

Proyecto Fin de Carrera Ingeniería Aeronáutica

Modelado, simulación y control de un quadrotor para transporte de carga colgante. Extensión al caso tridimensional

Autor: Octavio Alfredo García Campos

Tutor: Manuel Vargas Villanueva

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017



Proyecto Fin de Carrera

Ingeniería Aeronáutica

Modelado, simulación y control de un quadrotor para transporte de carga colgante. Extensión al caso tridimensional

Autor:

Octavio Alfredo García Campos

Tutor:

Manuel Vargas Villanueva

Profesor Titular

Dep. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Carrera: Modelado, simulación y control de un quadrotor para transporte de carga colgante. Extensión al caso tridimensional

Autor: Octavio Alfredo García Campos
Tutor: Manuel Vargas Villanueva

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

El proyecto de final de carrera supone el fin de una etapa muy importante en mi vida. Atrás quedan años de mucho trabajo y esfuerzo, pero también momentos buenos que me han ayudado a crecer y mejorar como persona. No constituye una asignatura más, sino la puesta en práctica de muchos de los conocimientos adquiridos durante la carrera. Me ha permitido ampliar mi campo de visión sobre vehículos aéreos no tripulados y mejorar mis conocimientos en campos como la mecánica o el control. Llegar hasta aquí no hubiera sido posible sin las personas que me han estado apoyando en todo momento. Es por eso, que quiero dedicar mis más sinceros agradecimientos a esas personas.

Mis primeros agradecimientos van dedicados a mi familia, al esfuerzo realizado por mis padres para que pudiera estudiar la carrera de ingeniería aeronáutica y cumplir así mi sueño, aportándome siempre apoyo incondicional y moral. No se me olvida mi hermana, que ha tenido que aguantar mis quejas y cambios de humor durante todos estos años, pero siempre ha estado ahí para escucharme y darme buenos consejos.

También quiero agradecer la paciencia que ha tenido mi pareja, que ha visto cómo era imposible planificar algo conmigo, pero que ha estado siempre pendiente de mí. Sobre todo, agradecerle la paz y sosiego que me ha transmitido en los momentos de mayor nerviosismo, siempre sin dejar de creer en mí.

Darle las gracias a mis amigos, que han estado preocupados siempre por mis progresos y a la confianza depositada en mí. A mis compañeros de carrera, con los que he pasado algunos de los mejores momentos de mi vida y con los que he compartido largas jornadas de biblioteca. Ellos han supuesto la cara más amable entre tanto estudio, compartiendo sus conocimientos con el fin de superar juntos el objetivo marcado hace años.

Finalmente, quiero agradecer a mi tutor de proyecto por guiarme en esta difícil tarea. También a todos los profesores y tutores que me han aportado su experiencia en el campo de la ingeniería y, como no, a los compañeros de proyecto por compartir su sabiduría en el campo de UAV's.

Quiero dar las gracias una vez más, porque sin la ayuda de todas estas personas no hubiera sido posible.

Octavio Alfredo García Campos

Escuela Técnica Superior de Ingeniería

Sevilla, 2017

Resumen

El espectacular desarrollo de los vehículos aéreos no tripulados durante los últimos años y su aplicación a campos cada vez más numerosos, han hecho posible el conocimiento de los mismos por toda la sociedad actual. Constituyen un área de estudio e investigación creciente tanto en el ámbito académico como laboral. El siguiente documento trata de dar respuesta al problema de una aeronave con carga colgante, buscando un control seguro y estable de la misma.

Abstract

The unmanned aerial vehicles development during these last years and their application to increasingly several fields has enabled today's society to acquire knowledge on this type of vehicles. Unmanned aerial vehicles constitute an increasing study and research area in both academic and occupational spheres. This work tries to solve the problem of an aircraft with a hanging load. It intends to find a safe and stable control of the aircraft.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
1 Introducción	1
2 Modelado Matemático del Sistema Quadrotor más Carga	5
2.1 Fuerzas y Pares actuando sobre el Quadrotor	5
2.2 Hipótesis y Parámetros	6
2.3 Cinemática	7
2.4 Dinámica	11
2.5 Modelo Matricial	20
2.6 Ecuaciones del Quadrotor sin Carga	25
3 Linealización	27
3.1 Velocidad Nula	27
3.2 Velocidad Constante	37
4 Modo Stabilize	49
4.1 Control del Roll	50
4.2 Control del Pitch	58
4.3 Control del Yaw	65
5 Modo Alt-Hold	73

