error sistemático implicaba tiempos de corrección del ángulo mayores a lo aceptable. Se utilizan las matrices Q y R :

$$Q = \begin{pmatrix} 1000 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{pmatrix} \quad R = \begin{pmatrix} 0,1 & 0 \\ 0 & 0,1 \end{pmatrix} \quad (16.2)$$

la matriz de realimentación obtenida es la mostrada en la ecuación 16.3

$$K = \begin{pmatrix} 38,32 & 10,78 & 21,36 \\ -38,32 & -10,78 & -21,36 \end{pmatrix} \quad (16.3)$$

En la figura 16.2 se observa la respuesta del ángulo de Roll en lazo cerrado, con el setpoint fijo en $\psi = 0$. Se observa que el módulo del ángulo es siempre inferior a los 2° a excepción del arranque y algún pico aislado. Se puede decir que presenta un error típico de $\pm 1^\circ$. En este sentido se puede afirmar que el control implementado es exitoso, ya que logra el objetivo planteado. Puede observarse además que una vez que el controlador comienza a actuar se produce un cambio en el ángulo alcanzando un valor cercano a los 5° . Este error es producido por la diferencia del empuje de los motores frente a una misma orden. El control integral es el encargado de corregir esta diferencia en aproximadamente 2,5 segundos, tiempo que dependiendo de la aplicación puede ser o no aceptable. Un ángulo de algunos grados durante 2 segundos provocará un desplazamiento que puede llegar a ser inaceptable dependiendo de la aplicación. Como se sabe que es causado por una no idealidad sobre el empuje de los motores ante igual comando, es posible evitar este desplazamiento inicial muy fácilmente con tan solo inicializar al integrador en algún valor apropiado (distinto de cero). De esta forma se logra un despegue más prolífico ya que el ángulo ψ permanecerá todo el tiempo más cerca de cero.

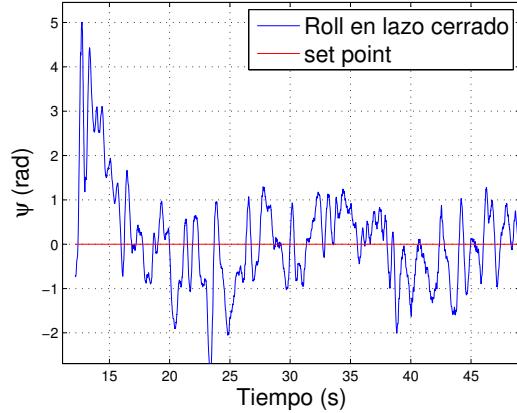


Figura 16.2.: Ángulo de Roll en lazo cerrado

Ante este tipo de imperfecciones el control proporcional es el encargado de volver a estabilizar al cuadricóptero, pero lo hará en algún otro punto de equilibrio que no necesariamente es el deseado. Por otro lado es el control integral el encargado de que el cuadricóptero alcance el equilibrio deseado, que será probablemente distinto al equilibrio hallado por el proporcional.

En la figura 16.3 se puede observar la diferencia entre utilizar un controlador puramente proporcional y un controlador proporcional con una corrección integral. El primero no logra corregir el error sistemático debido a la diferencia en el empuje de los motores alcanzando así un punto de equilibrio distinto del *set point*. El controlador con el término integral sí logra corregir este error y el ángulo ψ toma valores en el entorno del *set point*.

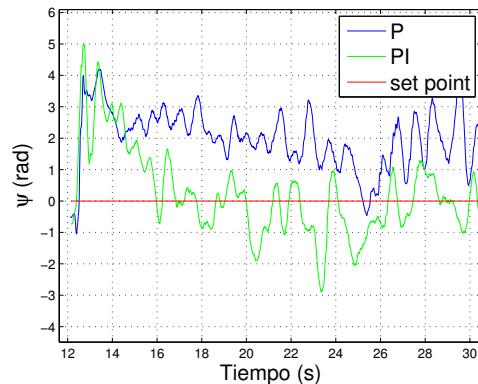


Figura 16.3.: Ángulo de Roll en lazo cerrado

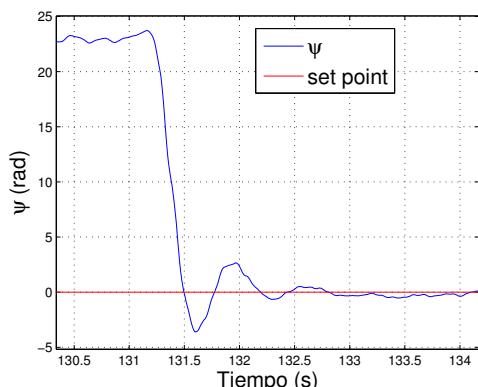


Figura 16.4.: Respuesta al escalón de Roll en lazo cerrado

sumamente satisfactorio.

Con este análisis se concluye que la matriz de realimentación obtenida es adecuada para controlar el subsistema del ángulo de Roll.

16.2. Control del subsistema del Yaw

De manera análoga al caso anterior, es importante verificar el buen funcionamiento del control sobre el giro en “z”, para lo cual se utiliza un dispositivo de prueba que restringe los grados de libertad del cuadricóptero. En este caso se lo sujetó con una cuerda desde arriba de los cuatro brazos de modo de realizar la fuerza lo más

16. Pruebas del controlador

pareja posible, como se muestra en la figura 16.5. El cuadricóptero queda sujetado colgando horizontal y conserva el libre giro según “z” (θ). Se setea una velocidad de *hovering* inferior a la necesaria para levantar vuelo, de modo que el cuadricóptero no se eleve y la cuerda quede siempre tensa. Si bien esta diferencia de velocidad genera diferencias en el comportamiento del sistema, el comportamiento respecto de la situación de vuelo será similar. En definitiva resulta una buena forma de verificar globalmente el comportamiento del sistema realimentado.

El giro en θ es generado por un desequilibrio entre los pares ejercidos por las hélices. Si el par neto de todas las hélices resulta por ejemplo positivo, el cuadricóptero realizará un movimiento hacia los negativos, equilibrando el par, como se explica en el capítulo 1.

Para la estimación de θ se utiliza por un lado la integral de la velocidad

angular en el eje “z” y por otro la proyección del vector del campo magnético medido sobre el plano horizontal, medidas que son combinadas en el filtro de Kalman. El dato obtenido del magnetómetro no distingue entre giros de 360° , limitando el valor al rango $[-180^\circ : 180^\circ]$. Es necesario entonces realizar un reajuste de la medida deducida del campo magnético para lograr la continuidad en el ángulo estimado.

Análogamente a la parte anterior se realizan 2 pruebas, la primera dejando que el sistema alcance su posición de equilibrio en $\theta = \theta_{inicial}$, y la segunda apartándolo del equilibrio y dejándolo regresar a su punto de *set point*.

El controlador, tal como sucede con los 2 ángulos anteriores (ψ y ϕ), consta de 2 términos proporcionales, uno para θ y otro para w_{qz} , además de un término integral en θ . El subsistema queda descripto por la siguiente ecuación:

$$\begin{pmatrix} \dot{\theta} \\ \dot{w}_{qz} \\ \dot{\theta}_I \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ w_{qz} \\ \theta_I \end{pmatrix} + b \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{pmatrix} \quad (16.4)$$

En este caso la matriz de realimentación utilizada es la siguiente:

$$K = \begin{pmatrix} -48,93 & -7,51 & -14,73 \\ 48,93 & 7,51 & 14,73 \\ -48,93 & -7,51 & -14,73 \\ 48,93 & 7,51 & 14,73 \end{pmatrix} \quad (16.5)$$

Los resultados obtenidos se muestran en las siguientes figuras. Se analiza en primer lugar el comportamiento del cuadricóptero dejándolo en equilibrio, luego se analiza la respuesta al escalón del subsistema en θ y por último se compara el comportamiento si se utiliza o no el término del integrador.



Figura 16.5.: Dispositivo de prueba de θ

En la figura 16.6 se observa la dinámica del ángulo θ obtenida en el dispositivo de prueba de la figura 16.5 utilizando un integrador en dicho ángulo. Nuevamente se puede observar un crecimiento al principio que es ocasionado por las diferencias de las respuestas de los motores. Rápidamente el integrador actúa integrando la diferencia con el *set point* y corrigiendo el error cometido. Luego, en régimen, el ángulo en cuestión presenta un error típico de $\pm 2^\circ$, lo cual resulta aceptable.

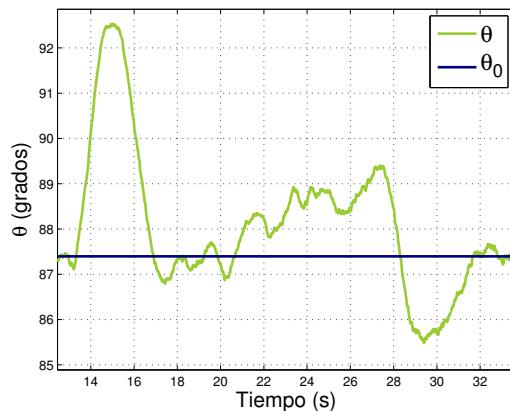


Figura 16.6.: Ángulo de Theta en lazo cerrado

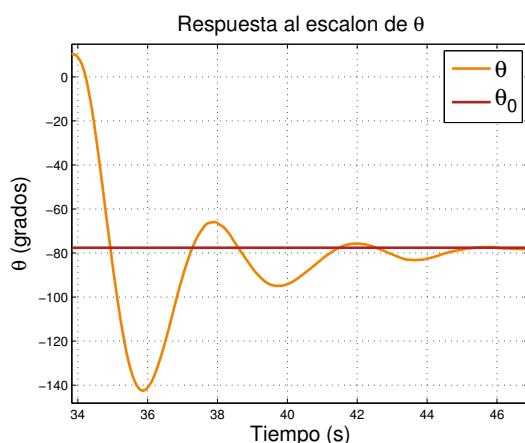


Figura 16.7.: Respuesta al escalón de Theta en lazo cerrado

un tiempo sensiblemente mayor en volver al equilibrio. A su vez, el sobretiro en θ no parece ocasionar problemas de vuelo considerables para la mayoría de las aplicaciones.

Por último en la figura 16.8 se evidencia la necesidad del término integral. Mientras que las diferencias entre los motores que ocasionan un aumento en el ángulo son corregidas rápidamente si se controla con el término integral, no ocurre lo mismo al utilizar un control solamente proporcional. Se puede observar que en este último caso el cuadricóptero llega a un equilibrio en θ distinto al *set point*. A su vez el movimiento al inicio de dicho ángulo

En la figura 16.7 se muestra la respuesta al escalón del ángulo en cuestión al apartarlo de su equilibrio aproximadamente 90° . Puede notarse claramente que el cuadricóptero vuelve a su equilibrio en forma satisfactoria en aproximadamente 6 segundos, tiempo que parece aceptable. Es importante destacar, de todas formas, que presenta un sobretiro considerable, alcanzando aproximadamente el 66 % del valor del escalón. Si bien es posible mejorar este aspecto, no resulta conveniente ya que en ese caso demora

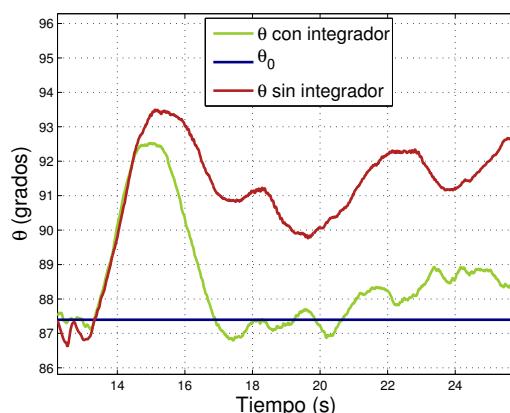


Figura 16.8.: Ángulo de Theta en lazo cerrado

16. Pruebas del controlador

es mayor al utilizar el control solamente proporcional.

Al igual que en el análisis de *Roll* y *Pitch*, se concluye que la matriz de realimentación utilizada es adecuada para controlar de buena forma al ángulo *Yaw*.

En la siguiente sección se presentarán los resultados del control completo del cuadricóptero en condiciones de vuelo, extendiendo la matriz a las variables de estado necesarias para lograr el control deseado.

16.3. Control del sistema completo

En las presentes pruebas el control realizado no incluye realimentación de la posición x e y ni de las velocidades v_{qx} y v_{qy} . Lo que se puede esperar con este controlador es que el sistema adquiera la orientación y la altura deseada, sin embargo es altamente probable que se produzca un desplazamiento horizontal. Al trabajar con el GPS se puede obtener una estimación más adecuada de la posición y de la velocidad horizontal y se la puede incluir en la realimentación, obteniendo así un mejor controlador.

En la figura 16.9 se observa la medida y la estimación de los tres ángulos de Euler a lo largo del tiempo de prueba en una situación de vuelo. Puede apreciarse claramente como el valor estimado de los ángulos de *Roll* y *Pitch* mantienen valores cercanos a cero. El ángulo de *Pitch* presenta en los primeros segundos de la prueba un comportamiento oscilatorio debido a la diferencia de empuje realizada por los motores de adelante y atrás. Cinco segundos después de que empieza a funcionar el controlador (comienza a funcionar en el segundo once) se compensa este desperfecto logrando el equilibrio en torno al cero.

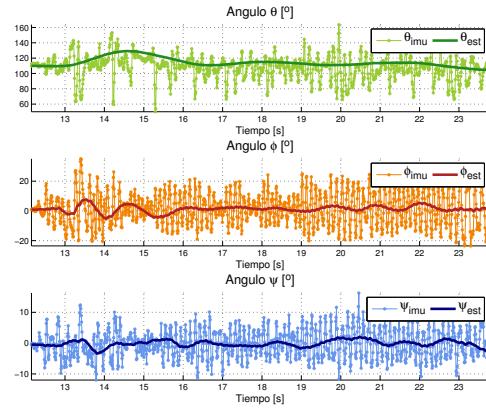


Figura 16.9.: Ángulos de Euler

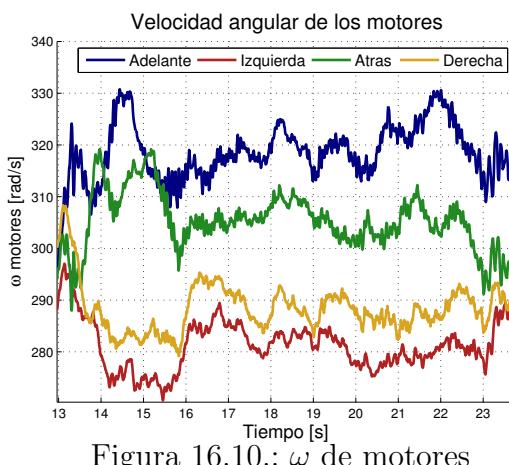


Figura 16.10.: ω de motores

En el caso del ángulo de *Yaw* se observa que el mismo aumenta cerca de 20° y luego se corrige, volviendo al valor de *set point*. Se observó que algunos motores no se encuentran perfectamente perpendiculares al plano horizontal del cuadricóptero y la contribución de cada uno de los motores es tal que a la velocidad angular de *set point* se produce un giro según k_q .

Este desperfecto se corrige con el integrador en el ángulo de Yaw. En la figura 16.10 se observa claramente el aumento de las velocidades angulares de los motores de adelante y atrás y la disminución en los motores laterales.

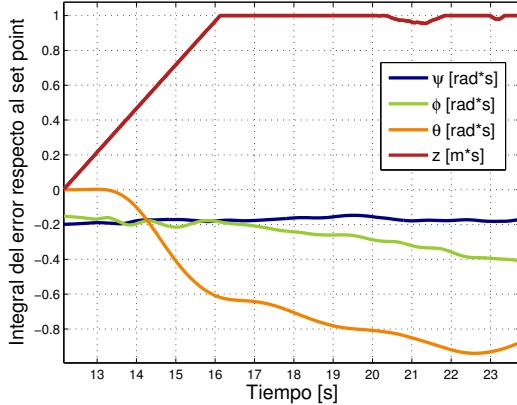


Figura 16.11.: Integrales

En la figura 16.11, se observa como la integral del error en el ángulo de Pitch (φ) disminuye, confirmando la observación anterior. A su vez en la figura 16.10 se observa que existe una diferencia en las velocidades angulares seteadas en los motores en cuestión. Se fija un valor mayor para el motor de adelante que para el de atrás a fin de compensar este efecto. Sucede algo similar aunque en menor medida para el ángulo de Roll (ψ) y los motores laterales.

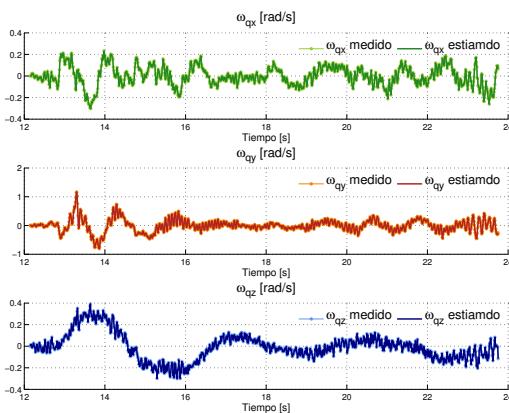


Figura 16.12.: Velocidades angulares del cuadricóptero

En la figura 16.11 se observa como el valor de la integral del error cometido en el ángulo de Yaw aumenta en módulo hasta que en alrededor del segundo 19 de la prueba comienza a estabilizarse. Este valor de la integral del error es el responsable del comportamiento descripto anteriormente.

El integrador permite también explicar la compensación de la diferencia de empuje entre los motores de adelante y atrás. Recordamos el comportamiento oscilatorio del ángulo de Pitch en los segundos iniciales de la prueba de vuelo, donde presentaba un offset positivo. La explicación de este fenómeno es que el empuje del motor de atrás es mayor que el de adelante. En la figura 16.11, se observa como la integral del error en el ángulo de Pitch (φ) disminuye, confirmando la observación anterior. A su vez en la figura 16.10 se observa que existe una diferencia en las velocidades angulares seteadas en los motores en cuestión. Se fija un valor mayor para el motor de adelante que para el de atrás a fin de compensar este efecto. Sucede algo similar aunque en menor medida para el ángulo de Roll (ψ) y los motores laterales.

En lo que respecta a las velocidades angulares del sistema puede observarse que en los tres casos las mismas se encuentran en valores cercanos al cero. En los primeros segundos se obtienen algunas variaciones fundamentalmente en ω_{qy} y ω_{qz} . Este resultado se concide con lo observado anteriormente. Incluso se observa que el andamiento de las curvas de velocidades angulares se corresponde con la derivada de los ángulos de Euler, por lo que estos resultados se adecuan perfectamente a lo esperado.

A partir de los resultados analizados hasta el momento se puede asegurar que el control realizado sobre el subsistema de los ángulos de Euler y las velocidades

16. Pruebas del controlador

angulares del cuadricóptero en situación de *hovering* es satisfactorio. Volvemos a hacer énfasis en la importancia de este subsistema ya que es aquel que garantiza la estabilidad del cuadricóptero. Errores en la posición pueden no ser tan importantes ya que si no se logra una posición determinada pero se alcanza una cercana, el resultado global puede ser satisfactorio en una amplia gama de aplicaciones. Este no es el caso del subsistema de los ángulos (fundamentalmente el de los ángulos de Roll y de Pitch). Errores en estos ángulos producen la deriva del sistema o incluso que el sistema no pueda mantenerse en vuelo.

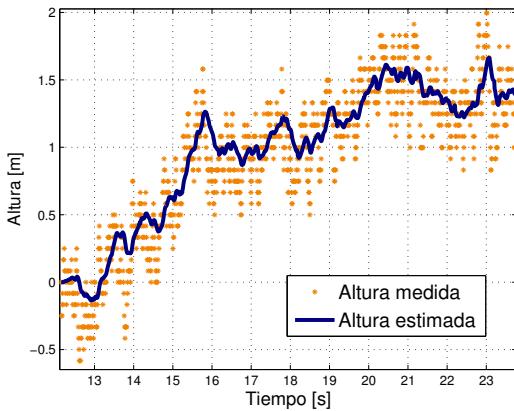


Figura 16.13.: Altura

dad angular mayor a la que se hubiera seteado sin agregar el integrador. Esto implica que sin el integrador no se hubiera podido alcanzar la altura objetivo, logrando un equilibrio a una altura inferior. Como se explicó en el capítulo 14, sin el integrador, variaciones en la masa o errores en la caracterización del empuje de los motores implican puntos de equilibrio distintos.

En esta sección se pudo verificar el correcto funcionamiento del control diseñado. En esta prueba, como fue aclarado previamente, no se realimenta la información de posiciones y velocidades horizontales. Los resultados obtenidos son satisfactorios permitiendo al sistema mantener el equilibrio en períodos donde la señal de GPS no se encuentra disponible o se encuentran disponibles muy pocos satélites conduciendo a medidas con errores superiores a lo deseado. Este controlador permite ignorar datos del GPS si estos están muy contaminados por ruído.

El control con señal de GPS no pudo ser verificado, pero es de esperar su correcto funcionamiento debido a que las simulaciones arrojan buenos resultados.

Por último se centrará la atención en analizar la altura. En la presente prueba se establece como valor objetivo de la altura 1,5m. En la figura 16.13 se observa que el valor objetivo es alcanzado en aproximadamente cinco segundos. Nuevamente, cabe destacar la importancia del integrador en la altura. En la figura 16.11 puede observarse la integral del error en la altura alcanza el valor máximo permitido. En esta situación se está seteando una veloci-

Parte VI.

Implementación

CAPÍTULO 17

MONTAJE

Al adquirir una plataforma comercial ya diseñada, no se dispone de espacio extra para instrumentarla. Se agregó una placa de sensores, una placa de desarrollo, una batería independiente para la electrónica, un GPS y un módulo Wi-Fi USB. Asimismo, para compatibilizar los diferentes voltajes que utilizaba cada placa, fue necesario hacer un circuito impreso que convirtiera los niveles lógicos de las señales.

A la hora de realizar el montaje de cada uno de los dispositivos mencionados, fue necesario realizar algunas consideraciones particulares. A continuación se presenta a grandes rasgos el trabajo de montaje realizado.

17.1. BeagleBoard

Se colocó la *BeagleBoard* junto con la batería de la electrónica (*Beagle-Juice*) en la parte inferior del cuadricóptero. Considerando que la *BeagleBoard* tiene una masa de 70g y la *Beagle-Juice* alrededor de 125g, la ubicación juega un rol fundamental en la posición del centro de masa del sistema global.

Se decidió ubicar las placas mencionadas en la parte inferior del cuadricóptero, implicando que el centro de masa del sistema se ubique algunos centímetros más abajo del centro geométrico, lo cual ayuda al sistema a lograr la estabilidad.

La *BeagleBoard* se monta sobre la *Beagle-Juice* mediante 4 tornillos de plástico con un tarugo (también de plástico) entre medio de ambas placas, de manera de mantenerlas sólidas y aisladas. En la imagen 17.1 se muestra la *BeagleBoard* con la batería en una caja de acrílico pensada para protegerlas ante eventuales golpes. En la misma figura

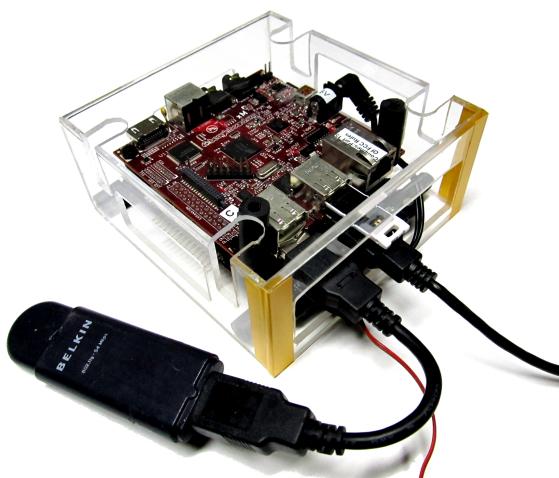


Figura 17.1.: Montaje del *BeagleBoard* Figura 17.1.: Montaje del *BeagleBoard* con la batería en una caja de acrílico pensada para protegerlas ante eventuales golpes. En la misma figura

17. Montaje

se puede observar además el módulo *Wi-Fi* que es ubicado lo más hacia el exterior posible, de modo de minimizar los obstáculos a la señal.

17.2. IMU

La ubicación y montaje de la placa de sensores juega un papel fundamental a la hora de obtener las medidas. Se deben tomar algunas consideraciones que juegan un rol fundamental para poder lograr el control del cuadricóptero.

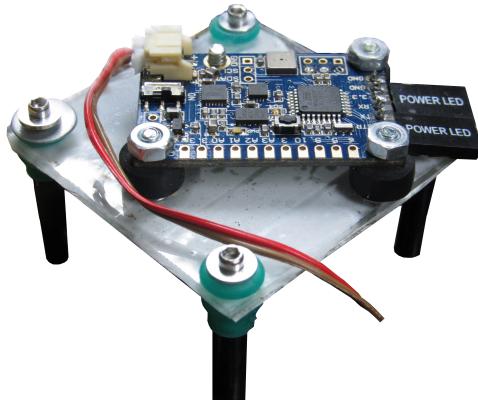


Figura 17.2.: Montaje de la IMU

sumamente importante para disminuir el ruido de medida. Al encender los motores el ruido mecánico ocasionado provoca un aumento en el ruido de los sensores, llegando a tomar valores 10 veces mayores que con los motores apagados. Por esta razón resulta vital la amortiguación en el montaje de la placa. Los amortiguadores utilizados constan de 1 sola pieza que pasa por un agujero en la placa, logrando que la misma no tenga contacto directo ni con la carrocería ni con el tornillo, dejándola aislada de los elementos que vibran solidarios los motores.

Otro aspecto fundamental para que las medidas de la IMU reflejen la realidad es la necesidad de ubicarla perfectamente horizontal. Para ello se utiliza el sistema mostrado en la figura 17.3, que aunque bastante precario, es efectivo. La idea es apoyar al cuadricóptero en los motores, simulando la fuerza que realizan en vuelo para compensar la fuerza del peso y lograr horizontalizar el plano de los brazos mediante la utilización de un nivel. Luego, ajustando o desajustando los tornillos se ubica la IMU de manera de obtener ángulos *Roll* y *Pitch* iguales a 0.

En primer lugar es importante que la placa se ubique en el centro del sistema. Más exactamente se deben ubicar a los acelerómetros de la placa lo más cerca del centro de masa posible, de modo de poder aproximar las aceleraciones medidas como las del centro de masa del cuadricóptero.

En la figura 17.2 se muestra el montaje de la IMU, donde se utilizaron cuatro apoyos con los que contaba la plataforma, ubicados en el centro de la misma. La IMU se fija a una placa de acrílico que a su vez está fijada a dichos apoyos mediante amortiguadores de doble fuelle. Esto último es



Figura 17.3.: Calibración de la vertical

17.3. GPS

El montaje del GPS se realiza de forma que tenga la mayor visibilidad de satélites posible. Como se mencionó en el capítulo 10, los rebotes en diferentes superficies provocan un deterioro considerable en la señal. Por estas razones se elige ubicarlo en la parte superior de la plataforma, como se muestra en la figura 17.4.



Figura 17.4.: Montaje del GPS

17.4. Conversor lógico

Como ya fue mencionado fue necesario la impresión de un circuito que convirtiera los voltajes entre las diferentes placas que componen el sistema. El mismo fue ubicado sobre 4 tuercas solidarias a la carrocería que se encontraban sin utilizar. La placa (mostrada en la figura 17.5), fue diseñada con el tamaño justo para atornillarse allí.

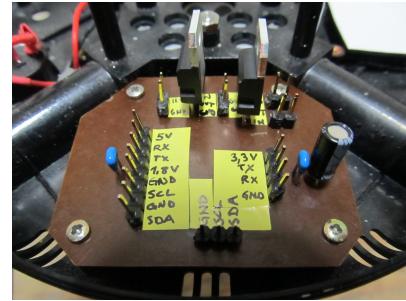


Figura 17.5.: Placa de conversores lógicos

17.5. Protección

Debido a la poca confianza que se tiene sobre los ESCs, se diseño una protección de aluminio para disminuir los daños causados por una eventual caída o golpe.



(a) Vista general

(b) Detalle de motor

Figura 17.6.: Protección

El material fue elegido teniendo como principales restricciones el peso, que disminuirá la carga útil del cuadricóptero, y su resistencia. El total de las protecciones aumenta en aproximadamente 250 gramos el peso del cuadricóptero. Las láminas de aluminio tienen forma de “L” para lograr mayor robustez. Las protecciones se pueden ver en la figura 17.6.

CAPÍTULO 18

SOFTWARE

El objetivo de esta sección es explicar la implementación en software del vuelo autónomo del cuadricóptero. Se incluyen algunos comentarios sobre su configuración y uso. Para entender en detalle qué hace cada función, referirse a los comentarios en el código fuente, disponible en el repositorio *Git* en la carpeta `src/`. Todas las referencias a archivos son relativas a la raíz del repositorio.

En el anexo F se explica como compilar y configurar las partes involucradas.

18.1. Esquema general

El código tiene una estructura modular, está escrito en C, y cada bloque está implementando como una biblioteca. Los programas están pensados para compilarse y ejecutarse en un entorno *Linux*. La estructura general del código se resume en la figura 18.1.

El software debe correr en tiempo real, por lo que resulta crítico evitar demoras durante la ejecución. Las operaciones de entrada/salida suelen ser un problema (en cualquier plataforma), por lo que en general se utilizan varios hilos, o varios programas intercomunicados, para evitar que la entrada/salida demore a partes del código que no dependen directamente de ella.

El software se distribuye entre varios microprocesadores en placas independientes:

- ***BeagleBoard*:**

- Se trabaja sobre un sistema operativo *Linux*, y se programa en C (se pueden utilizar otros lenguajes).

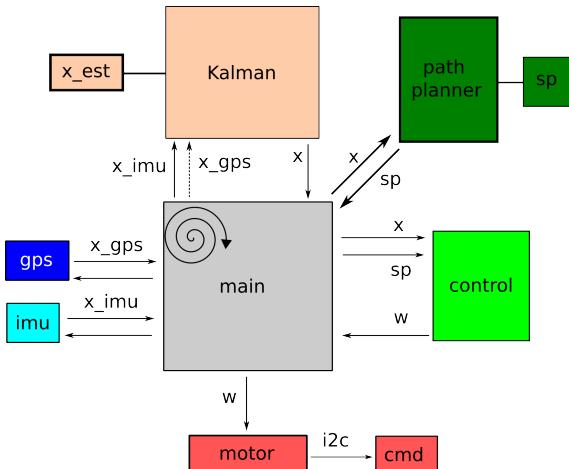


Figura 18.1.: Estructura general del código.
El software debe correr en tiempo real, por lo que resulta crítico evitar demoras durante la ejecución. Las operaciones de entrada/salida suelen ser un problema (en cualquier plataforma), por lo que en general se utilizan varios hilos, o varios programas intercomunicados, para evitar que la entrada/salida demore a partes del código que no dependen directamente de ella.

18. Software

- Centraliza toda la información, estima el vector de estados y ejecuta las acciones de control.
- Lee de 2 puertos serie, para obtener información de la IMU y del GPS.
- Se encarga de Lectura/escritura a una interfaz de comunicación *WiFi*.
- Lee/escribe a un puerto *I²C*, para comunicación con los *ESCs*.
- Se encarga de la escritura a memoria para almacenar datos relevantes generados durante la ejecución del programa principal. Se utiliza proceso independiente por cada tipo de datos que se desea guardar. Dicho proceso recibe los datos provenientes del programa principal y se encarga de escribirlos a memoria no volátil.
- Para evitar lidiar con múltiples hilos de ejecución, el programa encargado de mandar comandos a los motores corre por separado del programa principal. El programa principal se encarga de revisar los puertos de entrada/salida, y leer solamente cuando la lectura no requiere de un tiempo de espera. Esto se realiza mediante la función `select()`, un mecanismo que permite simular varios hilos de ejecución sin las complejidades de trabajar con multiples hilos.

■ *ESCs*:

- Son 4 microprocesadores conectados como esclavos a un bus *I²C* que comparten con el maestro. Según la configuración del switcheo, el maestro puede ser la *BeagleBoard* o la *CPU1*.
- Cada microprocesador se encarga de hacer girar uno de los motores a una velocidad fijada mediante *I²C*.
- El código que ejecutan no está disponible, solamente se conoce el protocolo para manejarlos mediante *I²C*.

■ *IMU*:

- Se programa en *Arduino*, un lenguaje muy similar a *C*, con algunas simplificaciones.
- Lee datos de los sensores y se comunica mediante un puerto serie.
- El código está fuertemente basado en el que venía con la IMU, fue modificado para adecuarlo a las necesidades del proyecto.

■ *GPS*:

- Cuenta con un microprocesador independiente, no se dispone del código que ejecuta.
- Envía datos a través de un puerto serie (*USB*).

En la sección 18.3 se describe el software que corre en la IMU. A continuación se describe el software que corre en la *BeagleBoard*, el cual incluye implementaciones del filtro de Kalman descrito en 12, el sistema de control descrito en 14, las calibraciones descritas en IV, el protocolo descrito en 5, y diversas funcionalidades auxiliares.

18.2. Software en la *BeagleBoard*

18.2.1. Requerimientos e inicialización del programa principal

Para correr el programa principal (de ahora en más: *main*) debería bastar con ejecutar el script `src/go.sh`. Durante la inicialización el *main* debe encontrar lo siguiente:

- `K*.txt`: Matrices de control utilizados en el modo *hover*.
- `lqr-*.txt`: Parámetros del algoritmo *LQR*.
- IMU: La IMU envía datos a través de una *UART* que es mapeada por el sistema operativo a un “archivo” `/dev/tty*`. El *main* recibe como parámetro la ruta a este archivo, o en su defecto un log `imu_raw.txt` generado por el *main* en una ejecución previa.
- `imu_calib.txt`: Parámetros de calibración, la biblioteca `imu_comm` los necesita para convertir los datos crudos provenientes de la IMU.
- GPS: Los datos provenientes del GPS (*USB*) son mapeados por el sistema operativo a `/dev/ttUSB*`. No se interactua directamente con este archivo, se utiliza la biblioteca `gps_comm` para iniciar un cliente que se comunique con el `gpsd`, que es el programa que se encarga de leer y analizar los datos crudos provenientes del GPS. El `gpsd` es iniciado por el script `go.sh`.

Si no se dispone de señal del GPS se puede configurar un modo de prueba en el que se simulan los datos del GPS (a 1Hz) generando ceros, o números al azar dentro de un rango dado. También es posible deshabilitar completamente el GPS y trabajar con un vector de estados reducido. Más adelante se explica como configurar los distintos modos.

- `cmd`: El driver de los motores, encargado exclusivamente de enviar continuamente comandos I^2C a los ESCs con la última velocidad configurada¹. La comunicación entre el driver y el *main* se realiza mediante la biblioteca `motor`, que a su vez se comunica con el driver mediante colas de kernel (IPC²), utilizando la biblioteca `uquad_kernel_msgq`.

Durante pruebas, se puede configurar el driver para que simule la presencia de los motores, o para que lea de la entrada estándar. Ver `src/i2c_beagle/README` por información sobre como compilar los distintos modos.

¹Los motores se apagan si no reciben comandos continuamente.

²Interprocess Communication: <http://www.cs.cf.ac.uk/Dave/C/>

18.2.2. Loop

En la figura 18.2 se muestra un diagrama del *main*, y en la 18.3 se muestra el loop que se ejecuta durante el funcionamiento normal (una vez terminada la inicialización).

A continuación se describen las etapas del loop, explicando brevemente las funcionalidades de cada una de las bibliotecas involucradas:

1. **imu:** La IMU genera datos nuevos cada 10ms. Al comienzo del loop, el *main* revisa si hay datos nuevos, y en caso afirmativo llama a la biblioteca para que los lea. Cuando se completa una trama, los datos crudos se almacenan en una cola circular mantenida por la biblioteca *imu_comm*. Al terminar de recibir una trama, se vuelve al principio del loop para verificar que no hay más nada para leer. En caso de haber más datos entonces hay que leerlos para evitar atrasarse respecto a la IMU, en caso contrario se convierten los datos y se avanza.
2. **gps:** El GPS genera datos nuevos a una tasa mucho menor que la IMU. Cada vez que se dispone de una muestra nueva en la IMU, el *main* revisa si también hay un dato nuevo del GPS. Avanza aunque no se disponga de datos nuevos del GPS.
3. **kalman:** El filtro de Kalman está implementado en la biblioteca *kalman*. Recibe una estructura de datos generada por *imu_comm* y otra (opcional) generada por *gps_comm*. Mantiene una estructura de datos que almacena el estado estimado y las matrices de covarianza.
4. **path planner:** El módulo generador de rutas está implementado en la biblioteca *path_planner*. Compara el estado actual con el objetivo, y determina si se completó el objetivo actual³. En caso afirmativo, devuelve una bandera que le indicará al módulo de control que debe actualizar la matriz de control para ajustarla a la nueva trayectoria.
5. **control:** El módulo de control está implementado en la biblioteca *control*. Mantiene una estructura con las matrices del control proporcional e integral (si corresponde). Recibe como argumento el estado estimado (de la biblioteca *kalman*), la velocidad actual de los motores (Mantenida por la biblioteca *motor*), y una estructura generada por *path_planner* que indica el estado objetivo y la trayectoria a seguir. Devuelve la acción de control (velocidades angulares) a aplicar sobre los motores.
6. **motor:** Recibe las velocidades angulares determinadas por el módulo *control*, las convierte a I^2C y envía el resultado al driver, quien se encarga de establecer la nueva velocidad deseada.

Por información relativa a bloques, configuración, compilación, ejecución, etc, referirse a `src/README`.

³Solamente se implementó el modo *hover*.

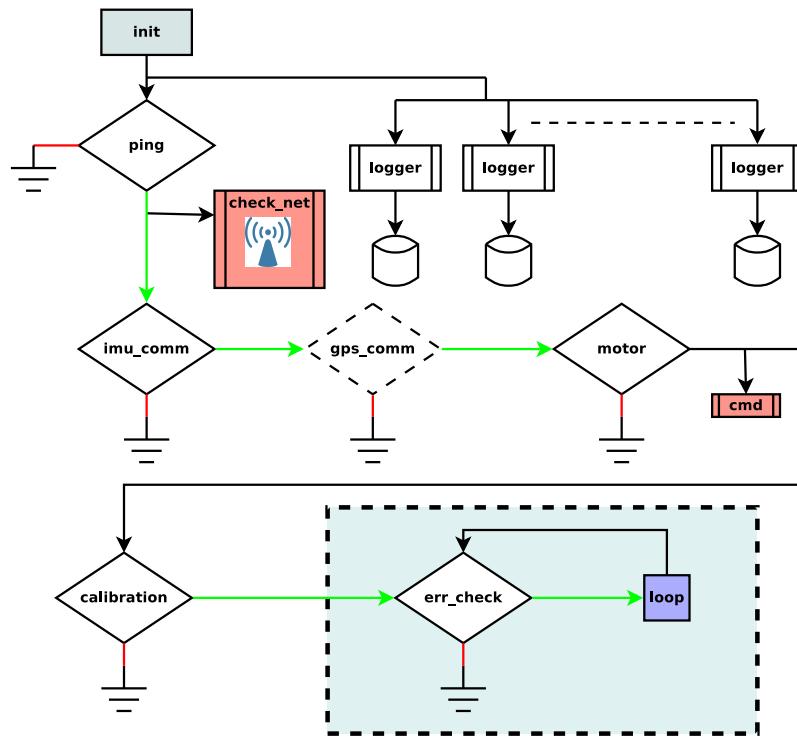


Figura 18.2.: Diagrama de flujo del programa principal.

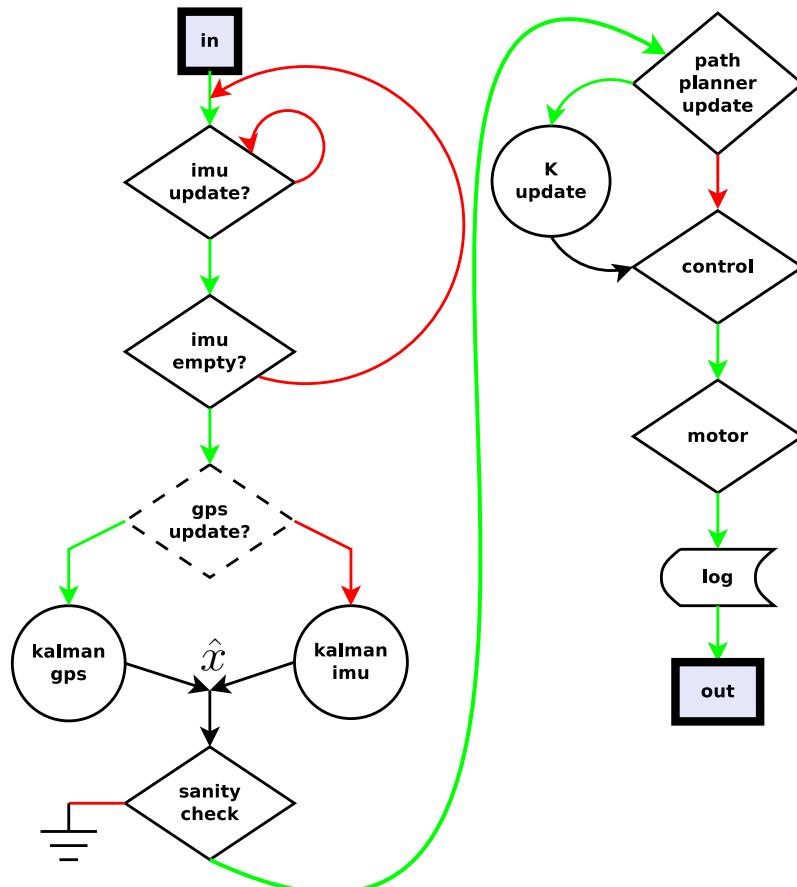


Figura 18.3.: Loop principal.

18.2.3. Módulo *imu.comm*

A continuación se describen algunas de las funcionalidades a destacar de la biblioteca *imu.comm*.

- **Calibración:** Se acumulan un conjunto de muestras que después se utilizan para estimar el offset de los giróscopos, la altura inicial, y pueden ser utilizados para inicializar el filtro de *Kalman*. Durante la calibración es crítico que el cuadricóptero no se mueva, ya que en caso de hacerlo el offset de los giróscopos será mal estimado.

La inclinación durante la calibración afecta la estimación inicial del offset en los acelerómetros, pero en caso de no estar perfectamente horizontal se acomodará luego de unos segundos, no es algo crítico.

- **Conversión:** Cargando parámetros de calibración, es posible convertir datos crudos provenientes de los sensores (cuentas de un ADC) a datos útiles:

- Acelerómetro → Aceleraciones.
- Giróscopo → Velocidad angulares.
- Acelerómetro + Magnetómetro → Ángulos de Euler⁴.
- Barómetro → Altura y temperatura.

Para la conversión se utilizaron las calibraciones obtenida de la parte IV.