5.1	Esquema de Control para Alt-Hold	74
5.2	Análisis de la Respuesta en Alt-Hold	80
6	Modo Loiter	85
6.1	Esquema de Control para Loiter	85
6.2	Análisis de la Respuesta en Loiter	92
7	Conclusión Final	101
	<i>Índice de Figuras</i>	103
	<i>Índice de Tablas</i>	105
	<i>Índice de Códigos</i>	107
	<i>Bibliografía</i>	109

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
1 Introducción	1
2 Modelado Matemático del Sistema Quadrotor más Carga	5
2.1 Fuerzas y Pares actuando sobre el Quadrotor	5
2.2 Hipótesis y Parámetros	6
2.3 Cinemática	7
2.3.1 Cuerpo	7
2.3.2 Carga	10
2.4 Dinámica	11
2.4.1 Formulación de Newton	12
Equilibrio de Fuerzas	12
Balance de Momentos	13
Equilibrio de Momentos entorno al punto de unión	15
2.4.2 Formulación de Lagrange	16
Energía Cinética	16
Energía Potencial	17
Fuerzas Generalizadas	17
2.5 Modelo Matricial	20
2.5.1 Coordenadas Cuerpo	21
2.5.2 Coordenadas Carga	23
2.6 Ecuaciones del Quadrotor sin Carga	25
3 Linealización	27
3.1 Velocidad Nula	27
3.1.1 Coordenadas Cuerpo	27
Modelo Matricial	28
Transformada de Laplace	29
Expresión Linealizada en el Espacio de Estados	31
3.1.2 Coordenadas Carga	32
Modelo Matricial	33
Transformada de Laplace	34
Expresión Linealizada en el Espacio de Estados	35
3.2 Velocidad Constante	37
3.2.1 Velocidad Vertical Constante	37
Coordenadas Cuerpo	37

Coordenadas Carga	39
3.2.2 Velocidad Horizontal Constante	41
Coordenadas Cuerpo	41
Coordenadas Carga	43
4 Modo Stabilize	49
4.1 Control del Roll	50
4.1.1 Esquema de Control para Roll	50
4.1.2 Análisis de la Respuesta en Roll	55
4.2 Control del Pitch	58
4.2.1 Esquema de Control para Pitch	58
4.2.2 Análisis de la Respuesta en Pitch	62
4.3 Control del Yaw	65
4.3.1 Esquema de Control para Yaw	65
4.3.2 Análisis de la Respuesta en Yaw	68
5 Modo Alt-Hold	73
5.1 Esquema de Control para Alt-Hold	74
5.2 Análisis de la Respuesta en Alt-Hold	80
6 Modo Loiter	85
6.1 Esquema de Control para Loiter	85
6.2 Análisis de la Respuesta en Loiter	92
6.2.1 Control de " x_B "	92
6.2.2 Control de " y_B "	95
7 Conclusión Final	101
<i>Índice de Figuras</i>	103
<i>Índice de Tablas</i>	105
<i>Índice de Códigos</i>	107
<i>Bibliografía</i>	109