- Magnetómetro:

$$\text{conv} = T.K_{inv}.(\text{crudo} - b) \quad (18.1)$$

donde

- T corrige el problema del *cross axis sensitivity*.
- K_{inv} es la inversa de la matriz de ganancias.
- b Es un offset.
- Acelerómetro: Además de una calibración como la del magnetómetro, se implementó una compensación por temperatura:

$$\text{conv} = T.\text{inv}(K).(\text{crudo} - b + b_t.(t - t_0)) \quad (18.2)$$

donde

- T , K y b cumplen el mismo rol que en el magnetómetro.
- t es la temperatura actual, y t_0 la temperatura a la que se realizó la calibración de donde surgieron T , K y b .
- b_t es el factor que permite la compensación por temperatura.
- Giróscopo: Se implementó algo análogo a lo que se hizo para el acelerómetro, solo que al final se le resta un offset que se determina durante la calibración al inicio del programa. Este offset es sencillo de estimar, basta con que el cuadricóptero no se mueva durante la calibración.
Implementar algo análogo pero para el caso del acelerómetro requeriría

⁴La estructura de datos convertidos **no** incluye información sobre el campo magnético, incluye solamente los ángulos de Euler obtenidos a partir del acelerómetros y del magnetómetro. De ser necesaria la información de campo magnético, es sencilla de agregar.

que el cuadricóptero estuviese perfectamente horizontal durante la calibración, y para le magnetómetro que estuviese mirando hacia el norte. Estos requisitos son más complicados de cumplir (requerirían de una plataforma de despegue) por lo que no se implementó.

- **Verificación:** Se llevan 2 banderas que indican si la norma del vector de aceleración y la del vector de campo magnético caen dentro del rango esperado. Esto puede ser de utilidad en el filtro de Kalman.
- **Filtrado:** Se disponen de funciones que permiten obtener el elemento más nuevo que aún no ha sido utilizado, o el resultado de aplicar un filtro FIR⁵ a los n elementos más recientes de la cola. Si por problemas de tiempo el *main* se retrasa, pueden haber datos que nunca sean etiquetados como “el dato más nuevo”, ya que se leerá hasta ponerse al dia. De cualquier forma, serán tomados en cuenta en el filtro.

El filtro implementado trabaja sobre los datos crudos convertidos a punto flotante:

$$\begin{aligned} y[n] = & 0,2x[n] + 0,2x[n - 1] + 0,2x[n - 2] + 0,2x[n - 3] \\ & + 0,1x[n - 4] + 0,1x[n - 5] \end{aligned} \quad (18.3)$$

El objetivo del filtro es reducir el ruido en los datos crudos. En la figura 18.4 se observa los datos crudos provenientes del giróscopo, junto con los datos filtrados.

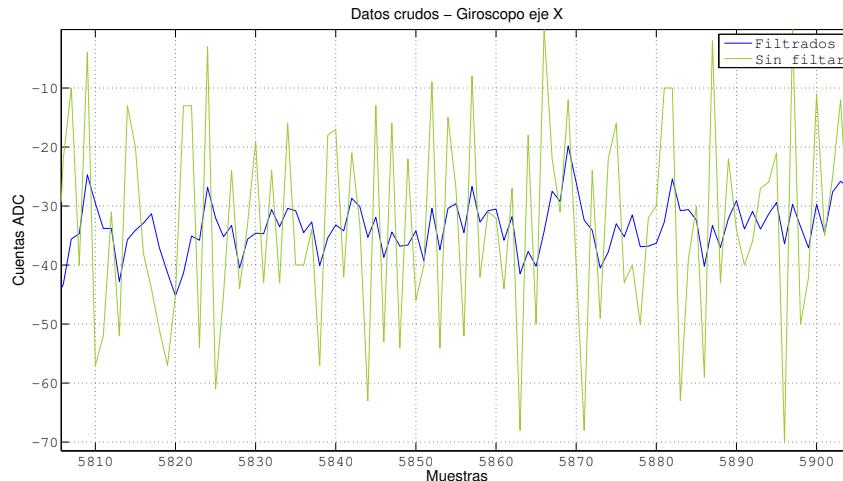


Figura 18.4.: Comparación del ruido presente en los datos crudos provenientes del giróscopo eje x con el cuadricóptero quieto y el resultante luego de aplicar el filtro de la ecuación 18.3.

Configurando adecuadamente los coeficientes del filtro y el largo del mismo se puede hacer que trabaje como un promedio, que no haga nada, etc.

⁵Los coeficientes del filtro están definidos en *imu.comm.init()*.

NOTA: El filtro fue desactivado, ya que introduce un retardo que deteriora la performance del sistema, y en realidad el filtro FIR es redundante con el filtro de Kalman.

- **Modo FAKE:** Seteando `IMU_COMM_FAKE` a 1, la biblioteca leerá de un log `ascii` en lugar de utilizar el puerto serie. En el modo *FAKE* los tiempos no son un problema crítico, ya que no correrá en tiempo real.
- **Ángulos de Euler:** Para calcular los ángulos de Euler (en radianes) a partir del vector de aceleración y del campo magnético se tomó como referencia [12]. Se utilizan las siguientes ecuaciones:

$$\phi_k = \begin{cases} -\arcsin\left(\frac{Acc_x}{||Acc||}\right) & |Acc_x| < 0,99||\vec{Acc}|| \\ -\frac{\pi}{2}sign(Acc_x) & |Acc_x| \geq 0,99||\vec{Acc}|| \end{cases}$$

$$\psi_k = \begin{cases} -\arctan2\left(\frac{Acc_y}{Acc_z}\right) & |Acc_x| < 0,99||\vec{Acc}|| \\ 0 & |Acc_x| \geq 0,99||\vec{Acc}|| \end{cases}$$

$$M = \begin{pmatrix} \cos(\phi) & \sin(\phi) \sin(\psi) & \cos(\psi) \sin(\phi) \\ 0 & \cos(\psi) & -\sin(\psi) \\ -\sin(\phi) & \cos(\phi) \sin(\psi) & \cos(\phi) \cos(\psi) \end{pmatrix}$$

$$v = M \cdot \vec{Magn}$$

$$\theta_k = -\arctan2\left(\frac{v[1]}{v[0]}\right) + 0,17$$

El offset en θ_k es debido a que el campo magnético en Uruguay no apunta directamente hacia el norte⁶.

18.2.4. Módulo *kalman*

A parte de implementar el filtro de Kalman descrito en la sección 12, la biblioteca *kalman* se encarga de:

- Suavizar la estimación del ángulo *theta* dada por los acelerómetros y los magnetómetros antes de introducirlo en el filtro de Kalman. El ruido presente en los acelerómetros, sumado a la mala performance del magnetómetro en lugares cerrados⁷, hace que sea necesario utilizar una lógica de suavizado más inteligente que un simple filtrado para combatir los saltos en la discontinuidad en la función *atan2()*, utilizada para calcular θ . La corrección se aplica al dato convertido previo a su utilización en el filtro de Kalman. La implementación es la siguiente ecuación:

$$\theta_k = \begin{cases} \theta_k & |\theta_k - \hat{\theta}_{k-1}| < \pi \\ \theta_k - 2\pi fix\left(\frac{\theta_k - \hat{\theta}_{k-1} + \pi sign(\theta_k - \hat{\theta}_{k-1})}{2\pi}\right) & |\theta_k - \hat{\theta}_{k-1}| \geq \pi \end{cases}$$

⁶Fuente: <http://www.ngdc.noaa.gov/geomagmodels/Declination.jsp>

⁷Se probó en lugares con muchos materiales metálicos, distorsionan las lecturas del magnetómetro.

donde θ_k es el dato nuevo proveniente de la IMU, $\hat{\theta}_{k-1}$ es la estimación actual, generada por el filtro de Kalman en el paso previo.

Para ψ y ϕ no es implementó una corrección de este tipo, ya que el cuadricóptero no debe inclinarse 90° , y el ruido en los datos no es suficiente para que durante el funcionamiento normal la medida llegue a valores cercanos a la discontinuidad de las funciones utilizadas ($\text{asin}()$ y $\text{atan2}()$).

- Llevar una estimación del *bias* en los acelerómetros, cuyo objetivo es corregir errores sistemáticos.
- Existe la posibilidad de modificar la matriz de ruidos de observación, para las estimaciones realizadas a partir de información proveniente del acelerómetro y del magnetómetro, en función de la norma del vector medido por cada sensor⁸.

Por detalles referirse a `src/kalman/uquad_kalman.c`.

18.2.5. Módulo de *control*

La implementación del módulo de control se hizo en la biblioteca *control*. Las funcionalidades implementadas son las siguientes:

- Control proporcional e integral.
- Linealización del sistema en torno a una trayectoria y un *set point* dados, y cálculo de las matrices de control correspondientes mediante *LQR*.

Como se mencionó en 18.2.4, el vector de estados almacenado por el filtro de Kalman incluye, además de las variables de estado del sistema, tres variables para la estimación del *bias* de los acelerómetros. Estos tres términos no se utilizan en la módulo de control.

Control proporcional

El control proporcional se rige por la siguiente ecuación:

$$\vec{\omega}_{prop} = K_{prop}(\vec{s}\vec{p}_x - \vec{x}_{est}) \quad (18.4)$$

donde

- $\vec{s}\vec{p}_x$ Es el estado deseado, dado por el módulo *path_planner*.
- \vec{x}_{est} Es la estimación del estado del sistema en el momento actual, dada por el filtro de Kalman.

⁸Esta idea se tomó de <http://www.vectornav.com>.

Control integral

El control integral es más complejo, ya que incluye restricciones que son necesarias en la práctica. A continuación se presenta un pseudocódigo de la función que implementa la integral. Se aplica de manera independiente a cada una de las variables sobre la cual se desea llevar un control integral:

$$err = \max(\min(sp_{x_i} - \hat{x}_i.T_s, th_{max}), -th_{max})$$

$$int_k = \begin{cases} int_{k-1} & |err| > th_{dist} \\ \max(\min(err + int_{k-1}, th_{accum}), -th_{accum}) & |err| \leq th_{dist} \end{cases}$$

donde:

- sp_i y \hat{x}_i : Setpoint y estado estimado para el i -ésimo elemento del vector de estados (el cual se desea control usando un control integral).
- err : Diferencia entre el estado deseado y el actual: $err = sp_x - x$.
- int_{k-1} : Valor acumulado hasta el instante previo.
- int_k : Nuevo valor para la integral.
- T_s : Período de muestreo.
- Umbrales (definidos, para cada una de las variables a integrar, en `src/control/control.h`) para los 3 controles implementados:
 1. No se integra si la diferencia entre el estado actual y el deseado es mayor a un umbral dado por th_{dist} , ya que se asume que esa situación debe ser resuelta por el control proporcional.
 2. Para evitar que la integral crezca muy rápido se utiliza un umbral th_{max} , el máximo error que se acepta integrar está acotado por $th_{max}.T_s$.
 3. Por último, si por algún motivo la integral acumula demasiado el sistema tardará mucho en recuperarse, por lo que se satura el integrador en th_{accum} .

Una vez calculada la integral, la ecuación que genera la acción de control integral es:

$$\vec{\omega}_{int} = K_{int}x_{int} \quad (18.5)$$

Control total

La acción de control viene dada por

$$\vec{\omega} = \vec{sp}_\omega + \vec{\omega}_{prop} + \vec{\omega}_{int} \quad (18.6)$$

donde \vec{sp}_ω es velocidad angular que se desea setearle a los motores, definida por el módulo `path_planner`.

Las matrices de control para el modo *hover* se encuentran en:

- Proporcional (K_{prop}): `src/control/K_prop_pptz.txt`
- Integral (K_{int}): `src/control/K_int_full_pptz.txt`.

Estas matrices son utilizadas en el modo *hover*, donde no hace falta utilizar *LQR*. Son archivos de texto plano, y si se modifican entonces los cambios serán tomados en cuenta al ejecutar el script `src/go.sh`.

18.2.6. Generador de rutas

La versión actual del código implementa solamente el modo *hover*. El *set point* inicial cero para todas las variables, excepto para el ángulo θ , para el cual se tomará el ángulo inicial, y para la altura (1m por defecto).

Se pueden modificar las condiciones de *hovering* en `src/main/main.c`.

En *MatLab* hay una implementación del generador de rutas, queda pendiente pasarlo a C.

Modo Manual

En el modo *hover* es posible modificar el *set point* desde la línea de manera remota. Para ellos, iniciar el *main* y apretar la tecla `m`, seguida de un `ENTER`. Esto seteará el *main* en modo manual, y estará dispuesto a recibir comandos. Cada comando modificará el *set point*, y será considerado solamente luego de presionar `ENTER`. La lista de comandos se encuentra en `src/common/manual_mode.h`.

18.2.7. Driver de los motores y módulo *motor*

La biblioteca *motor* y el driver `src/i2c_beagle/cmd_motores.c` (de ahora en más *cmd*) tienen una fuerte relación, y deben ser coherentes. El driver no se pudo incluir como una biblioteca más, ya que requiere de un encabezado que solamente está disponible en la *BeagleBoard*.

Algunas consideraciones relevantes:

- El *cmd* espera una velocidad superior a cierto mínimo, de lo contrario no arrancará los motores. Este umbral debe estar apareado, de lo contrario *motor* será incapaz de arrancar los motores en el momento apropiado.
Luego del arranque, *motor* se encargará de no enviar valores por debajo de los valores definidos como mínimo y máximo. Usar valores por debajo del mínimo puede hacer que se apaguen los motores, y valores por encima del máximo pueden sobrecalentar los contactos de los cables que alimentan a los motores. El máximo también debe estar apareado entre el *cmd* y *motor*.
El *cmd* reportará un error en caso de recibir valores fuera de rango.
- Al arrancar los motores, el *cmd* setea velocidades en torno una rampa⁹ desde 0 hasta el valor definido como mínimo, al cual el cuadricóptero no es capaz de levantar vuelo.
- Por cada comando que *motor* envía al *cmd*, este último responde con un *ack*. Así *motor* verifica que el *cmd* está funcionando¹⁰.

ADVERTENCIA: Cualquier mensaje de error reportado por el *cmd* es motivo suficiente para detener el vuelo y analizar el problema.

⁹La implementación son valores que saltan por encima y por debajo de la rampa, esta técnica ha demostrado ser eficiente para hacer arrancar los motores.

¹⁰Solamente se verifica que hay comunicación, pero la implementación es tal que si la comunicación es exitosa, entonces todo debería estar funcionando correctamente.

18.2.8. Tiempos

El período de muestreo resulta fundamental para tanto el filtro de Kalman como el control integral. Para llevar el tiempo se dispone de funciones del sistema operativo que tienen precisión de microsegundos. Se consulta el tiempo al momento de llamar a las bibliotecas *kalman* y *control* y se lo almacena, de manera de poder estimar un período de muestreo a partir del tiempo transcurrido entre llamadas sucesivas.

El máximo retardo entre que se lee un dato nuevo de la IMU y que se efectua una acción de control es de 10ms, en general es de 8ms. Retardos mayores llevarían a perder muestras de la IMU, lo cual sería detectable en el log de errores. Por más información sobre los logs referirse al anexo F.

18.2.9. Comunicación

La comunicación con la *BeagleBoard* se hace mediante *ssh*. En el anexo F se explica como configurar las partes involucradas.

18.2.10. Configuración

Parametros del sistema

- **`src/common/uquad_types.h`**: La mayor parte de los parámetros del sistema están definidos aqui, entre ellos la masa, el tensor de inercia, el orden del vector de estados, período de muestreo, etc.
- **`src/CMakeLists.txt`**: La variable `USE_GPS` es un booleano que determina si ha de utilizarse el GPS o no.
- **`src/common/uquad_config.h`**: En este archivo se configura el modo de funcionamiento. Se realizan controles sobre la opciones elegidas, evitando que el usuario seleccione un modo inválido. Es posible elegir:
 - Trabajar con 8 estados o con 12.
 - Guardar logs.
 - Habilitar control integral.
 - Si el GPS fue habilitado, es posible utilizar un GPS de mentira, útil para pruebas.
 - Cada cuantas muestras de la IMU se desea efectuar una acción de control.
- **`src imu imu_comm.h`**: Permite:
 - Permite configurar el largo del filtro LPF, cuyos coeficientes se definen en `src imu imu_comm.c`.
 - Elegir entre leer de un log o de la IMU.
 - Elegir el largo de la cola circular donde se almacenan los datos crudos (debe ser mayor que la cantidad de coeficientes del filtro).
 - Seleccionar cuantas muestras han de usarse para la calibración (512 por defecto).

Control

Para el modo *hover*, basta con modificar las matrices en `src/control/`, de donde se leen las ganancias a utilizar. Para el resto de los modos hay que modificar los archivos de donde se configura el *LQR*. En la sección 18.2.10 se explicó como modificar la estrategia de control.

Ruidos Kalman

En `src/kalman/uquad_kalman.c` se definen los ruidos de transición de estados y de observación. Los valores que varían al utilizar el modo de covarianza dinámica (en función de la norma de los vectores de aceleración y campo magnético) se definen en `src/kalman/uquad_kalman.h`.

18.3. Software en la IMU

La IMU cuenta con un microprocesador *ATmega328p*. Queda disponible poder computacional como para agregar más funcionalidades. Las tareas que se le asignaron al programa que ejecuta la IMU son:

- Leer de los sensores: Acelerómetro, Giróscopo, Magnetómetro y Barómetro.
- Armar una trama de datos, que puede ser en ASCII o binario, y enviarlos mediante una *UART*.
- Enviar una trama cada 10ms en la cual se incluyen:
 - Un carácter ASCII de inicio de trama: A.
 - El tiempo (en microsegundos) que tardó el loop de lectura.
 - El promedio de dos muestras del:
 - Acelerómetro (*x,y,z*), en cuentas del ADC.
 - Giróscopo (*x,y,z*), en cuentas del ADC.
 - Magnetómetro (*x,y,z*), en cuentas del ADC.
 - Un dato nuevo del barómetro:
 - Temperatura (en C° multiplicados por un factor de 10).
 - Presión (en Pascales).
 - Un carácter ASCII de fin de trama: Z.

Si la IMU trabaja en modo ASCII, una trama tiene el siguiente aspecto:

```
A 11968 -8 -4 211 11 42 6 -14 -3 154 160 102025 Z
```

Los elementos se separan mediante un carácter `\t`, y la línea se termina con un `\n`. En modo binario el formato es el mismo, solo que sin caracteres de separación ni de fin de línea.

18.3.1. Cambios al software original

El código original que trae la IMU fue modificado. El problema principal que tenía era que la frecuencia de muestreo no era estable. Los cambios más relevantes fueron:

- Transmisión de datos en segundo plano, mediante interrupciones, evitando demorar el loop principal al momento de transmitir.
- Posibilidad transmitir en binario (se mantuvo el modo ASCII, pero no es posible transmitir a 10ms en dicho modo).
- Se modificó el formato de los datos enviados al puerto serie.
- El barómetro demora varios milisegundos entre que se le pide un dato y que lo tiene disponible. El código original esperaba durante este tiempo. La versión modificada sigue trabajando y vuelve para recoger el dato luego que transcurrió el tiempo en cuestión. Se implementó una máquina de estados para el manejo del barómetro. No se implementó mediante interrupciones, ya que las transacciones a través del puerto I^2C en general son secuencias del tipo:
 1. Indicar de qué dirección se quiere leer.
 2. Seleccionar el registro del cual se desea leer.
 3. Recibir los datos.

No se debe usar el puerto I^2C durante una secuencia de este tipo.

18.3.2. Diagrama de flujo

En la figura 18.5 se muestra un diagrama del programa que ejecuta la IMU.

- Las operaciones en círculos rojos requieren de acceso al puerto I^2C .
- Las flechas verdes indican que el resultado de evaluar la condición fue positivo, las rojas indican que la condición no se cumplió.

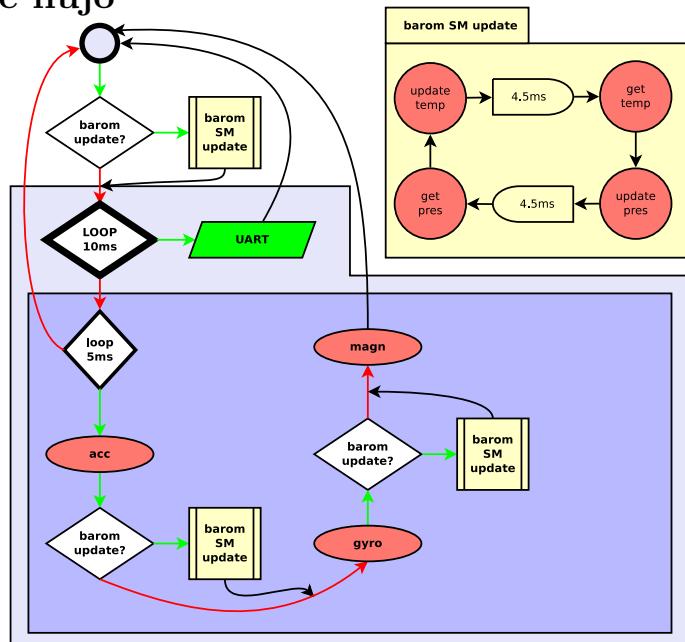


Figura 18.5.: Diagrama de flujo del programa que ejecuta la IMU.

- La condición *barom update?* se verifica cuando se cumplió alguno de los retardos de 4.5ms requeridos por el barómetro, en este caso se debe avanzar la máquina de estados.

CAPÍTULO 19

ELECTRÓNICA

Desde el momento que se elige realizar un trabajo de integración se corre el riesgo de trabajar con componentes que trabajan a distintos niveles de tensión. En este caso, los niveles de tensión presentes son:

- 11.1 V: motores
- 5 V: alimentación de la BeagleBoard
- 3.3 V: electrónica de los ESCs
- 1.8 V: líneas de *rx*, *tx* y bus *i²c* de la BeagleBoard
- La IMU puede alimentarse con cualquier voltaje menor a 16 V

Para poder integrar correctamente los diferentes componentes es necesario diseñar una placa de conversores lógicos para que lograr una correcta comunicación entre todas las partes del sistema.

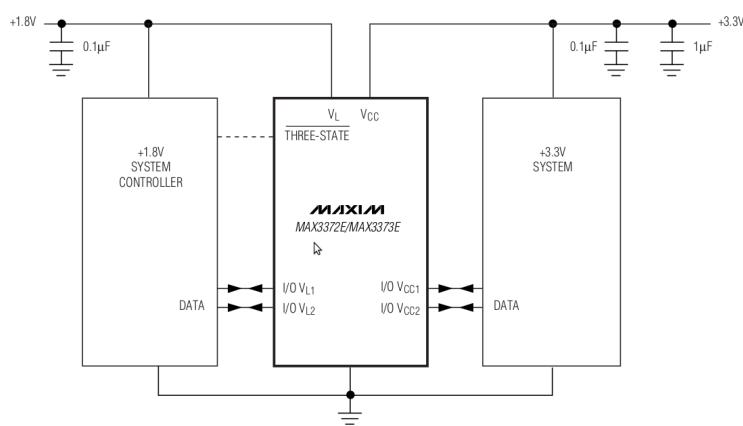


Figura 19.1.: Diagrama de conexionado del MAX3372E

Se elige utilizar como voltaje de entrada para la IMU el mismo voltaje que la Beagleboard, 5V. Concretamente se debe realizar la conversión de niveles entre las líneas de bus *i²c* de los ESCs, que manejan 3.3V, y las líneas del bus de la Beagleboard, que manejan 1.8V y entre las líneas *rx* y *tx* de la IMU, que manejan 3,3V y las mismas líneas de la Beagleboard, que manejan 1.8V. Para realizar la conversión se

19. Electrónica

utilizará el chip **MAX3372E**. El diagrama de conexionado se muestra en la figura 19.1, tomada de la hoja de datos del dispositivo mencionado.

Es importante destacar la necesidad de la bidireccionalidad de las líneas del bus i^2c ya que tanto el amo como los esclavos deben ser capaces de realizar cambios sobre la misma. Como se puede ver en la figura antes mencionada, el chip posee 2 líneas bidireccionales que serán utilizadas para las líneas *SDA* y *SCL* del bus i^2c . Para las líneas de transmisión serial es suficiente con un par de líneas unidireccionales, pero de todas formas se decide utilizar el mismo chip por razones de simplicidad.

La placa de conversión de niveles lógicos será alimentada con las líneas de 5V y 1.8V presentes en la cabecera de expansión de la Beagleboard. Para generar los 3.3V necesarios para alimentar los 2 chips MAX3372E se utiliza el regulador de tensión **LD33V**. A su vez la placa otorgará la posibilidad de alimentarla con la misma batería que utilizan los motores, es decir, alimentarla con 11.1V. En este caso se utiliza el regulador de tensión **L7805CV** para generar una línea de 5V a partir de la entrada de 11.1V, que será utilizada para alimentar la Beagleboard. De esta manera se logra eliminar una de las baterías del sistema, la BeagleJuice, logrando reducir aproximadamente 130g la masa del sistema.

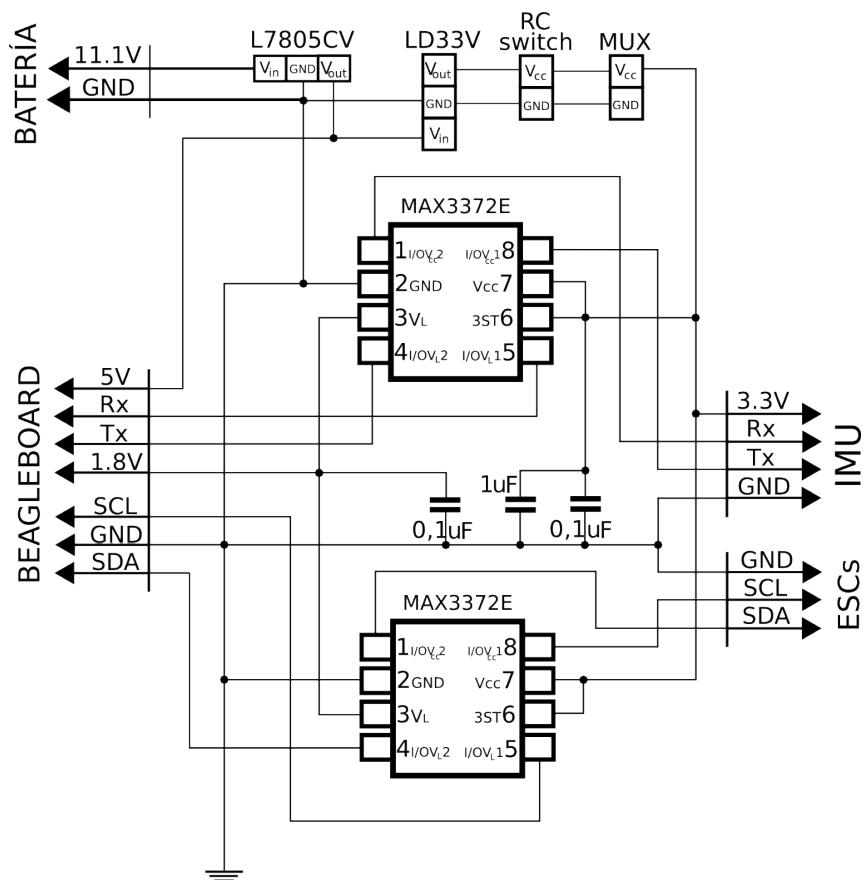


Figura 19.2.: Placa conversores lógicos y switcheo

Por último se incluye en la misma un par de pines necesarios para alimentar el *switch RC* y el *multiplexor* necesarios para realizar la comutación entre el mando automático y el manual, explicado en el capítulo 20.

El diseño final de la placa es el mostrado en la figura 19.2.

CAPÍTULO 20

SWITCHEO MANUAL/AUTOMÁTICO

Se pretende lograr que el sistema implementado pueda ser comandado de forma tanto manual como automática, buscándose además que la transición entre las dos formas de comando pueda realizarse en forma remota.

Para ello es necesario encontrar una forma de indicar al sistema que tipo de comando se desea utilizar. Se cuenta con un control remoto diseñado para comandar una gran variedad de vehículos radiocontrolados. Por este motivo, el control envía y recibe señales que no son utilizadas por el cuadricóptero. Se opta por reutilizar alguna de dichas señales para lograr el switcheo manual/automático.

20.1. Señales del control remoto

El sistema de transmisor/receptor que utiliza el control remoto (Walkera WK-2801 PRO) transmite la información a través de 8 canales mediante modulación PPM (pulse position modulation).

En dicho esquema de modulación se parte de un frame temporal de duración fija que es dividido en N partes iguales. Dichas partes son luego llevadas a 0 o Vcc para codificar así la información.

20.2. Señal elegida

Luego de un estudio detallado del transmisor y el receptor, pudo determinarse que la señal que se envía a través del canal etiquetado como GEAR no es utilizada en el comando del cuadricóptero. De igual manera, dicha señal es afectada únicamente por una llave existente en el control remoto. Dicha llave sólamente afecta a la señal presente en el canal GEAR, dejando invariantes el resto de las señales enviadas. La señal GEAR entonces verifica:

- Es comandada por un único elemento del control remoto
- El elemento que la comanda no afecta ninguna otra señal

20. Switcheo Manual/Automático

- El control de cuadricóptero no utiliza dicha señal

Cada canal es transmitido a través de 3 conductores, donde dos de ellos proveen voltajes de referencia Vcc (4.8V) y tierra (0V), mientras que el tercero es usado para transmitir la señal PPM.

Los conductores pueden ser identificados por su color según la siguiente tabla:

Color	Señal	Tensión
Negro	Tierra	0V
Naranja	Vcc	4.8V
Blanco	Datos(PPM)	0V-4.8V

En cuanto al control remoto, la señal del canal GEAR puede ser modificada mediante la llave indicada como GEAR SW en la figura 20.1

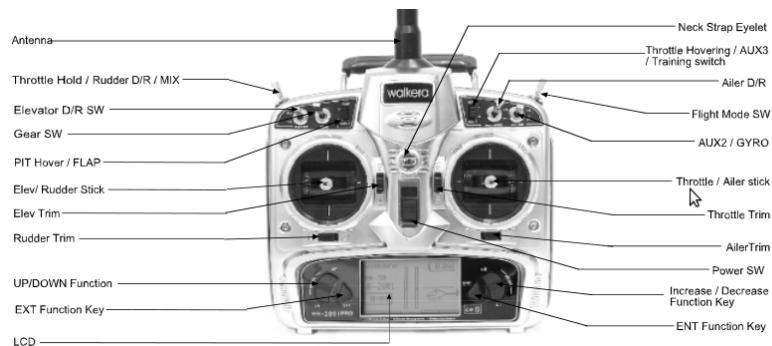


Figura 20.1.: Control remoto

20.3. Análisis de la señal

Se procedió a estudiar la señal seleccionada mediante la adquisición de la misma con un osciloscópico. Pudo verse que cuando la llave GEAR SW está “abajo” la señal observada es una onda cuadrada (Tierra-Vcc) de frecuencia $f = 52,41Hz$ y ciclo de trabajo 5.25 %, mientras que al estar la llave “arriba” el ciclo de trabajo varía, siendo este ahora 9.50 %.

Es claro que lo que verdaderamente está sucediendo es que la posición de la llave baja se codifica seteando un cierto número de frames temporales consecutivos a Vcc y dejando el resto de los frames a tierra, mientras que la posición alta se codifica seteando un mayor número de frames consecutivos a Vcc y dejando el resto a tierra.

Lo expuesto anteriormente puede observarse en las figuras 20.2a y 20.2b.

20.4. Detección de la posición de la llave

Para poder implementar el switcheo manual/automático será necesario poder detectar la posición de la llave. Esto es, debemos convertir la señal modulada proveniente del control en una señal binaria ON/OFF. Existen soluciones ya implementadas que se comercializan a muy bajo precio. Se opta por adquirir el circuito

20.4. Detección de la posición de la llave

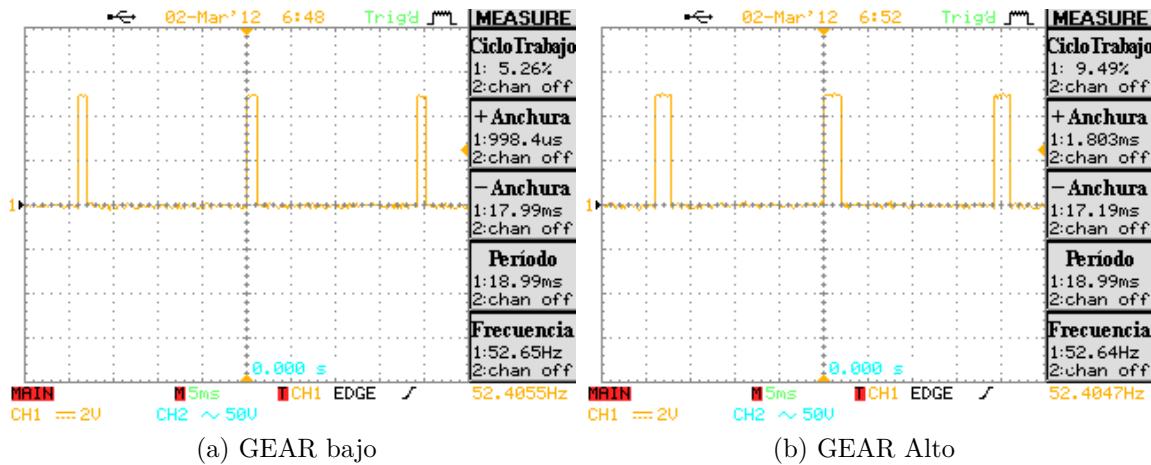


Figura 20.2.: Señales del receptor

con número de parte 0-PL752¹. En la figura 20.3 puede apreciarse el circuito elegido.



Figura 20.3.: Circuito para interpretar la señal DCM

A dicho circuito se le pueden conectar directamente las señales provenientes del receptor, ya que el mismo cuenta con la electrónica necesaria para interpretar dichas señales. De esta forma, la salida del circuito será “1” si el tiempo en alto de la señal proveniente del receptor es mayor a $1,6ms$ y “0” si dicho tiempo es menor a $1,5ms$. Es interesante, además, destacar que la placa adquirida es considerablemente pequeña, ya que la misma mide $1cm$ de lado y pesa apenas $0,3g$.

Finalmente, introduciendo dicha señal a un multiplexor TS3A5017 de Texas Instruments, donde cada canal tiene un contacto normal abierto y otro normal cerrado, es posible realizar el switcheo en forma sencilla.

En la figura 20.4 puede observarse el esquema de conexión de los componentes del sistema de switcheo.

La señal OUT será siempre “0” mientras no la señal de GEAR sea “0” o no se encuentre presente. En ese caso el multiplexor conecta 1S2 con 1D y 2S2 con 2D. Si se detecta que GEAR pasa a “1” la señal OUT es también “1”, esto produce que el multiplexor conecte 1S4 con 1D y 2S4 con 2D. Tenemos implementado un circuito que permite según la señal de GEAR del control remoto que los ESCs queden conectados al cerebro de fábrica o a la BeagleBoard a través de la placa de conversión de niveles lógicos. Esto permite el switcheo entre el modo automático y manual.

¹<http://www.robotcombat.com/products/0-PL752.html>

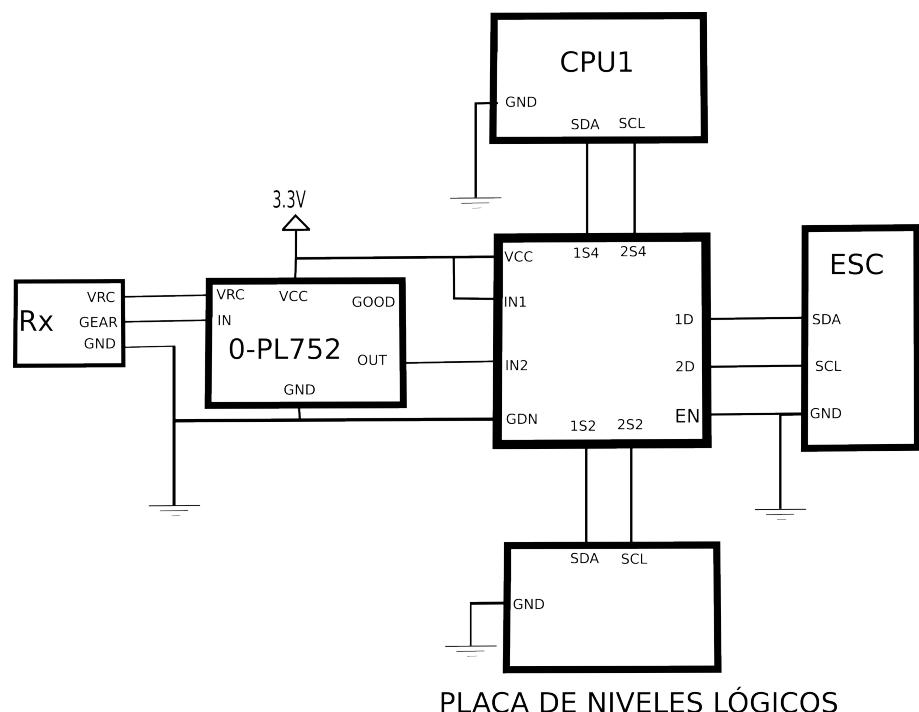


Figura 20.4.: Esquema eléctrico del switcheo

CONCLUSIÓN

A lo largo del presente texto se ha intentado transmitir la esencia del trabajo realizado durante la totalidad del Proyecto de Fin de Carrera. En lo que respecta estrictamente a los objetivos trazados se lograron cumplir la mayoría. Sin embargo el proyecto resultó ambicioso en cuanto al alcance planteado inicialmente en función de la poca experiencia del grupo en la gran mayoría de los desafíos que se encontraron. El proyecto en el que se trabajó comprende una gran variedad de áreas como la electrónica y el procesamiento de señales, pero es en esencia, un proyecto de control. Paradójicamente, los problemas específicos sobre el diseño en sí del controlador y del modelado del sistema no fueron las áreas en las que se encontraron mayores dificultades, probablemente debido a la experiencia ya adquirida a lo largo de la formación que hemos recibido en la Facultad de Ingeniería de la Universidad de la República. Previo al comienzo del proyecto, las dificultades de pasar de la teoría y de las simulaciones al sistema real eran inimaginables. Esta fue probablemente la mayor dificultad frente a la cual nos enfrentamos diariamente a lo largo de este trabajo.

A pesar de las dificultades planteadas y de no lograr cumplir con la totalidad de los ambiciosos objetivos trazados inicialmente, se desprenden diversas valoraciones positivas en cuanto a las enseñanzas del proyecto tanto en lo académico como en lo que refiere a la dinámica del trabajo en grupo.

En lo que respecta al alcance del proyecto, la gran mayoría de los puntos planteados fueron logrados con éxito. Se modeló el sistema, se desarrolló un simulador donde es posible testear la performance de las técnicas de control implementadas, se logró la integración del sistema dejando una plataforma estable y con gran capacidad, se decodificó el protocolo de comunicación con los motores logrando actuar sobre ellos, se desarrolló un control LQR automático y fundamentalmente **se logró el vuelo autónomo de una plataforma aérea no tripulada**. A su vez, como objetivo secundario, se desarrolló una aplicación para obtener la posición y orientación del cuadricóptero con una muy buena precisión (del orden de algunos centímetros en la posición), utilizando una cámara montada en el cuadricóptero, la cual por razones de tiempo no pudo ser testeada sobre el mismo (por más detalles, visitar el anexo A).

Además del modelado del sistema se desarrolló la teoría que permite trabajar con algunos tipos de trayectorias como rectas, círculos y equilibrio logrando simulaciones

20. Switcheo Manual/Automático

que arrojan resultados muy satisfactorios. Lamentablemente, por falta de tiempo estas trayectorias no pudieron ser testeadas en la práctica completamente. La única prueba que se realizó fue la de lograr el *hovering*, logrando resultados ampliamente satisfactorios.

Por otro lado, y aún más importante, se implementaron dos plataformas idénticas que le permiten al departamento de Sistemas y Control del instituto de Ingeniería Eléctrica de la Facultad de Ingeniería contar con nuevos elementos para continuar desarrollando técnicas de control orientas a la navegación no tripulada, ya sea en líneas de investigación independientes, en futuros Proyectos de Fin de Carrera o postgrados. El sistema integrado permite, con algunas salvedades, la puesta a prueba de las técnicas a desarrollar en un ambiente de laboratorio.

Gran parte del trabajo realizado consituyó en comprender la forma en que se debía actuar sobre los motores (dado que esta información no se encontraba disponible), y en comprender como lograr una adecuada medida de las variables de estado en función de los sensores de los que se disponía. La falta de experiencia llevó además a no realizar las compras adecuadas en una primera instancia, teniendo así demoras a lo largo del proyecto y duplicación del trabajo. A modo de ejemplo fue necesaria la inclusión de un magnetómetro, barómetro y termómetro para poder lograr una medida completa de todas las variables de interés, sensores que no fueron previstos en una primera instancia. Más allá de los contratiempos, idas y vueltas, quedó demostrado que efectivamente es posible lograr el vuelo autónomo de una vehículo aéreo no tripulado con los sensores utilizados, presupuesto acotado y mucha dedicación.

Como se adelantó en la introducción a este texto no era un objetivo lograr un producto final ni definitivo, sino que se trataba de integrar una plataforma que permita continuar la investigación en temáticas relacionadas con la navegación. En particular a partir de este momento se abren diversas líneas de investigación. En lo que refiere por ejemplo a la estimación del estado la principal linea a explorar es la inclusión de cámaras ya sea en el cuadricóptero o en la zona de prueba, a fin de evitar el problema de la falta de GPS en ambientes cerrados. Otra línea de trabajo posible en lo que refiere a la estimación del estado consiste en explorar en profundidad la forma de reducir los errores aportados por los sensores de forma de lograr mejores estimaciones de algunas variables (en particular las velocidades y posiciones en el plano horizontal).

A lo largo del presente proyecto se observaron en diversas oportunidades deficiencias en el control de los motores, en particular se comprobó que los mismos tienen un grave problema de diseño en lo que respecta a la disipación térmica, produciendo que algunos de los transistores de potencia que los componen se quemen. Esto produce que algunos motores dejen de funcionar. Esta deficiencia se observó sobre el final del proyecto en el momento de realizar las pruebas sobre el sistema completo, por lo tanto no hubo tiempo de resolver dicha dificultad. Debe señalarse la necesidad de re-diseñar estos controladores o adquirir otros de forma de eliminar un factor de riesgo. Asimismo puede aprovecharse para mejorar otros aspectos secundarios de los ESCs como por ejemplo lograr eliminar la variación de la velocidad que los motores presentan en función de la batería disponible. Asimismo, para algunas aplicaciones puede ser interesante contar con motores que puedan girar tanto en sentido horario

como anti-horario.