Notación

f_i^T	Empuje del rotor i (N)
k_{th}	Coefficiente de empuje rotor
k_{dr}	Coefficiente de resistencia rotor
c	Cociente entre los coeficientes de resistencia y empuje (k_{dr}/k_{th})
Ω_i	Velocidad de giro del rotor i (rad/s)
$\Omega_{i_{max}}$	Velocidad de giro máxima del rotor i (rad/s)
${}^{(B)}f^T$	Empuje total en coordenadas cuerpo (N)
${}^{(B)}\tau_{B_x}^T$	Par de alabeo en coordenadas cuerpo (N m)
${}^{(B)}\tau_{B_y}^T$	Par de cabeceo en coordenadas cuerpo (N m)
${}^{(B)}\tau_{B_z}^T$	Par de guiñada en coordenadas cuerpo (N m)
${}^{(B)}\tau$	Vector de fuerzas generalizadas en coordenadas cuerpo
a	Distancia de los rotores al c.g. del vehículo (m)
l	Longitud del cable (m)
g	Constante de gravedad (m/s^2)
C	Constante para pasar de radianes a centígrados
T	Tiempo de muestreo (s)
m_B	Masa del quadrotor (kg)
m_L	Masa de la carga suspendida (kg)
I_B^{xx}	Momento de Inercia en x , expresado en ejes cuerpo ($Kg \cdot m^2$)
I_B^{yy}	Momento de Inercia en y , expresado en ejes cuerpo ($Kg \cdot m^2$)
I_B^{zz}	Momento de Inercia en z , expresado en ejes cuerpo ($Kg \cdot m^2$)
$C_{B_{xy}}$	Coefficiente de resistencia aerodinámica en dirección horizontal ($N/(m/s)^2$)
C_{B_z}	Coefficiente de resistencia aerodinámica en dirección vertical ($N/(m/s)^2$)
C_L	Coefficiente de resistencia aerodinámica de la carga ($N/(m/s)^2$)
b	Coefficiente de fricción en la unión ($N \cdot m/(rad \cdot s)$)
p_B	Posición del quadrotor en ejes inerciales (m)
$\dot{p}_B$	Velocidad del quadrotor en ejes inerciales (m/s)
$\ddot{p}_B$	Aceleración del quadrotor en ejes inerciales (m/s^2)
ϕ_B	Ángulo Roll del cuerpo en ejes inerciales ($^\circ$)
θ_B	Ángulo Pitch del cuerpo en ejes inerciales ($^\circ$)
ψ_B	Ángulo Yaw del cuerpo en ejes inerciales ($^\circ$)
p	Velocidad angular Roll expresada en el cuerpo ($^\circ/s$)
q	Velocidad angular Pitch expresada en el cuerpo ($^\circ/s$)
r	Velocidad angular Yaw expresada en el cuerpo ($^\circ/s$)
p_L	Posición de la carga en ejes inerciales (m)

$\dot{p}_L$	Velocidad de la carga en ejes inerciales (m/s)
$\ddot{p}_L$	Aceleración de la carga en ejes inerciales (m/s^2)
ϕ_L	Ángulo Roll de la carga en ejes inerciales ($^\circ$)
θ_L	Ángulo Pitch de la carga en ejes inerciales ($^\circ$)
${}^B R_I$	Transforma un vector de ejes inerciales a ejes cuerpo
${}^I R_B$	Transforma un vector de ejes cuerpo a ejes inerciales
$\dot{\eta}_B$	Variaciones angulares de los ángulos de Tait-Bryan
${}^{(B)}\omega_B$	Velocidad angular expresada en ejes cuerpo
$W(\eta_B)$	Transforma las variaciones de Tait-Bryan a velocidad angular en ejes cuerpo
${}^L R_I$	Transforma un vector de ejes inerciales a ejes carga
${}^{(L)}p_L$	Posición de la carga respecto al cuerpo en coordenadas de la carga
${}^{(L)}p_B$	Posición del cuerpo respecto a la carga en coordenadas de la carga
${}^{(B)}I_B$	Tensor de inercia expresado en el propio cuerpo
L	Lagrangiano del sistema
q_B	Vector coordenadas generalizadas cuerpo
q_L	Vector coordenadas generalizadas carga
Q	Vector de fuerzas y pares generalizados
v	Función de disipación Rayleigh
f_B^T	Vector fuerzas generalizadas en ejes tierra
τ_B^T	Vector de pares generalizados en ejes tierra
$WI(\eta_B)$	Matriz para transformar los momentos de ejes cuerpo a tierra
M	Matriz de masa o inercia
v	Matriz de fuerzas centrífugas o de coriolis
D	Matriz de resistencia
$g(q)$	Vector de fuerzas gravitatorias
$\mathcal{L}$	Transformada de Laplace
x	Vector de estado ampliado en el modelo LTI
u	Vector de actuación en el modelo LTI
y	Vector de salida en el modelo LTI
A	Matriz de derivadas parciales del vector de estado en el modelo LTI
B	Matriz de derivadas parciales del vector de actuación en el modelo LTI
C	Matriz del sistema LTI
$ANGLE_MAX$	Ángulo máximo permitido para Roll, Pitch o Yaw ($^\circ$)
$roll_angle_ef$	Ángulo deseado en Roll o de entrada ($^\circ$)
$pitch_angle_ef$	Ángulo deseado en Pitch o de entrada ($^\circ$)
yaw_rate_ef	Velocidad deseada en Yaw o de entrada ($^\circ/s$)
$rate_ef_desired$	Velocidad angular prealimentada en ejes tierra ($^\circ/s$)
$angle_ef_error$	Error en ángulo en ejes tierra ($^\circ$)
$angle_ef_target$	Ángulo objetivo en ejes tierra ($^\circ$)
$rate_bf_desired$	Velocidad angular prealimentada en ejes cuerpo ($^\circ/s$)
$angle_bf_error$	Error en ángulo en ejes cuerpo ($^\circ$)
$ACRO_YAW_P$	Ganancia en Yaw para controlar la velocidad de giro
STB_RLL_P	Ganancia proporcional en ángulo para Roll
STB_PIT_P	Ganancia proporcional en ángulo para Pitch
STB_YAW_P	Ganancia proporcional en ángulo para Yaw
$rate_bf_target$	Velocidad angular objetivo en ejes cuerpo
$salida_roll$	Salida del controlador en Roll
$salida_pitch$	Salida del controlador en Pitch
$salida_yaw$	Salida del controlador en Yaw
$RATE_RLL_P$	Ganancia proporcional en velocidad para Roll
$RATE_RLL_I$	Ganancia integral en velocidad para Roll