Finalmente se puede nombrar una linea de trabajo orientada a la creación de rutas óptimas y el desarrollo de algoritmos de control más sofisticados que involucren técnicas de control no lineales de forma de abrir el abanico de las trayectorias realizables. A modo de ejemplo a lo largo de este proyecto trabajamos con giros respecto del eje vertical, sin embargo algunas maniobras requieren giros respecto de otros ejes, pero las técnicas de control desarrolladas en este proyecto limitan estas posibilidades.

El trabajo realizado a lo largo del Proyecto de Fin de Carrera nos permitió enfrentarnos a un problema de ingeniería real y suficientemente complejo de el cual se obtienen enseñanzas más allá del plano académico. Principalmente se debe destacar la forma de encarar un problema que tiene una gran cantidad de sub problemas asociados que deben resolverse para lograr el correcto funcionamiento del conjunto. La identificación de las diferentes problemáticas a resolver es un ejercicio que hasta el momento no se había realizado. El segundo aspecto a destacar es la autogestión del tiempo para cumplir los objetivos a largo plazo, los plazos del mismo (a pesar de las entregas intermedias) son impuestos fundamentalmente por el grupo a diferencia de la experiencia en el resto de las asignaturas de la facultad.

Anexos

ANEXO A

APLICACIÓN CON CÁMARA

La idea principal de la aplicación consiste en implementar un método para determinar la posición del cuadricóptero utilizando una cámara filmadora. Surge como una solución a la necesidad de obtener una realimentación de posición estando a puertas cerradas, donde la señal del GPS es muy débil o directamente nula. A su vez es posible obtener una medida de los ángulos pitch, roll y yaw.



Figura A.1.: Presentación de la aplicación

parámetros intrínsecos de la cámara, obtener la posición espacial de la cámara respecto a cada patrón. La ubicación de los patrones respecto a la habitación es conocida, deduciendo así la posición del UAV en la habitación. Una vez detectados e identificados los marcadores, se halla la transformación que lleva el sistema de coordenadas referenciado en el marcador al sistema de coordenadas referenciado en la cámara, obteniéndose una matriz de rotación \mathbf{R} y un vector de traslación \mathbf{T} , los cuales describen completamente la posición relativa entre la cámara y el marcador.

Para poder determinar los 6 grados de libertad que tiene el problema son necesarios al menos 3 puntos de correspondencia entre las coordenadas del mundo (3D) y las de la foto (2D), aunque cuantos más puntos se tengan, mejores serán los resultados obtenidos.

En la figura A.1 se observa un diagrama que representa al problema a solucionar. Se desean hallar los parámetros de posición “ x ”, “ y ” y “ z ” y los parámetros de orientación: *roll*, *pitch* y *yaw*, mediante la identificación de algunos marcadores cuya posición y orientación es conocida.

Se consideran distintos patrones (marcadores) para identificar las distintas paredes de una habitación. Los patrones son reconocidos e identificados para luego, conociendo los

A. Aplicación con cámara

Es importante destacar que el algoritmo debe ser lo suficientemente rápido como para procesar las imágenes en tiempo real. El cuadricóptero equipado con la cámara estará filmando y es necesario obtener la posición sin un retardo demasiado grande. Es evidente que puede ocurrir que durante varios frames no se tenga ningún marcador a la vista, situación en la que se confiará plenamente en la estimación del estado obtenida del filtro de Kalman. Para la estimación de los ángulos esta situación no es un problema, ya que se puede obtener realimentación mediante el acelerómetro y el giróscopo.

A.1. Sistemas de coordenadas

Es importante definir algunas convenciones que se utilizarán más adelante para poder entender la aplicación. Por ello se aclaran los sistemas de coordenadas utilizados.

Por un lado se tienen las coordenadas del “mundo”, representadas en la figura A.2 como “ i ”, “ j ” y “ k ”, que son coordenadas espaciales (3D). Por otro lado, también representadas en la figura, están las coordenadas de la cámara que también son espaciales y se representan mediante “ i_c ”, “ j_c ” y “ k_c ”.

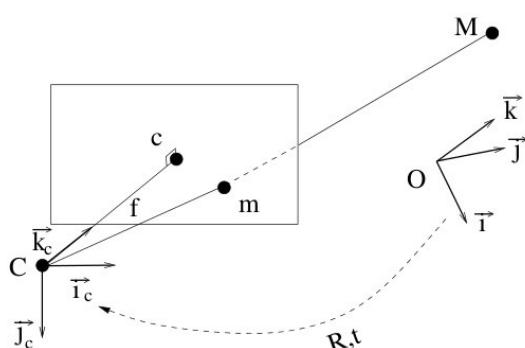


Figura A.2.: Sistemas de coordenadas

las coordenadas del mundo de “ M ” y “ X_c ” las coordenadas de “ M ” referenciadas a la cámara, se puede afirmar:

$$X_c = RX + T$$

“ m ” es la proyección de “ M ” en el plano de la imagen, llevando coordenadas en el plano (2D). “ c ” es el punto principal y “ f ” la distancia focal.

A.2. Marcadores

El marcador consta básicamente de 3 bloques de cuadrados concéntricos. El 4º lugar será ocupado por algún identificador para diferenciar distintos marcadores. El hecho de que tenga 3 bloques en 3 esquinas distintas permite conocer con facilidad

Además, y no menos importantes están las coordenadas de la foto. Son coordenadas en el plano y por lo tanto en 2 dimensiones. “ M ” representa un punto en el espacio con coordenadas “en el mundo” según (i, j, k) . A su vez es posible expresar las coordenadas de “ M ” en el marco de referencia de la cámara según los versores (ic, jc, kc) .

El pasaje entre las coordenadas “del mundo” y las coordenadas “de la cámara” se realiza mediante la rotación R y la translación T . Siendo “ X

la orientación del mismo. El tamaño del marcador es de 20x29.7 cm, el tamaño de una hoja A4 y se puede observar en la figura A.3.

A.3. Detección del marcador

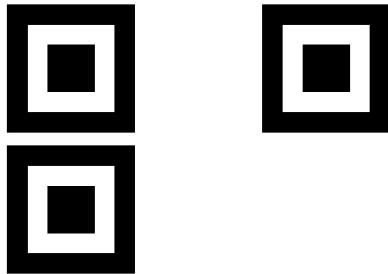


Figura A.3.: Marcador utilizado

Para la detección del marcador se utilizará como base la detección de segmentos con el algoritmo **LSD**¹.

Una vez detectados los segmentos se estudia cuáles de ellos pertenecen a estructuras formadas por 4 segmentos conexos y cerrados, es decir, se conservan solamente los segmentos pertenecientes a algún cuadrilátero. Posteriormente se realiza un nuevo filtrado dejando únicamente a las estructuras compuestas por 3 cuadrados concéntricos.

Una vez obtenidos los 3 bloques de 3 cuadrados concéntricos se utilizan todas las esquinas de todos los cuadrados para hallar la correspondencia entre las coordenadas del mundo y las esquinas detectadas en la foto. Se tienen así 36 puntos de correspondencias.

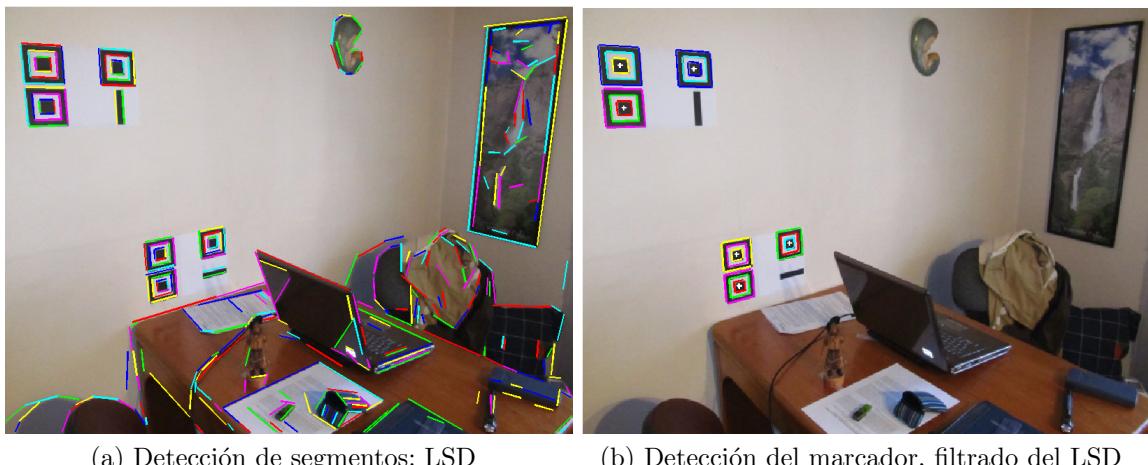


Figura A.4.: Detección y filtrado de segmentos

En la figura A.4 se muestra una fotografía donde aparecen 2 marcadores distintos. Al aplicarle el detector de segmentos LSD se obtuvo la figura A.4a. Luego se detectaron cuáles de estos segmentos pertenecen a estructuras como las descriptas anteriormente (3 bloques de 3 cuadrados concéntricos corresponden a un marcador), y los resultados se muestran en la figura A.4b.

¹ http://www.ipol.im/pub/algo/gjmr_line_segment_detector/

A. Aplicación con cámara

Por un lado es muy improbable obtener un falso positivo ya que los marcadores llevan una estructura muy particular, difícil de confundir. Por otro lado la debilidad de este método de detección está en poder detectar todos los segmentos de los 9 cuadrados involucrados. Es decir que si en la detección de segmentos no se detecta alguno de los 36 segmentos involucrados en los 9 cuadrados, el algoritmo no detectará ningún marcador.

Claramente se puede implementar algún tipo de algoritmo que introduzca cierta tolerancia, permitiendo que falten algunos segmentos, lo cual significaría una importante mejora a implementar ya que agrega una gran robustez al algoritmo.

A.4. Resultados

El lugar de la habitación donde se encuentra el marcador es conocido, entonces alcanza con obtener la posición relativa de la cámara respecto al marcador para obtener la posición de la cámara en relación a la habitación. Como ya se mencionó, existe una rotación **R** y una traslación **T** que llevan las coordenadas del mundo referenciadas en el marcador a las coordenadas del mundo referenciadas en la cámara. Entonces al obtener **R** y **T** estamos obteniendo la posición de la cámara. Observar que al transformar el origen de las coordenadas del mundo se obtienen unas coordenadas de la cámara iguales a **T**.

Como forma de verificar el buen funcionamiento del programa, se realizan pruebas en condiciones controladas, sabiendo con una precisión aceptable la verdadera posición relativa entre la cámara y el marcador. Se realizaron 7 pruebas en condiciones distintas, como se indica en la tabla A.1.

Prueba	Roll (°)	Pitch (°)	Yaw (°)	Z (cm)
1	0	0	0	50
2	0	-30	0	50
3	-30	0	0	50
4	-30	-30	0	50
5	0	0	-15	50
6	0	-30	-15	50
7	0	0	0	100

Cuadro A.1.: Pruebas realizadas

Se muestran a continuación únicamente los resultados obtenidos para la prueba 5, ya que los resultados para el resto de las pruebas es muy similar.

A.4.1. Prueba N° 5

Al ejecutar el programa se obtiene la siguiente salida:

```
roll    = 357.34°
pitch   = 5.68°
yaw     = 342.97°
x       = -0.59mm
```

$$\begin{aligned}y &= -0.58\text{mm} \\z &= 521.36\text{mm}\end{aligned}$$

El marcador tiene en el 4o lugar una barra horizontal.

Como se puede observar los resultados son aceptables, obteniendo un error de unos pocos grados en los ángulos, y apenas 2 centímetros en la posición. Además es importante destacar que se cometió un error sistemático en el proceso de adquisición ya que la distancia en el eje “z” se midió desde el marcador hasta el extremo del lente más cercano. En su lugar se debería haber considerado la distancia hasta donde se forma la imagen en la cámara, que es un tanto más adentro. Si se hubiera considerado este aspecto, la medida sería aún más fidedigna, ya que la distancia del lente al centro del plano de proyección sobre el que se forma la imagen en la cámara sumaría un par de centímetros a la medida realizada.

Reproyección

Este es quizás el método más confiable para determinar si los resultados obtenidos son correctos. En la figura A.5 se muestra con círculos azules las esquinas detectadas en la foto y con cruces rojas se indican la proyección de las coordenadas de la estructura 3D del marcador, utilizando las matrices \mathbf{R} y \mathbf{T} obtenidas. Puede verse claramente la cercanía entre los círculos azules y las cruces rojas, indicando que el conjunto de parámetros extrínsecos obtenidos son los correctos.

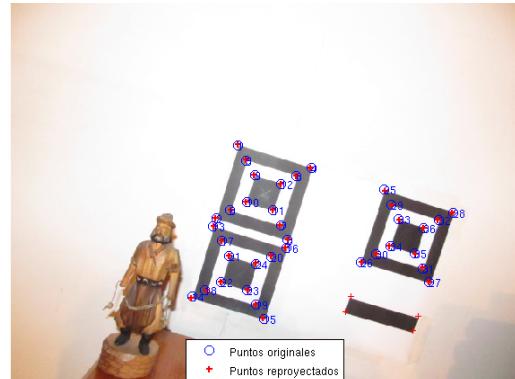


Figura A.5.: Reproyección de esquinas

Es importante destacar, de todas maneras, que esta verificación no es 100 % confiable ya que pueden haber diferentes combinaciones de parámetros intrínsecos de la cámara que determinen la misma reproyección. Por ejemplo los siguientes 2 juegos de parámetros:

- Field of view = A ; Distancia = B
- Field of view = A/2 ; Distancia = 2*B

Ploteo de la posición relativa

Consiste en una herramienta gráfica que permite verificar a grandes rasgos si los resultados son coherentes con la realidad.

En la figura A.6 se presentan 2 imágenes: la primera basada en las coordenadas de la cámara, es decir dejando la cámara fija y ubicando el marcador, y la segunda basada en las coordenadas del mundo, es decir dejando el marcador fijo y ubicando la posición relativa de la cámara.

A. Aplicación con cámara

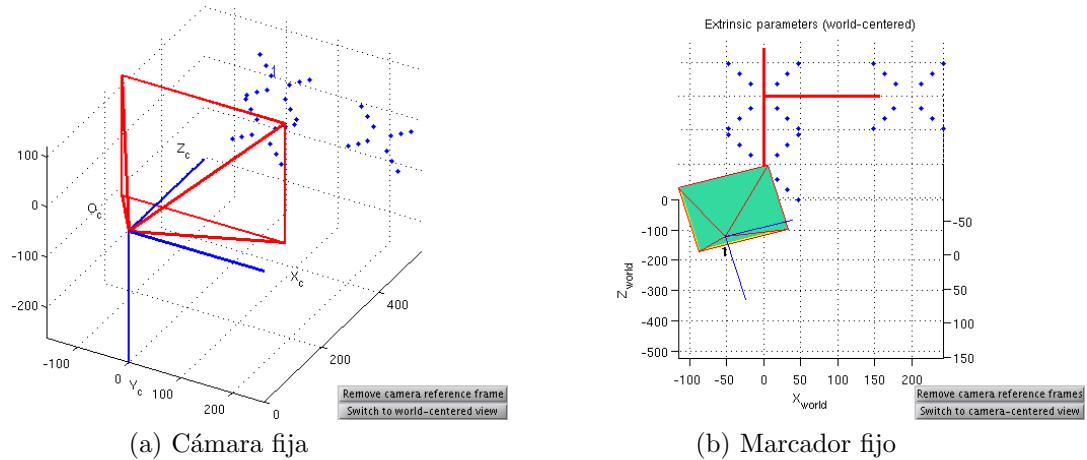


Figura A.6.: Ploteo de posición y orientación obtenida

Realidad aumentada

Por último y a modo didáctico, se presenta un ejemplo de Realidad Aumentada. Se trata de proyectar en la imagen una estructura tridimensional como si estuviera en la habitación. Resulta interesante ya que se puede apreciar con claridad la perspectiva con la que está mirando al marcador y analizar si la perspectiva del objeto añadido se corresponde.

Se añade un cubo con una de las caras coincidente con uno de los cuadrados del marcador, y sobre él aparece parado un “fosforito”. Los resultados se pueden apreciar en la figura A.7.

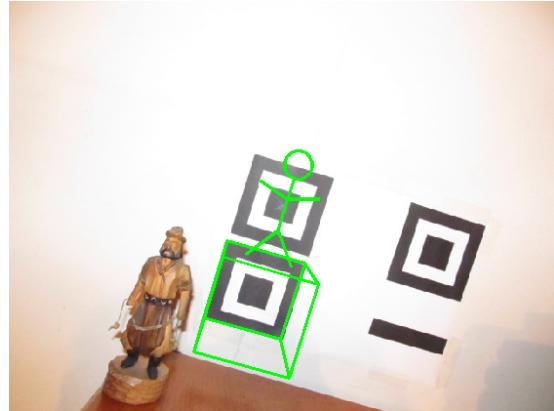


Figura A.7.: Realidad aumentada

Nota general: Si bien el algoritmo fue implementado y testeado, todo el trabajo fue realizado en *MatLab*. Por razones del tiempo acotado del proyecto no se pudo portar el código a *C*, quedando como trabajo a ser realizado en el futuro. Se cuenta con todo el hardware necesario para su implementación ya que la *BeagleBoard* tiene capacidad para procesar video en tiempo real, cuenta con librerías especiales para el tratamiento de imágenes y se adquirió una cámara pensada para la placa que tiene las ventajas de un fácil montaje y compatibilidad asegurada.

ANEXO B

ESPECIFICACIONES TÉCNICAS DE LA MONGOOSE 9DOF IMU

B.1. Acelerómetro

Rango	$\pm 16g$
Resolución	10-13 bits (siempre 4mg/LSB)
Datos nuevos	0.1 a 800 Hz
Ruído XY	0.75@100Hz - 3@3200Hz LSB-rms
Ruído Z	1.1@100Hz - 4.5@3200Hz LSB-rms
Cross Axis	$\pm 1\%$

NOTAS:

- Output data rate puede llegar a 3200Hz, pero usando SPI. Con I²C a 400kHz solamente se puede llegar a 800Hz.
- Ancho de banda = *Datos_nuevos*/2

B.2. Giróscopo

Rango	$\pm 2000^\circ/s$
Resolución	14.475 LSB/(°/s)
Datos nuevos	3.9Hz a 8kHz
Ancho de banda	256Hz
Cross Axis	$\pm 2\%$
Ruído	0.38 °/s-rms

NOTAS:

- Datos nuevos: La muestras pasan por un LPF digital de 256 a 5Hz, esto limita el ancho de banda.

B. Especificaciones técnicas de la Mongoose 9DoF IMU

B.3. Magnetómetro

Rango	± 8 Ga
Resolución	5mGa@GN=2
Datos nuevos	0.75 - 75Hz
Ancho de banda	37Hz
Cross Axis	$\pm 0.2\%$ FS/Ga
Ruído	-

NOTAS:

- El rango queda determinado por la ganancia, que se configura con 3 bits:

GN	0	1	2	3	4	5	6	7
Rango (Ga)	± 0.88	± 1.3	± 1.9	± 2.5	± 4.0	± 4.7	± 5.6	± 8.1

- Se puede configurar para que el dato que muestre sea el promedio de hasta 8 muestras.

B.4. Sensor de presión

Rango	300 a 1100 hPa (9000 a -500m)
Resolución	1Pa
Precisión. Abs.	typ/max $\pm 1.0/\pm 3.0$ hPa
Precisión Rel.	± 0.5 hPa
Datos nuevos	typ/max: 3/4.5ms - 17/25ms
Ancho de banda	333/40Hz
Ruído (hPa)	0.06 - 0.03
Ruído (m)	0.5 - 0.25

NOTAS:

- El rango, en altura, se refiere a la altura sobre el nivel del mar.
- El modo de operación (cantidad de muestras promediadas) afecta:
 - El tiempo de conversión.
 - El ancho de banda.
 - La resolución.
 - El ruído.
- Es necesario hacer una medida de temperatura de vez en cuando (1Hz) para mejorar la lectura del sensor de presión.

B.5. Sensor de temperatura

Rango	0 a 65 °C
Resolución	0.1 °C
Precisión Abs.	typ/max $\pm 1.0/\pm 2.0$ °C
Datos nuevos	typ/max: 3/4.5ms
Ruído	-

NOTAS:

- El sensor de temperatura está incorporado al sensor de presión.

ANEXO C

CÁLCULO DEL TENSOR DE INERCIA

C.1. Magnitudes a considerar

Según el modelo considerado explicado en 3 dividiremos el sistema en una esfera central, cuatro cilindros y cuatro varillas (también cilíndricas). Las magnitudes que debemos conocer para calcular los momentos de inercia del sistema son:

- Radio de la esfera central $R = 8 \times 10^{-2}m$
- Largo de las varillas $L = 26 \times 10^{-2}m$
- Radio de los motores $r = 1,65 \times 10^{-2}m$
- Altura de los motores $h = 3,5 \times 10^{-2}m$

Además nos interesa conocer las distancias de cada uno de los elementos del sistema al centro de masa del cuadricóptero.

- Distancia entre el centro de masa de la varilla y el centro de masa del cuadricóptero $d_v = 14 \times 10^{-2}m$
- Distancia del eje de los motores al centro de masa del cuadricóptero $d_m = 0,29m$

Por último las masas de los elementos en cuestión son:

- Masa de la esfera central $M_E = 1,037kg$
- Masa de las varillas $M_v = 0,013kg$
- Masa de los motores $M_m = 0,113kg$

C.2. Tensor de inercia del sistema

El tensor de inercia del sistema puede calcularse como la suma de los tensores de inercia de los rígidos que lo componen. Se considera como fue expresado anteriormente el centro del cuadricóptero como una esfera maciza. El tensor de inercia de dicha esfera puede calcularse a partir de la definición misma de tensor de inercia, sin embargo por ser una forma geométrica de vasto uso en el campo de la mecánica su tensor de inercia se encuentra ya tabulado. Sucede lo mismo con las restantes formas geométricas que componen al sistema. Los tensores utilizados pueden obtenerse en [16]. En el caso de la esfera se tiene que el tensor de inercia respecto de su centro de masa es:

$$\Pi_{G_E}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}} = M_E \begin{pmatrix} \frac{2R^2}{5} & 0 & 0 \\ 0 & \frac{2R^2}{5} & 0 \\ 0 & 0 & \frac{2R^2}{5} \end{pmatrix} \quad (\text{C.1})$$

En este caso el centro de masa del sistema corresponde al centro de masa de la esfera a partir de ciertas suposiciones que se realizan sobre la simetría del sistema. Por dicho motivo podemos afirmar que $\Pi_{G_E}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}} = \Pi_{O'_E}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}}$, siendo O' el centro de la esfera.

Por otra parte el tensor de inercia de una varilla, cuyo eje principal coincide con el vedor \vec{i}_q , respecto a su centro de masa tiene la forma:

$$\Pi_{G_{Vx}}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}} = M_V \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{L^2}{12} & 0 \\ 0 & 0 & \frac{L^2}{12} \end{pmatrix} \quad (\text{C.2})$$

Sin embargo resulta mucho más interesante obtener el tensor de inercia expresado respecto del centro de masa del sistema. Para realizar dicho cambio se utiliza el *Teorema de Steiner*. Dicho teorema afirma que: $\Pi_Q = \Pi_G + J_Q^{M,G}$, donde los términos de $J_Q^{M,G}$ pueden calcularse como: $(J_Q^{M,G})_{\alpha\beta} = M(G-Q)^2\delta_{\alpha\beta} - M(G-Q)_\alpha M(G-Q)_\beta$. El término $\delta_{\alpha\beta}$ es conocido como Delta de Kronecker. Su valor es uno si $\alpha = \beta$ y cero si $\alpha \neq \beta$. En el caso en consideración dicha matriz resulta en:

$$J_{O'}^{M_{Vx},G} = M_V \begin{pmatrix} 0 & 0 & 0 \\ 0 & (\frac{L}{2} + d_v)^2 & 0 \\ 0 & 0 & (\frac{L}{2} + d_v)^2 \end{pmatrix} \quad (\text{C.3})$$

Por lo tanto el momento de inercia total de dicha varilla es:

$$\Pi_{O'_{Vx}}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}} = M_V \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{L^2}{3} + (\frac{Ld_v}{2})^2 + d_v^2 & 0 \\ 0 & 0 & \frac{L^2}{3} + (\frac{Ld_v}{2})^2 + d_v^2 \end{pmatrix} \quad (\text{C.4})$$

Análogamente, el tensor de inercia de una varilla cuyo eje principal se encuentra respecto de la dirección \vec{j}_q respecto del centro de masa del sistema es:

$$\Pi_{O'_{Vy}}^{\{\vec{i}_q, \vec{j}_q, \vec{k}_q\}} = M_V \begin{pmatrix} \frac{L^2}{3} + (\frac{Ld_v}{2})^2 + d_v^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{L^2}{3} + (\frac{Ld_v}{2})^2 + d_v^2 \end{pmatrix} \quad (\text{C.5})$$

Sucede algo similar en lo que respecta a los motores. Tendremos un tensor de inercia para los motores que se encuentran sobre la dirección \vec{i}_q y otro para los

motores que se encuentran sobre la dirección \vec{j}_q . El momento de inercia de un cilindro en su centro de masa es:

$$\Pi_{G_M}^{\{i_q, j_q, k_q\}} = M_M \begin{pmatrix} \frac{3r^2+h^2}{12} & 0 & 0 \\ 0 & \frac{3r^2+h^2}{12} & 0 \\ 0 & 0 & \frac{r^2}{2} \end{pmatrix} \quad (\text{C.6})$$

La entrada (3, 3) de la matriz de la ecuación C.6 la pasaremos a llamar I_{zmm}

Por lo tanto el momento de inercia respecto del centro de masa del sistema para un cilindro que se encuentra en la dirección \vec{i}_q es:

$$\Pi_{O'_{Mx}}^{\{i_q, j_q, k_q\}} = M_M \begin{pmatrix} \frac{3r^2+h^2}{12} & 0 & 0 \\ 0 & \frac{3r^2+h^2}{12} + d_m^2 & 0 \\ 0 & 0 & \frac{r^2}{2} + d_m^2 \end{pmatrix} \quad (\text{C.7})$$

En el caso de un cilindro que se encuentra en la dirección \vec{j}_q se tiene que::

$$\Pi_{O'_{My}}^{\{i_q, j_q, k_q\}} = M_M \begin{pmatrix} \frac{3r^2+h^2}{12} + d_m^2 & 0 & 0 \\ 0 & \frac{3r^2+h^2}{12} & 0 \\ 0 & 0 & \frac{r^2}{2} + d_m^2 \end{pmatrix} \quad (\text{C.8})$$

Finalmente, el tensor de inercia del sistema, se calcula como la suma de los tensores de inercia de las partes que lo componen:

$$\Pi_{O'}^{\{i_q, j_q, k_q\}} = \Pi_{O'_E}^{\{i_q, j_q, k_q\}} + 2\Pi_{O'_{Vx}}^{\{i_q, j_q, k_q\}} + 2\Pi_{O'_{Vy}}^{\{i_q, j_q, k_q\}} + 2\Pi_{O'_{Mx}}^{\{i_q, j_q, k_q\}} + 2\Pi_{O'_{My}}^{\{i_q, j_q, k_q\}} \quad (\text{C.9})$$

Dado que todos los tensores de inercia considerados hasta el momento son diagonales, podemos escribir el tensor de inercia del sistema como:

$$\Pi_{O'}^{\{i_q, j_q, k_q\}} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (\text{C.10})$$

C.3. Resultados

En base al análisis realizado hasta el momento se obtienen los siguientes valores:

- $I_{zmm} = 1,54 \times 10^{-5} \text{kgm}^2$
- $I_{xx} = I_{yy} = 2,32 \times 10^{-2} \text{kgm}^2$
- $I_{zz} = 4,37 \times 10^{-2} \text{kgm}^2$

ANEXO D

CÁLCULOS NECESARIOS PARA LA LINEALIZACIÓN DEL SISTEMA

Tal como se explica en el capítulo 13 se linealiza el MVE obtenido en 3, esto es, aproximar el sistema no lineal por un sistema lineal de la forma:

$$\dot{\tilde{X}}(t) = A(t)\tilde{X}(t) + B(t)\tilde{u}(t) \quad (\text{D.1})$$

Donde $A(t)$ y $B(t)$ son tales que $a_{ij} = \frac{\partial f_i}{\partial x_j}|_{x=x^*}^{u=u^*}$ y $b_{ij} = \frac{\partial f_i}{\partial u_j}|_{x=x^*}^{u=u^*}$

D.1. Linealización para cualquier trayectoria

Para una trayectoria genérica al linealizar el sistema se obtienen las matrices:

$$A(t) = \begin{pmatrix} 0 & A_{12} & A_{13} & 0 \\ 0 & A_{22} & 0 & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & A_{42} & 0 & A_{44} \end{pmatrix} \quad (\text{D.2})$$

$$B(t) = \begin{pmatrix} 0 \\ 0 \\ B_{31} \\ B_{41} \end{pmatrix} \quad (\text{D.3})$$

Las entradas de la matriz de D.2 representan matrices de 3×3 mientras que las entradas de la matriz D.3 representan matrices de 3×4 . Es importante conocer las entradas de las matrices anteriores, al menos en lo que respecta a las dependencias de cada una de ellas con las variables de estado.

$$A_{12} = \begin{pmatrix} f_{A_{12_1}}(\psi, \varphi, \theta, v_{qy}, v_{qz}) & f_{A_{12_2}}(\psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}) & f_{A_{12_3}}(\psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}) \\ f_{A_{12_4}}(\psi, \varphi, \theta, v_{qy}, v_{qz}) & f_{A_{12_5}}(\psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}) & f_{A_{12_6}}(\psi, \varphi, \theta, v_{qx}, v_{qy}, v_{qz}) \\ f_{A_{12_7}}(\psi, \varphi, v_{qy}, v_{qz}) & f_{A_{12_8}}(\psi, \varphi, v_{qx}, v_{qy}, v_{qz}) & 0 \end{pmatrix} \quad (\text{D.4})$$

D. Cálculos necesarios para la linealización del sistema

$$A_{13} = \begin{pmatrix} f_{A_{13_1}}(\psi, \varphi, \theta) & f_{A_{13_2}}(\psi, \varphi, \theta) & f_{A_{13_3}}(\psi, \varphi, \theta) \\ f_{A_{13_4}}(\psi, \varphi, \theta) & f_{A_{13_5}}(\psi, \varphi, \theta) & f_{A_{13_6}}(\psi, \varphi, \theta) \\ f_{A_{13_7}}(\varphi) & f_{A_{13_8}}(\psi, \varphi) & f_{A_{13_8}}(\psi, \varphi) \end{pmatrix} \quad (\text{D.5})$$

$$A_{22} = \begin{pmatrix} f_{A_{22_1}}(\psi, \varphi, \omega_{qy}, \omega_{qz}) & f_{A_{22_2}}(\psi, \varphi, \omega_{qy}, \omega_{qz}) & 0 \\ f_{A_{22_4}}(\psi, \omega_{qy}, \omega_{qz}) & 0 & 0 \\ f_{A_{22_7}}(\psi, \varphi, \omega_{qy}, \omega_{qz}) & f_{A_{22_8}}(\psi, \varphi, \omega_{qy}, \omega_{qz}) & 0 \end{pmatrix} \quad (\text{D.6})$$

$$A_{24} = \begin{pmatrix} 1 & f_{A_{24_2}}(\psi, \varphi) & f_{A_{24_3}}(\psi, \varphi) \\ 0 & f_{A_{24_5}}(\psi) & f_{A_{24_6}}(\psi) \\ 0 & f_{A_{24_8}}(\psi, \varphi) & f_{A_{24_9}}(\psi, \varphi) \end{pmatrix} \quad (\text{D.7})$$

$$A_{32} = \begin{pmatrix} 0 & g \cos \varphi & 0 \\ -g \cos \varphi \cos \psi & g \sin \varphi \sin \psi & 0 \\ g \cos \varphi \sin \psi & g \sin \varphi \cos \psi & 0 \end{pmatrix} \quad (\text{D.8})$$

$$A_{33} = \begin{pmatrix} 0 & \omega_{qz} & -\omega_{qy} \\ -\omega_{qz} & 0 & \omega_{qx} \\ \omega_{qy} & -\omega_{qx} & 0 \end{pmatrix} \quad (\text{D.9})$$

$$A_{34} = \begin{pmatrix} 0 & -v_{qz} & v_{qy} \\ v_{qz} & 0 & -v_{qx} \\ -v_{qy} & v_{qx} & 0 \end{pmatrix} \quad (\text{D.10})$$

$$A_{42} = Mgd \begin{pmatrix} -\frac{\cos \varphi \cos \psi}{I_{xx}} & \frac{\sin \varphi \sin \psi}{I_{xx}} & 0 \\ 0 & -\frac{\cos \phi}{I_{yy}} & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{D.11})$$

$$A_{44} = \begin{pmatrix} 0 & \frac{I_{zzm}}{I_{xx}}(\omega_1 - \omega_2 + \omega_3 - \omega_4) + \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_{qz} & \frac{I_{yy} - I_{zz}}{I_{xx}}\omega_{qy} \\ \frac{I_{zzm}}{I_{yy}}(\omega_1 - \omega_2 + \omega_3 - \omega_4) + \frac{-I_{xx} + I_{zz}}{I_{yy}}\omega_{qz} & 0 & \frac{-I_{xx} + I_{zz}}{I_{yy}}\omega_{qx} \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{D.12})$$

$$B_{31} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ f_{B_{31_9}}(\omega_1) & f_{B_{31_10}}(\omega_2) & f_{B_{31_11}}(\omega_3) & f_{B_{31_12}}(\omega_4) \end{pmatrix} \quad (\text{D.13})$$

$$B_{41} = \begin{pmatrix} 0 & f_{B_{41_2}}(\omega_2) & 0 & f_{B_{41_4}}(\omega_4) \\ f_{B_{41_5}}(\omega_1) & 0 & f_{B_{41_6}}(\omega_3) & 0 \\ f_{B_{41_9}}(\omega_1) & f_{B_{41_10}}(\omega_2) & f_{B_{41_11}}(\omega_3) & f_{B_{41_11}}(\omega_4) \end{pmatrix} \quad (\text{D.14})$$

D.2. Linealización para la condición de hovering

Con las condiciones 13.6 y 13.7 obtenidas en el capítulo 13 se pueden obtener las matrices A y B en el caso particular de hovering. Además puede verificarse que todas las entradas de dichas matrices son constantes y que por lo tanto estamos frente a un sistema lineal invariante en el tiempo.

$$A_{hov} = \begin{pmatrix} 0 & 0 & A_{hov_{13}} & 0 \\ 0 & 0 & 0 & Id \\ 0 & A_{hov_{32}} & 0 & 0 \\ 0 & A_{hov_{42}} & 0 & 0 \end{pmatrix} \quad (\text{D.15})$$

$$B_{hov} = \begin{pmatrix} 0 \\ 0 \\ B_{hov_{31}} \\ B_{hov_{41}} \end{pmatrix} \quad (\text{D.16})$$

Donde Id es la matriz identidad y las matrices $A_{hov_{13}}, A_{hov_{32}}, B_{hov_{31}}, A_{hov_{41}}$ y $A_{hov_{42}}$ son:

$$A_{hov_{13}} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad A_{hov_{32}} = \begin{pmatrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{D.17})$$

$$A_{hov_{42}} = Mgd \begin{pmatrix} -\frac{1}{I_{xx}} & 0 & 0 \\ 0 & -\frac{1}{I_{zz}} & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (\text{D.18})$$

$$B_{hov_{31}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1,5 \times 10^{-2} s^{-2} & 1,5 \times 10^{-2} s^{-2} & 1,5 \times 10^{-2} s^{-2} & 1,5 \times 10^{-2} s^{-2} \end{pmatrix} \quad (\text{D.19})$$

$$B_{hov_{41}} = \begin{pmatrix} 0 & 2,9 \times 10^{-1} s^{-2} & 0 & -2,9 \times 10^{-1} s^{-2} \\ -2,9 \times 10^{-1} s^{-2} & 0 & 2,9 \times 10^{-1} s^{-2} & 0 \\ 5,0 \times 10^{-2} s^{-2} & -5,0 \times 10^{-2} s^{-2} & 5,0 \times 10^{-2} s^{-2} & -5,0 \times 10^{-2} s^{-2} \end{pmatrix} \quad (\text{D.20})$$

Dado que fue impuesto que θ fuese constante para este movimiento, tenemos efectivamente un sistema lineal invariante en el tiempo.

D. Cálculos necesarios para la linealización del sistema

D.3. Vuelo en linea recta a velocidad constante

Las matrices A y B obtenidas para esta situación de vuelo son:

$$A_{rec} = \begin{pmatrix} 0 & A_{rec_{12}} & A_{rec_{13}} & 0 \\ 0 & 0 & 0 & Id \\ 0 & A_{rec_{32}} & 0 & A_{rec_{34}} \\ 0 & A_{rec_{42}} & 0 & 0 \end{pmatrix} \quad (\text{D.21})$$

$$B_{rec} = \begin{pmatrix} 0 \\ 0 \\ B_{rec_{31}} \\ B_{rec_{41}} \end{pmatrix} \quad (\text{D.22})$$

Donde $A_{rec_{13}} = A_{hov_{13}}$, $A_{rec_{32}} = A_{hov_{32}}$, $A_{rec_{42}} = A_{hov_{42}}$, $B_{rec} = B_{hov}$ y

$$A_{rec_{12}} = \begin{pmatrix} v_{qz} \sin \theta & v_{qz} \cos \theta & -v_{qy} \cos \theta - v_{qx} \sin \theta \\ -v_{qz} \cos \theta & v_{qz} \sin \theta & v_{qx} \cos \theta - v_{qy} \sin \theta \\ v_{qy} & -v_{qx} & 0 \end{pmatrix} \quad (\text{D.23})$$

$$A_{hov_{32}} = \begin{pmatrix} 0 & -v_{qz} & v_{qy} \\ v_{qz} & 0 & -v_{qx} \\ -v_{qy} & v_{qx} & 0 \end{pmatrix} \quad (\text{D.24})$$

D.4. Vuelo a velocidad angular constante

Luego de las modificaciones introducidas en el MVE en la sección 13 se procede a linealizar el sistema obtenido.

$$A(t) = \begin{pmatrix} A_{cir_{11}} & 0 & Id & A_{cir_{14}} \\ 0 & A_{22} & 0 & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & A_{42} & 0 & A_{44} \end{pmatrix} \quad (\text{D.25})$$

$$B(t) = \begin{pmatrix} 0 \\ 0 \\ B_{31} \\ B_{41} \end{pmatrix} \quad (\text{D.26})$$

Las matrices obtenidas son todas idénticas a las obtenidas en la linealización del MVE original a excepción de las matrices $A_{cir_{11}}$ y $A_{cir_{14}}$. Estas tienen la forma:

$$A_{circ_{11}} = \begin{pmatrix} 0 & -\omega_{qz} & \omega_{qy} \\ \omega_{qz} & 0 & -\omega_{qx} \\ -\omega_{qy} & \omega_{qx} & 0 \end{pmatrix} \quad A_{cir_{14}} = \begin{pmatrix} 0 & z_z & -y_q \\ -z_q & 0 & x_q \\ y_q & -x_q & 0 \end{pmatrix} \quad (\text{D.27})$$

ANEXO E

GPS - INFORMACIÓN ADICIONAL

E.1. Geometría: DOP - *Dilution of precision*

El método que utiliza el GPS para determinar su ubicación consiste básicamente en:

1. Determinar la distancia r_i a cada satélite S_i , cuya posición es P_i .
2. Repetir el paso anterior para cada satélite disponible.
3. Intersestar las “cáscaras” de las esferas de centros P_i y radios r_i .

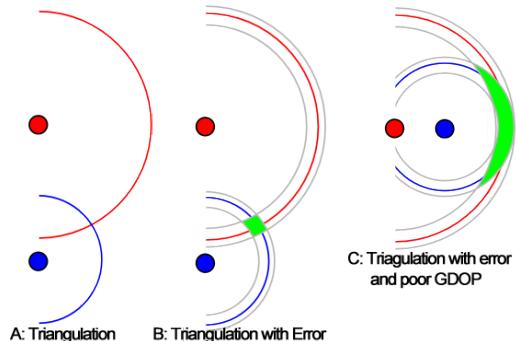


Figura E.1.: DOP en 2 dimensiones.

Las “cáscaras” de las esferas no serán de ancho despreciable, ya que hay un cierta incertidumbre asociado a los datos. Esto implica que la intersección será un volumen, en lugar de un punto. La geometría de la distribución de los satélites determinará el tamaño de este volumen, y por lo tanto la incertidumbre en la determinación de la posición. En la figura E.1¹ se muestra un ejemplo ilustrativo, en dos dimensiones.

El DOP es el cociente entre la exactitud de la ubicación y exactitud de la medida[17]:

$$\sigma = \sigma_o \cdot DOP$$

donde

- σ : Exactitud de la medida.
- σ_o : Exactitud de la posición.

Básicamente, el DOP representa la sensibilidad de localización frente a errores en las medidas. Dicho de otro modo, da una idea de qué tanto error se cometerá en la ubicación ante errores en las medidas. Cuanto más bajo sea el DOP, se cometerá un error menor. En la tabla E.1 se muestra como interpretar valores típicos (información tomada de [18]).

¹Imagen obtenida de Wikipedia

E. GPS - Información adicional

1 1-2	Ideal Excelente	Máxima exactitud posible. La exactitud a este nivel se considera suficiente para casi cualquier aplicación.
2-5	Bueno	Este nivel marca el mínimo apropiado para navegación.
5-10	Moderado	Las medidas se pueden utilizar, pero es recomendado buscar un lugar con cielo más abierto.
10-20	Regular	Solo se deben usar los datos para estimaciones de muy poca precisión.
>20	Malo	A este nivel, las exactitud de las medidas puede tener un error de hasta 300m, deben descartarse.

Cuadro E.1.: Interpretación de los valores de DOP.

El GPS envía información sobre:

- *HDOP* - DOP horizontal:

$$HDOP = \sqrt{\sigma_{easting}^2 + \sigma_{northing}^2} \quad (\text{E.1})$$

- *VDOP* - DOP vertical:

$$VDOP = \sqrt{\sigma_{altitud}^2} \quad (\text{E.2})$$

- *PDOP* - DOP de la posición:

$$PDOP = \sqrt{\sigma_{easting}^2 + \sigma_{northing}^2 + \sigma_{altitud}^2} \quad (\text{E.3})$$

Esta información se puede utilizar en el algoritmo de control, para ponderar los datos provenientes del GPS.

ANEXO F

CONFIGURACIÓN Y COMPILACIÓN DEL CÓDIGO

En este anexo se explican los conceptos básicos necesarios para trabajar con la *BeagleBoard*.

F.0.1. Obtención del código - Git

El código está almacenado en un repositorio git disponible en el DVD adjunto a esta documentación y en *github*: `git://github.com/r1rosa/uquad.git`. El repositorio ocupa aproximadamente 4GB, por lo que no es conveniente poner otras cosas en la tarjeta SD de la Beagleboard, ya que sino se excederá su capacidad.