<i>RATE_RLL_D</i>	Ganancia derivativa en velocidad para Roll
<i>RATE_PIT_P</i>	Ganancia proporcional en velocidad para Pitch
<i>RATE_PIT_I</i>	Ganancia integral en velocidad para Pitch
<i>RATE_PIT_D</i>	Ganancia derivativa en velocidad para Pitch
<i>RATE_YAW_P</i>	Ganancia proporcional en velocidad para Yaw
<i>RATE_YAW_I</i>	Ganancia integral en velocidad para Yaw
<i>RATE_YAW_D</i>	Ganancia derivativa en velocidad para Yaw
<i>vel_desired</i>	Velocidad lineal deseada en x, y, z (m/s)
<i>pos_target</i>	Posición objetivo en x, y, z (m)
<i>POS_Z_P</i>	Ganancia proporcional en posición para z
<i>VEL_Z_P</i>	Ganancia proporcional en velocidad para z
<i>vel_target</i>	Velocidad lineal objetivo en x, y, z (m/s)
<i>accel_target</i>	Aceleración lineal objetivo en x, y, z (m/s^2)
<i>ACCEL_Z_P</i>	Ganancia proporcional en aceleración para z
<i>ACCEL_Z_I</i>	Ganancia integral en aceleración para z
<i>ACCEL_Z_D</i>	Ganancia derivativa en aceleración para z
<i>thr_out</i>	Salida del controlador de altitud
<i>POS_XY_P</i>	Ganancia proporcional en posición para x, y
<i>VEL_XY_P</i>	Ganancia proporcional en velocidad para x, y
<i>VEL_XY_I</i>	Ganancia integral en velocidad para x, y
<i>pitch_target</i>	Ángulo Pitch a la salida del Loiter ($^{\circ}$)
<i>roll_target</i>	Ángulo Roll a la salida del Loiter ($^{\circ}$)

1 Introducción

El crecimiento espectacular de los vehículos aéreos no tripulados (UAV's) en los últimos años ha sido posible gracias al desarrollo de nuevas tecnologías que se adaptan mejor a los requerimientos actuales, [10]. Baterías más ligeras y con mayor autonomía de vuelo o equipos embarcados más compactos y con mejores prestaciones, como las IMU's electrónicas o los microactuadores, hacen posible su auge actual. Esta posibilidad de incluir nuevos dispositivos junto con su elevada maniobrabilidad y su capacidad para mantenerse fijos en el aire, lo hacen ideal para un amplio abanico de tareas. Una de sus mayores ventajas reside en la posibilidad de despegar y aterrizar verticalmente pudiendo operar en todo tipo de terrenos.