F.0.2. Comunicación con la *BeagleBoard*

La forma básica de comunicarse con la Beagleboard es mediante el puerto serie, usando un conversor *RS232* a USB. Durante el vuelo, la comunicación se establece mediante *WiFi*, usando un dongle USB. El procedimiento para hacerlo funcionar es el siguiente:

1. Copiar el firmware `scripts/rt73.bin` a la Beagleboard en `/lib/firmware/`.
2. Instalar el software necesario haciendo
`opkg install kernel-module-rt73usb rt73-firmware`.

Una forma de conectarse es usando una red *Ad-Hoc*, aunque también se puede utilizar un servidor *DHCP*. Para configurar la interfaz `wlan0`¹ para trabajar en modo *Ad-Hoc* agregar las siguientes líneas a `/etc/network/interfaces`:

```
auto wlan0
iface wlan0 inet static
    address 10.42.43.2
    netmask 255.255.255.0
    wireless-mode ad-hoc
```

¹El dongle puede ser asociado a otra interfaz, como `wlan1`, configurar la apropiada.

F. Configuración y compilación del código

```
wireless-essid uquad  
wireless-key s:uquaduquad123
```

Luego, levantar una red en una laptop, en modo *Ad-Hoc*, con nombre *uquad* y contraseña *uquaduquad123* en *WEP 40/128-bit Key(Hex or ASCII)*. La laptop tendrá la IP *10.42.43.1*, y la Beagleboard *10.42.43.2*. El usuario en la Beagleboard es *root* y la clave cualquiera. Para conectarse ver la subsección sobre **ssh**.

Ssh

El usuario *root* acepta cualquier contraseña. Para evitar que la Beagleboard pida contraseña, se puede agregar la clave pública de una laptop a la Beagleboard, en²:

```
/home/root/.ssh/authorized_keys
```

Para conectarse hacer:

```
ssh root@10.42.43.2
```

Ethernet

La Beagleboard sabe conectarse a una red en la que haya un servidor *DHCP*. Elegirle un nombre en */etc/hostname*, por ejemplo *beagle*, y luego hacer:

```
ssh root@beagle.local
```

Configuración del proxy

Para trabajar con la Beagleboard en el laboratorio de medidas, es necesario configurar el proxy. Para ello, ejecutar los siguiente comandos en la Beagleboard:

```
echo "option http_proxy http://httpproxy.fing.edu.uy:3128/" \  
>> /etc/network/options.conf  
echo "option http_proxy http://httpproxy.fing.edu.uy:3128/" \  
>> /etc/opkg/arch.conf
```

F.0.3. Tiempos

El *main* corre sobre linux, lo cual simplifica algunas cosas, pero complica otras. Correr arriba de un sistema operativo que no es RT³ inevitablemente introduce demoras. Se configura el *main* y el driver de los motores para tener máxima prioridad.

La performance del *main* es **INACCEPTABLE** si se accede a memoria no volátil o si se abren sesiones ssh durante su ejecución.

²Crear el archivo y/o el directorio, si estos no existen.

³RealTime.

F.0.4. Consideraciones de seguridad

Al comenzar el *main*, intentará establecer una conexión TCP con un servidor en la laptop, y abortará en caso de no tener éxito. Si en algún momento se pierde la conexión, por ejemplo por algún error en el driver del dongle *WiFi*, el *main* abortará, apagando los motores del cuadricóptero. Para iniciar el servidor en la laptop ubicarse en el directorio `src/build/check_net` y ejecutar `./server`.

En caso de perderse la conexión ssh, y que el cuadricóptero siga funcionando (porque la conexión TCP sigue abierta), se puede detener el servidor que corre en la laptop, lo cual hará que el *main* aborte y apague los motores.

El *main* verifica que la inclinación del cuadricóptero no supere cierto límites durante un tiempo dado. Si esto sucede, apagará los motores. Esto puede ser útil si por algún motivo se pierde el control mediante software, pero se puede acceder al cuadricóptero, ya que inclinándolo lo suficiente se apagarían los motores.

El driver de los motores (*cmd*) tiene que estar funcionando para que los motores permanezcan prendidos. La pérdida de energía en la *BeagleBoard* detendría al driver, y por lo tanto también a los motores (El frenado tardaría unos segundos, ya que se omitiría la secuencia de frenado).

F.1. Salida - Logs

El *main* genera varios logs durante su ejecución, que sirven como realimentación al usuario para determinar la causa de errores o comportamientos extraños durante un vuelo:

- Datos crudos provenientes de la IMU: *imu_raw.log*
- Datos utilizados para estimar el estado: *kalman_in.log*
- Estado estimado: *x_hat.log*
- Acción de control: *w.log*

La primer columna de *imu_raw.log*, *kalman_in.log* y *w.log* es el tiempo en segundos desde el comienzo del *main*. La diferencia de tiempo entre *kalman_in.log* y *x_hat.log* es despreciable, por lo que *x_hat.log* no tiene columna de tiempos. Referirse al código fuente por información sobre como habilitar logs, nombres, contenido, etc.

Por cada log se crea un programa independiente (de ahora en más *logger*), que se comunica con el *main* mediante *pipes*. Cada logger tiene un espacio limitado en RAM (definido al iniciar el *logger*), donde irá almacenando lo que reciba del *main*. Cuando el espacio se acabe dejará de guardar información, mostrando una alerta en la consola. Solamente se escribirá a disco al finalizar la ejecución del *main*. Los logs se guardarán, por defecto, a `/media/sda1/`, una memoria flash comercial. Si no se dispone de una memoria flash se puede modificar los argumentos de arranque del *main* para que guarde en otro lado. Esto se implementó para evitar la escritura

F. Configuración y compilación del código

a la tarjeta *microSD*, que es muy lenta y hace inaceptable la performance del *main*⁴.

F.2. Debugging

Para probar el *main* sin volar el cuadricóptero, puede resultar cómodo apagar el control de conectividad. Para ello, setear `CHECK_NET_BYPASS` a 1 en

```
src/common/uquad_config.h
```

ADVERTENCIA: **NO** es recomendable hacer esto para una prueba con los motores prendidos, ya que la pérdida conexión implicaría la pérdida del control sobre el cuadricóptero.

F.2.1. Ejecución a partir de un log

Para debuggear el *main*, se lo puede correr con un log conocido como entrada, lo cual permite analizar el efecto de cambios concretos. Para esto se debe configurar el módulo que lee de la IMU, avisándole que debe leer de un log. Esto se logra seteando `IMU_COMM_FAKE` a 1, en `src/imu/imu_comm.h`.

C Vs. MatLab

Existe un script en *MatLab* que reproduce el comportamiento del *main*. Es muy útil para hacer pruebas y verificar el correcto funcionamiento del código en C:

```
kalman/kalman_main.m
```

Como argumento toma un log de datos crudos de la imu (*imu_raw.log*), como los generados por el *main* o por *imu_comm_test*:

```
src/test/imu_comm_test/imu_comm_test.c
```

F.3. Kernel

La Beagleboard corre linux *2.6.37*, de la distribución *Angstrom*:

- <http://www.angstrom-distribution.org/>

El kernel que viene por defecto es suficiente para *casi* todo. Solo hace falta configurar el puerto *i2c-2* para que trabaje a 333kHz (en lugar de 400kHz).

Para compilar el kernel se utilizó *Bitbake+OpenEmbedded*, y se lo compiló desde *Ubuntu 11.10* (64bits).

La información que se presenta a continuación se obtuvo de [19], [20] y del canal IRC `#oe`.

⁴Dado que la versión actual del *logger* no accede a disco hasta que termina de ejecutarse el *main*, no debería ser un problema guardar a la *microSD*. Esto no ha sido probado.

F.3.1. Compilación: Bitbake + OpenEmbedded

Para poder compilar programas para la Beagleboard se puede usar un entorno de desarrollo como OpenEmbedded (de ahora en más *OE*) y la herramienta para compilar, *bitbake*. La herramienta *bitbake* maneja recetas que listan programas y sus dependencias, y se encarga de compilar las cosas en el orden apropiado.

Todo lo que usa *OE* se baja de internet mediante *git*. A veces hay problemas con los servidores, y es cuestión de probar nuevamente en otro momento (o cambiar de servidor, revisar proxy, etc). Compilar una imagen entera, como para una SD, no es fácil, lleva tiempo, requiere que todos los servidores funcionen y un poco de suerte.

Para compilar en *Ubuntu*:

1. Ejecutar `sudo dpkg-reconfigure dash` y en el menu elegir la opción *no*. Por más información ver
<http://wiki.openembedded.org/index.php/OEandYourDistro>.
2. Descargar, configurar y actualizar el repositorio:

```
git clone git://git.angstrom-distribution.org/setup-scripts
cd startup-scripts
MACHINE=Beagleboard ./oebb.sh config Beagleboard
MACHINE=Beagleboard ./oebb.sh update
source ~/.oe/environment-oecore
```

3. Si se quisiera compilar ahora⁵, hacer:

```
bitbake console-image
```

El script `~/.oe/environment-oecore` es responsable de generar variables que se utilizan durante la compilación. Cada vez que se abre una consola se cargan las variables globales y las declaradas en `~/.bashrc`, una almacena las rutas a los ejecutables instalados, otra las bibliotecas, etc. Las variables en `~/.oe/environment-oecore` hay que cargarlas cada vez que se abre una nueva consola. Una de las cosas que hace el script es indicar la ruta al programa *bitbake*. Para verificar que el script fue correctamente ejecutado se puede escribir el comienzo del comando `bit` y apretar *tab* para ver sugerencias. Entre las opciones debería aparecer el comando *bitbake*.

Las recetas que utiliza *bitbake* (y nos interesan más) se encuentran en:

```
setup_scripts/sources/meta-ti/
```

Algunos ejemplos:

- Configuración del kernel:

```
setup-scripts/sources/meta-ti/recipes-kernel/\
linux/linux-3.0/Beagleboard/defconfig
```

- Receta para el kernel:

⁵Compilar la imagen lleva mucho tiempo, mejor configurar todo antes de compilar.

F. Configuración y compilación del código

```
setup-scripts/sources/meta-ti/recipes-kernel/\
linux/linux-omap_3.0.bb
```

- Receta para el u-boot:

```
setup-scripts/sources/meta-ti/recipes-bsp/\
u-boot/u-boot_2011.12.bb
```

Luego de haber incorporado las variables de `~/.oe/environment-oecore` ya no es necesario usar `MACHINE=Beagleboard ./oebb.sh`, se puede y debe usar directamente `bitbake`.

Es recomendable hacer `MACHINE=Beagleboard ./oebb.sh update` frecuentemente. Algunos paquetes necesarios para poder compilar correctamente (pueden faltar otros):

```
sudo apt-get install\
    python-ply python-progressbar\
    texi2html cvs subversion gawk\
    chrpath texinfo diffstat
```

Como modificar el contenido de OE

Para modificar un programa, como por ejemplo el *u-boot*:

```
bitbake -c devshell u-boot
# se abre una consola en el dir temporal del u-boot
emacs board/ti/beagle/beagle.h
# editar, por ejemplo habilitar la UART2
git add board/ti/beagle/beagle.h
git commit -m 'uquad: habilitando UART2'
git format-patch HEAD~1
cp 001-uquad:-habilitando-UART2.patch $OE_BASE/
# incrementar la linea que dice "PR = "r4"", ponerle r5
bitbake u-boot
# buscar el resultado en setup-scripts/build/tmp*/deploy/images
```

Para modificar el kernel y setear el *i2c-2* a 333kHz:

```
bitbake -c devshell virtual/kernel
# se abre una consola en el dir temporal del kernel
# ANTES de cambiar nada, hacer:
quilt new uquad-set-i2c-2-333kHz.patch
# Si se quiere hacer cambios en board-omap3beagle.c:
quilt add arch/arm/mach-omap2/board-omap3beagle.c
emacs arch/arm/mach-omap2/board-omap3beagle.c
# editar, por ejemplo setear i2c a 333kHz
# Ahora pedirle a quilt que arme un patch
quilt refresh
# El patch queda en patches/uquad-set-i2c-2-333kHz.patch
# Se copia a meta-ti/recipes-kernel/linux-3.0/
```

```
# Se edita meta-ti/recipes-kernel/linux_3.0.bb, agregando una
# línea antes de la q dice defconfig:
file://uquad-set-i2c-2-333kHz.patch;patch=1 \
# Ahora se compila haciendo
bitbake virtual/kernel
```

Comandos útiles - *bitbake+OE*

Algunos comando que pueden ser de utilidad:

- Para compilar un paquete individual, sin tomar en cuenta las dependencias:

```
bitbake -b receta.bb
```

Ejemplo:

```
bitbake -b sources/meta-ti/recipes-bsp/u-boot/u-boot_2011.12.bb
```

- Para ver todas las recetas que se ejecutan como dependencias, hacer:

```
bitbake <receta> -g
```

y luego mirar en `task-depends.dot`.

Ejemplo:

```
bitbake console-base-image -g
```

- Para borrar todo lo compilado sobre un paquete, por ejemplo el u-boot, hacer:

```
bitbake -c clean u-boot
```

F.4. IMU

La Mongoose viene cargada con un *bootloader Arduino*, que permite bajarle código mediante el puerto serie.

F.4.1. Compilación

Para programar la Mongoose se utiliza *Arduino*. Para instalarlo hacer:

```
sudo apt-get install arduino
```

El código utiliza una biblioteca que no viene con *Arduino*. Asumiendo que el repositorio git fue descargado a `~/uquad`, para agregar la biblioteca al *Arduino*, hacer:

```
sudo ln -s ${HOME}/uquad/src/mongoose_fw/HMC58X3/\
/usr/share/arduino/libraries/HMC58X3
```

F.5. Trabajo a futuro

- **Path planner:** Solamente está implementada la modalidad de *hovering*. Queda pendiente implementar rectas y círculos, y un sistema de *waypoints* que permita ir recorriendo trayectorias.

- **Control:** La matriz de control se carga de un archivo de texto. Para automatizar el cálculo de dicha matriz, permitiendo cambiar de trayectoria en tiempo real, habría que calcularla en C.

En `src/control/control.c` está lista una implementación del algoritmo LQR, y de la linealización del sistema según la trayectoria elegida. Se verificó el correcto funcionamiento de la implementación comparando contra la versión en *MatLab*, pero no se lo incorporó al *main*.

Un tema a considerar es el tiempo de computo, varios segundos según pruebas en una PC. Esto sería inaceptable durante el vuelo. Una solución simple sería hacer las cuentas de a poco, evitando pasar varios segundos sin ejecutar acciones de control.

- **Visión:** Queda pendiente implementar en C el algoritmo descrito en A.

- **Logger:** Una mejora sería que el *logger* guarde información en binario en lugar de usar ASCII, lo que permitiría ahorrar RAM. De cualquier forma, el RAM no es un problema por el momento.

ANEXO G

HARDWARE - CUIDADOS

G.1. Baterías del Cuadricóptero (LiPo)

Las baterías que se utilizan para el cuadricóptero son baterías de polímero de litio (LiPo). Pueden resultar peligrosas si se usan de manera inapropiada. A continuación se resumen algunos de los puntos más importantes:

- Nunca dejar cargando sin supervisión (no dejar cargando de noche).
- Cargar solamente utilizando cargadores balanceados, como los que fueron adquiridos por el proyecto. Recargar una batería puede llevar entre 4 y 6 horas.
- No dejar que se descargue completamente.
 - Si los motores comienzan a reducir su velocidad, es probable que la batería se esté acabando. Reemplazarla.
 - Verificar que la batería quede desconectada de los ESCs al terminar de utilizar el cuadricóptero para evitar que se descarguen completamente.
- Si se observa que la batería se ha inchado dejar de usar inmediatamente, y deshacerse de ella.

Por más información, los foros de *RCGroups* pueden resultar útiles, por ejemplo en <http://www.rcgroups.com/forums/showthread.php?t=209187>.

G.2. Baterías de la electrónica - Beaglejuice

La batería que alimenta la electrónica (*Beaglejuice*) fue comprada a la empresa *Liquidware*. Algunas cosas a tomar en cuenta:

- Recargar una *Beaglejuice* completamente lleva aproximadamente 12 horas, y el tiempo útil (usando *WiFi* y *GPS*) es de no más de 2 horas. No conviene dejarla cargando de noche, pero se ha hecho muchas veces, debido al tiempo que tardan en cargarse.

G. Hardware - Cuidados

- No hay documentación sobre estas baterías. Encienden 2 LEDs blancos cuando están prendidas, uno naranja cuando están cargando, y a veces otro LED naranja (de manera intermitente) durante el proceso de carga (no se sabe que significa, y no ha dado problemas).

G.3. Motores y Hélices

G.3.1. Hélices

Antes de realizar cualquier prueba, revisar que las hélices estén bien ajustadas. Se aflojan con el uso, y pueden llegar a salirse.

Se pueden seguir usando si reciben golpes leves, como raspones, etc, pero no conviene usarlas si tienen alguna parte quebrada, ya que se les exige mucho durante el vuelo, y pueden partirse.

Si se desea medir la velocidad de las hélices con el dispositivo de la sección 6.3.1, usar lentes de protección, por las dudas que alguna parte de una hélice salga despedida por accidentarse contra el dispositivo.

G.3.2. Motores

Se han hecho muchas pruebas usando cuerdas para tener más control sobre el cuadricóptero. En caso de repetirse este tipo de pruebas, verificar que no haya riesgo de enredar cuerdas con motores, ya que este tipo de exigencia ha demostrado ser capaz de dañar permanentemente un motor.

Los motores tienen imanes, y por lo tanto atraen restos ferromagnéticos que se encuentren cerca de ellos. Evitar exponer los motores a este tipo de condiciones, ya que no hay nada que evite el ingreso de partículas ferromagnéticas a las partes móviles de los motores, y podrían afectar su performance.

La aislación de los cables que van de los ESCs a los motores ha demostrado ser de mala calidad. Un cortocircuito a esa altura causó un accidente. Se ha reforzado la aislación entre fases para evitar problemas. Asimismo se comprobó que existen errores en el diseño de los ESCs en lo que respecta a la disipación térmica. Es recomendable realizar pruebas por períodos de tiempo reducidos (inferiores a los 5 minutos) y dejar enfriar los transistores de potencia.

ANEXO H

MANUAL DE USUARIO

Este manual de usuario indica los pasos a seguir para volar el cuadricóptero en modo hovering. Se asume que la IMU se encuentra previamente calibrada.

H.1. Montaje y conexionado

Componentes del sistema

Montar sobre la plataforma los distintos componentes del sistema:

- BeagleBoard, BeagleJuice y módulo de WiFi en la caja de acrílico como se detalla en el capítulo 17.
- IMU
- ESC
- GPS
- Placa de conversión de niveles lógicos
- Batería

Si se desea habilitar la opción de utilizar el control remoto debe incluirse la CPU de fábrica, el receptor y el circuito encargado del switcheo (ver capítulo 20).

Conexiones

En esta sección se detallan las conexiones de los componentes.

- Conexiones de la BeagleBoard
 - Alimentar la BeagleBoard con 5v provenientes de la BeagleJuice.
 - Conectar en un puerto USB el módulo de WiFi, en otro el GPS y en un tercero un pendrive.
 - Conectar con la placa de conversión de niveles lógicos, basándose en los nombres de las señales indicados sobre la placa, y la página 108 del BeagleBoard-xM System Reference Manual.

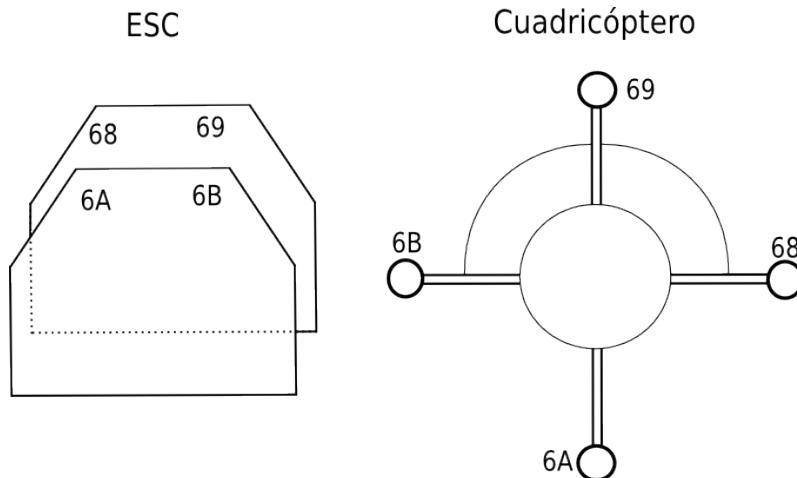


Figura H.1.: Correspondencia entre motores y ESC

- Conexiones de la IMU
 - Conectar la alimentación (5V) a la BeagleJuice.
 - Conectar las señales de Gnd, 3.3V, Tx y Rx a la placa de conversión de niveles lógicos.
- Conexión de los ESCs
 - Conectar las líneas del I^2C (SDA, SCLK, GND) a la placa de conversión de niveles lógicos.
 - Conectar a los motores. En la figura H.1 puede observarse la correspondencia entre ESC y motores.
 - Conectar la batería

En caso de desearse utilizar también el control remoto, conectar la CPU de fábrica y el circuito de Switcheo de acuerdo a las conexiones que se especifican en el capítulo 20.

Se debe hacer coincidir el plano de la IMU con el plano formado por los motores de acuerdo al procedimiento explicado en el capítulo 17. Para mostrar en consola las lecturas de la IMU en deben ejecutarse los siguiente comandos:

```
ssh root@10.42.43.2
cd work/uquad/src/build/test imu_comm_test
./imu_comm_test /dev/ttyS1
```

Las columnas 10 a 12 indican los ángulos de Euler. Se deben ajustar los tornillos de la IMU hasta que dichos ángulos sean lo más cercano posible al cero.

H.2. Ajuste del Offset

Para resolver el problema del offset se ejectura el driver de los motores y se le envían comando desde la consola, hasta lograr equilibrar el ángulo en cuestión. Para correr el driver ejecutar los siguientes comandos:

```
ssh root@10.42.43.2
cd work/uquad/src/i2c_beagle
make stdin
./cmdstdin X69 X6A X6B X68
```

Cada uno de los parámetros **X6M** puede ser 1 o 0: El 1 habilita el motor **6M**, y el 0 lo deshabilita.

- Offset del ángulo de Yaw

1. Colgar el cuadricóptero de forma que pueda girar respecto al eje vertical.
2. Ejecutar

```
./cmdstdin 1 1 1 1
```

a una velocidad de I^2C lo más alta posible sin que el cuadricóptero se levante (Se sugiere utilizar 120 I^2C trabajando con un peso de aproximadamente 1.7kg).

3. Ajustar los offsets en la consola, enviando distintos valores hasta que el cuadricóptero no gire, y luego escribir el resultado en

```
src/i2c_beagle/cmd_motores.c
```

para que sea tomado en cuenta siempre que se compile el driver.

- Offset del ángulo de Pitch y de Roll: Los ajustes en estos ángulos no deberían modificar mucho el offset encontrado para el Yaw, ya que el torque neto no se modifica. De cualquier forma, puede cambiar el punto de operación y por lo tanto afectar al Yaw.

1. Ubicar el cuadricóptero de forma que tenga libre un solo eje de giro.
2. Para el Pitch ejecutar:

```
./cmdstdin 1 1 0 0
```

Igual que en el caso anterior ajustar el offset de forma que el cuadricóptero se mantenga horizontal.

3. Repetir para el Roll ejecutando:

```
./cmdstdin 0 0 1 1
```

- Revisar los tres ángulos para verificar los offsets obtenidos.

H.3. Pruebas del Controlador

- Prueba de los ángulos de Roll y de Pitch

1. Ubicar el cuadricóptero de forma que tenga libre un solo eje de giro posible.
2. Modificar las matrices **K_prop_full_ppzt.txt** y **K_int_full_ppzt.txt** de forma que las únicas entradas distintas de cero sean las que corresponden a la realimentación del ángulo de interés, de la velocidad angular de interés y de la integral del error del ángulo de interés. Las matrices de realimentación se encuentran en **src/control**.

H. Manual de usuario

3. Modificar el driver `src/i2c_beagle/cmd_motores.c`, ya que por defecto prender todos los motores. Descomentar las líneas que inhabilitan los motores que no se van a utilizar (buscar cerca de la línea 697).
 4. Correr el loop de control (`./go.sh`), modificar dichas matrices hasta encontrarse conforme con la respuesta del sistema. Partir de los valores dados por el algoritmo LQR en MatLab.
 5. Repetir con el otro ángulo (cambiar nuevamente las matrices y el driver).
- Prueba del ángulo de Yaw.
 1. Colgar el cuadricóptero.
 2. Setear la Masa del cuadricóptero lo más alto posible sin que el cuadricóptero levante vuelo (se sugiere utilizar 1.4Kg si el cuadricóptero levanta vuelo con 1.7kg).
 3. Modificar las matrices `K_prop_full_ppzt.txt` y `K_int_full_ppzt.txt` hasta lograr el comportamiento deseado. Partir de los valores dados por el algoritmo LQR.

Una vez realizadas las pruebas anteriores el cuadricóptero debería presentar un comportamiento estable. De todas formas se sugiere agregar dos cuerdas de seguridad: una por encima y otra por debajo, de forma de limitar el movimiento del mismo y evitar accidentes. Tener especial con las cuerdas, en caso de enredarse con un motor cortar la ejecución de inmediato para evitar dañar ESCs o motores.

NOTA: Los ESCs disponibles presentan un problema conocido como FOD¹ que hace que alguno de los motores deje de funcionar. Es un evento aleatorio, respecto a cuando sucede y respecto a el motor que afecta. Evitar hacer pruebas muy largas o muchas pruebas consecutivas para reducir los riesgos².

¹Flip Of Death

²No fue comprobado que esto evite el FOD.

ANEXO I

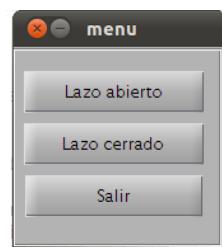
MANUAL DE USUARIO DEL SIMULADOR

En este anexo se explica como utilizar el simulador implementado.

Abriendo el programa

1. Abrir el programa *MatLab*.
2. Ubicarse en la carpeta uquad.
3. Desde la consola correr el script *startup*. Este script agrega al *path* las carpetas necesarias para que el simulador funcione adecuadamente.
4. Desde la consola correr el simulador ejecutando *menu*. Un menú como el que se muestra en la figura I.1 debería aparecer.

Dicho menú ofrece la posibilidad de realizar simulaciones tanto en lazo abierto como en lazo cerrado haciendo “click” en la opción deseada.



Simulaciones en lazo abierto

Luego de hacer “click” en la opción de lazo abierto en el menú principal se despliega el menú de la figura I.2.

Figura I.1.: Menú principal

Estableciendo las condiciones iniciales de la simulación

Para establecer las condiciones iniciales de la simulación se escribe el valor deseado para cada variable en el casillero correspondiente en la parte superior izquierda del menú. Las variables que se pueden establecer son:

I. Manual de usuario del simulador

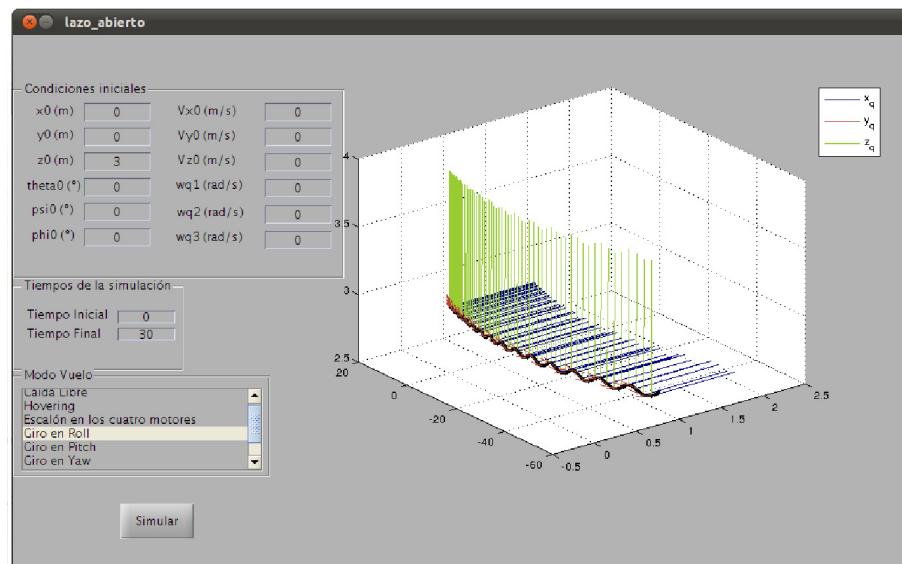


Figura I.2.: Interfaz del lazo abierto

- La posición inicial del cuadricóptero.
- La orientación inicial (Ángulos de Euler) del cuadricóptero.
- La velocidad inicial del cuadricóptero en el sistema inercial.
- La velocidad angular del cuadricóptero.

En la figura I.2 se realizó una simulación con todas las condiciones iniciales nulas excepto la altura (z) que se fijó a tres metros.

Estableciendo los tiempos de la simulación

La interfaz permite indicar los tiempos iniciales y finales de la simulación en el submenú que se encuentra al centro a la izquierda. En la figura se observa que para la simulación en cuestión se estableció un tiempo inicial de 0 segundos y un tiempo final de 30 segundos.

Estableciendo el modo de vuelo

Con el submenú que se encuentra en la parte inferior se define el tipo de simulación que se desea realizar. El usuario puede elegir el modo de vuelo de una lista pre establecida, los modos de vuelo disponibles son:

- Caída libre
- Hovering
- Escalón en los cuatro motores
- Giro en Roll
- Giro en Pitch
- Giro en Yaw

Realizando la simulación

Una vez establecidas las condiciones iniciales de la simulación, los tiempos iniciales y finales y el tipo de vuelo que se desea realizar se oprime el botón simular para iniciar la simulación.

Sobre la derecha de la interfaz gráfica se despliega la trayectoria realizada por el sistema y la evolución de los ejes principales del cuadricóptero: x_q , y_q y z_q en azul, rojo y verde respectivamente.

Para realizar un pos procesamiento de la información obtenida gracias a la simulación es importante tener en cuenta que el simulador devuelve un vector con los tiempos de la simulación y una matriz con las 12 variables de estado. La variable de los tiempos de la simulación toma el nombre **t** y la variable con las 12 variabes de estado toma el nombre **Y**. Cada columna de la matriz **Y** representa los distintos valores que toma cada variable de estado a lo largo del tiempo. El órden en el cual se guardan las variables de estado es:

- Northing (x)
- Westing (y)
- Altura (z)
- Roll (ψ)
- Pitch (φ)
- Yaw (θ)
- Velocidad en el sistema solidario al cuadricóptero según x_q (v_{qx})
- Velocidad en el sistema solidario al cuadricóptero según y_q (v_{qy})
- Velocidad en el sistema solidario al cuadricóptero según z_q (v_{qz})
- Velocidad angular del cuadricóptero según x_q (ω_{qx})
- Velocidad angular del cuadricóptero según y_q (ω_{qy})
- Velocidad angular del cuadricóptero según z_q (ω_{qz})

Simulaciones en lazo cerrado

Haciendo “click” en la opción Lazo cerrado en el menú principal se despliega la interfaz que se muestra en la figura I.3. Esta interfaz permite realizar simulaciones en lazo cerrado a fin de verificar el funcionamiento de los algoritmos de control.

Estableciendo las condiciones iniciales y tiempos de simulación

En el submenú que se encuentra en la esquina superior izquierda de la interfaz pueden establecerse las condiciones iniciales de la simulación al igual que en la simulación en lazo abierto. A la derecha de dicho submenú se pueden establecer el tiempo inicial y final de la simulación al igual que en la simulación del lazo abierto.

I. Manual de usuario del simulador

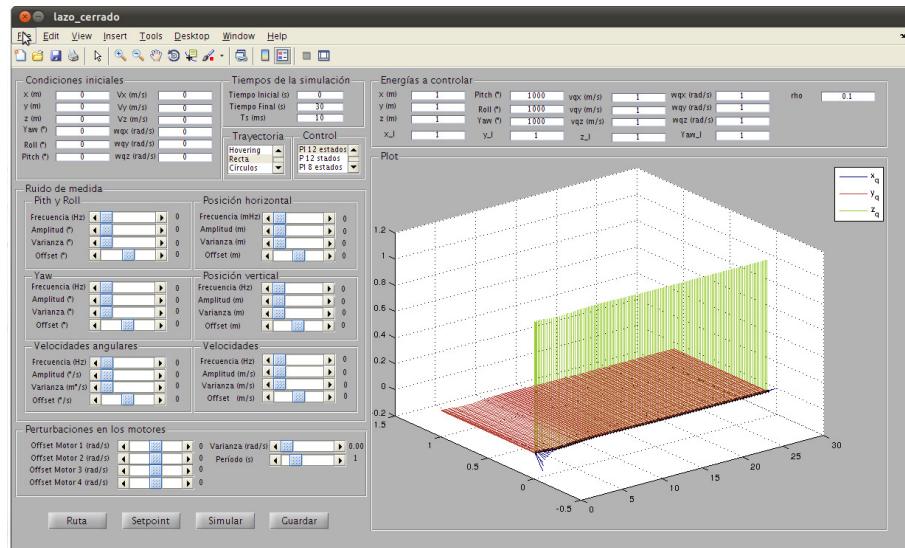


Figura I.3.: Interfaz del lazo cerrado

Trayectoria a controlar

En el panel Trayectoria puede elegirse el tipo de trayectoria a realizar. En esta versión del simulador se tiene implementada tres trayectorias que pueden elegirse desde el menú:

- Hovering.
- Vuelo en línea recta.
- Vuelo en círculos.

Además del tipo de trayectoria deben establecerse los parámetros con los cuales se desea realizar la trayectoria. Para esto, haga “click” en el botón *setpoint* de la interfaz de lazo cerrado. Una nueva ventana se abrirá ofreciendo casilleros para completar con las variables deseadas. Según el tipo de trayectoria escogida se pueden setear distintas variables:

- Hovering: Posición y ángulo de Yaw.
- Vuelo en línea recta: Velocidades en el sistema S_q y ángulo de Yaw.
- Vuelo en círculos: Velocidad angular según la vertical y velocidad tangencial.

Controlador

El simulador ofrece cuatro tipos de control posibles, seleccionables desde el panel *Control* de la interfaz de lazo cerrado. Estos son:

- P12: controlador proporcional de todas las variables de estado del sistema.
- PI12: controlador proporcional de todas las variables de estado del sistema y de las integrales del error en posición y en el ángulo de Yaw.
- P8: controlador proporcional de ocho de las doce variables de estado del sistema. Se deja por fuera la posición horizontal y las velocidades v_{qx} y v_{qy} .

- PI8: controlador idéntico al anterior con el agregado del control sobre la integral del error en los tres ángulos de euler y en la altura.

El control implementado es LQR. En el panel que se ubica en la esquina superior derecha se pueden variar los parámetros de las matrices Q y R utilizadas para determinar la matriz de realimentación. Se imponen dos restricciones para esta elección:

- Las matrices Q y R son diagonales
- Todas las entradas de la matriz R son idénticas

Simulación

Una vez establecidas las condiciones iniciales, los tiempos de la simulación, el tipo de trayectoria, el setpoint, el tipo de control y los parámetros de las matrices Q y R se puede proceder a simular el sistema en lazo cerrado haciendo “click” sobre el botón simular.

En la gráfica de la derecha se grafica, al igual que para el lazo abierto, la trayectoria obtenida y los ejes del cuadricóptero a lo largo del tiempo.

Se genera un vector **t** con los tiempos de la simulación y una matriz **Y** con la siguiente información:

- En las primeras doce columnas de la matriz **Y** se guardan las variables de estado del sistema en el mismo orden que para el lazo abierto.
- En las segundas doce columnas de la matriz **Y** se guardan los ruidos generados para las doce variables de estado en el mismo orden. (En las siguientes secciones se explica con mayor profundidad este aspecto).
- Las columnas 25 a 28 de la matriz **Y** corresponden a las velocidades angulares de los cuatro motores a lo largo del tiempo.
- Las columnas 29 a 40 corresponden al setpoint para cada variable de estado.

La matriz de realimentación generada puede ser guardada haciendo “click” sobre el botón guardar. La matriz se guardará en la carpeta `./src/control/` bajo los siguientes nombres:

- PI12:
 - K_prop_full.txt: matriz de realimentación proporcional.
 - K_int_full.txt: matriz de realimentación para la integral del error de la posición y el ángulo de Yaw.
- P12: K_full: matriz de realimentación proporcional.
- PI8:
 - K_prop.txt: matriz de realimentación proporcional de ocho estados.
 - K_int.txt: matriz de realimentación de la integral del error en los ángulos de Euler y la altura.
- P8: K.txt: matriz de realimentación proporcional de ocho estados.

Ruido de medida y no idealidades

El simulador ofrece la posibilidad de agregar ruido al estado que se utiliza para determinar la acción de control. Este ruido simula los ruidos introducidos por los sensores. Para cada variable de estado puede agregarse desde el panel *Ruido de medida* los siguientes ruidos:

- Componente sinusoidal. Puede ajustarse:
 - Amplitud
 - Frecuencia
- Ruido blanco: Puede ajustarse:
 - Varianza
 - Valor medio

Puede agregarse también en el panel *Perturbaciones en los motores offsets* distintos para cada motor, y un ruido blanco (puede configurarse su varianza y período) para simular el funcionamiento no ideal de los motores.

Ruta predeterminada

El simulador ofrece además la posibilidad de recibir *waypoints* por los cuales se desea pasar con una orientación determinada en un tiempo determinado. El simulador se encarga de concatenar las trayectorias a seguir para lograr el objetivo.

Todas las configuraciones descritas anteriormente son válidas para esta sección excepto que no tiene sentido definir una trayectoria ni los tiempos de simulación. Tampoco es posible guardar todas las matrices de realimentación utilizadas para lograr el objetivo planteado.

Para utilizar este modo:

1. Establecer las condiciones iniciales de la simulación, el controlador a utilizar, los ruidos y perturbaciones en los motores.
2. Hacer “click” sobre el botón ruta.
3. En la ventana que se despliega establecer la posición y la orientación del primer *waypoint* así como el tiempo en el cual se desea alcanzar dicho *waypoint*.
4. Hacer “click” sobre el botón *Agregar*.
5. Repetir los pasos 3 y 4 hasta agregar todos los puntos deseados.
6. Hacer “click” sobre el botón *Listo*.
7. Hacer “click” sobre el botón *Simular*.

Al igual que para la simulación de una trayectoria en particular se guardan las variables **t** y **Y** para su posterior análisis y estudio.

ANEXO J

COSTOS

En el presente anexo se detallan los costos asociados a una unidad de la plataforma diseñada.

Plataforma Física (*)	819.08 USD
Beagleboard	149.00 USD
Beaglejuice	88.73 USD
Mongoose 9DOF	115.00 USD
GPS	34.99 USD
WiFi	44.87 USD
Otros (**)	100.00 USD
Total	1351.67 USD

(*) Incluye transmisor, receptor y una batería

(**) Acrílico, cobre, componentes electrónicos, aluminio, etc.

Cuadro J.1.: Costos de las partes que componen el sistema

Cabe destacar que hoy en dia el costo de una plataforma similar podría verse reducido considerablemente.

- Es posible adquirir “esqueletos” de cuadricóptero a precios inferiores a los *20USD*¹.
- Si se elimina la posibilidad del vuelo a control remoto, se pueden adquirir ESCs, motores y hélices por *300USD*.
- Existen opciones de inteligencia que por *25USD* brindan la misma capacidad de procesamiento que la Beagleboard².

¹Ver www.himodel.com.

²Ver <http://www.raspberrypi.org>.

BIBLIOGRAFÍA

- [1] W. Brown, “Brushless dc motor control made easy,” tech. rep., Microchip Technology Inc., 2002.
- [2] M. Cavcar, “Blade element theory.” Anadolu University,School of Civil Aviation, Eskisehir, Turquia, 2004.
- [3] T. Bresciani, “Modeling, identification and control of a quadrotor helicopter,” Master’s thesis, Lund University, Octubre 2008.
- [4] M. Vendittelli, “Quadrotor modeling.” Curso: “Elective in robotics”, Sapienza Universitá Di Roma, Noviembre 2011.
- [5] I. Skog and P. Händel, “Calibration of a mems inertial measurement unit,” in *XVII Imeko World Congress, Metrology for a Sustainable Development*, (Rio de Janeiro), Setiembre 2006.
- [6] C. Konvalin, “Compensating for tilt, hard iron and soft iron effects,” Tech. Rep. MTD-0802, Memsense, Agosto 2008.
- [7] A. Barraud, “Magnetometers calibration.” <http://www.mathworks.com/matlabcentral/fileexchange/23398-magnetometers-calibration>, Marzo 2009.
- [8] Portland State Aerospace Society, *A quick derivation relating altitude to air pressure*, Diciembre 2004.
- [9] H. Zhao and Z. Wang, “Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion,” *Sensors Journal, IEEE*, vol. 12, pp. 943–953, Mayo 2012.
- [10] L. Tamás, G. Lazea, R. Robotin, C. Marcu, S. Herle, and Z. Szekely, “State estimation based on kalman filtering techniques in navigation,” in *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 147–152, IEEE, Mayo 2008.
- [11] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, vol. 1 of *Prentice Hall singal processing*. Pearson, 2 ed., 2011.

Bibliografía

- [12] B.-C. C. Chirag Jagadish, “Diversified redundancy in the measurement of euler angles using accelerometers and magnetometers,” in *46 th IEEE Conference on Decision and Control*, pp. 2669–2674, IEEE, Diciembre 2007.
- [13] R. Alonso, G. Hernández, and P. Iturralde, “Automatización del vuelo de un avión.” *Proyecto de Fin de Carrera de Ingeniería Eléctrica*, Facultad de Ingeniería, Universidad de la República, Abril 2009.
- [14] A. F. Soslashrensen, “Autonomous control of a miniature quadrotor following fast trajectories,” Master’s thesis, Aalborg University/U.C. Berkeley, 2009-2010.
- [15] M. Hakas, “Sistemas de control en tiempo discreto.” Curso: “Introducción a la Teoría de Control”, Facultad de Ingeniería Universidad de la República, 2005.
- [16] Wikipedia, “List of moment of inertia tensors.” http://en.wikipedia.org/wiki/List_of_moment_of_inertia_tensors.
- [17] Satellite Navigation and Positioning Lab - UNSW, *The Impact Of Satellite Geometry*, 2004.
- [18] Wikipedia, “Meaning of dop values.” http://en.wikipedia.org/wiki/Dilution_of_precision_%28GPS%29#Meaning_of_DOP_Values, 2012.
- [19] C. Brake, “Capture oe source changes.” <http://bec-systems.com/site/456/capture-oe-source-changes>.
- [20] SlimLogic, “Openembedded angstrom kernel work flow.” <http://www.slimlogic.co.uk/2011/05/openembeddedangstrom-kernel-workflow/>.