Muchos son los estudios que se han llevado a cabo sobre vehículos aéreos no tripulados, de esta forma hay drones que realizan tareas de vigilancia y seguimiento, y que operan tanto en zonas costeras y rurales como bélicas. Incluso cabe la posibilidad de que puedan incluir elementos externos como armamento en lugares de conflicto.



Figura 1.1 Drone con cámara incorporada para vigilancia tomada de [28].

Otras de las muchas posibilidades que ofrecen, es la capacidad de llegar a lugares inaccesibles para el hombre o zonas temporalmente peligrosas. Son los primeros que pueden adentrarse en lugares donde se ha producido un desastre natural o humano, como son incendios forestales o avalanchas, ofreciendo una primera visión de la zona al personal destinado para tal misión.

Dado su amplio espectro se busca también implementarlos en proyectos futuros que faciliten la vida al ser humano, como sería su incorporación en tareas relacionadas con la agricultura y el medio ambiente, bien para fumigar cultivos o prevenir incendios. Igualmente, podrían trabajar en la reparación de tendidos

eléctricos o infraestructuras de gran altura a través de brazos mecánicos o aportando el material necesario a los operarios que se encuentren realizando tales tareas, [14].

Finalmente, se plantea su utilización para el transporte de mercancías o pedidos en extensas áreas de población dispersa, aliviando el tráfico por carretera y haciéndolo de una forma más rápida y eficiente. Sin embargo, el hecho de transportar una carga colgante plantea serios inconvenientes como son la oscilación de la misma, que puede desestabilizar al vehículo e incluso hacerlo caer. Para evitarlo, es necesario desarrollar un control robusto que recupere la estabilidad del vehículo rápidamente ante inestabilidades provocadas por la carga. Este área de investigación ha sido tratada ampliamente por diferentes autores en el ámbito académico, como son el diseño de algoritmos de control antibalaceo [35], el transporte de carga mediante varios helicópteros pequeños [6] o el seguimiento de trayectorias en balanceo libre [30]. Y en el sector privado, empresas como Google o Amazon, se han lanzado a desarrollar sus propios vehículos con el fin de poder transportar sus mercancías. Este prototipo de vehículo se encuentra en la Figura 1.2.



Figura 1.2 Drone Amazon para transporte de carga tomado de [26].

En este contexto, se justifica el estudio de las ecuaciones que rigen el comportamiento del sistema formado por el quadrotor más la carga colgante. La dinámica de la carga influye en gran medida en las condiciones de vuelo del vehículo y es necesario tenerla en cuenta, bien como un parámetro incierto o bien como una perturbación externa. Este trabajo se centra en la determinación de las ecuaciones que rigen el comportamiento del sistema en tres dimensiones, y se expresarán tanto en coordenadas del vehículo como en coordenadas de la carga. Posteriormente, se linealizan las ecuaciones dinámicas obtenidas considerando los casos de velocidad nula y velocidad lineal constante.

En principio, este trabajo se planteó como una continuación del trabajo de fin de master realizado por el alumno Fränz Bahner [5], cuyo objetivo era extender el control óptimo desarrollado para dos dimensiones al caso tridimensional. Sin embargo, debido a la adquisición de un nuevo quadrotor por parte del Departamento de Ingeniería de Sistemas y Automática a través de Erle Robotics, se vio la posibilidad de implementar los esquemas de control incluidos en el modelo de Erle-Copter en Gazebo a las ecuaciones dinámicas aquí desarrolladas, [17]. El objetivo era obtener un modelo en MATLAB® SIMULINK® que utilizará el mismo esquema de controladores PID para compararlo con el modelo en Gazebo de Erle-Copter. Esto permite estudiar cómo se realiza el control de la actitud y de la posición del drone a través de los modos Stabilize, Alt-Hold y Loiter. También proporciona un modelo matemático sobre el que poder realizar ensayos antes de aplicar sus resultados al quadrotor real, pues el fin último sería poder disponer de un vehículo estable para el transporte de una carga desde un punto A a un punto B.

Teniendo en mente tal objetivo, se establece el control de las ecuaciones del quadrotor empleando para ello los modos Stabilize, Alt-Hold y Loiter recogidos en [4], [2] y [15]. La extensión al sistema formado por el quadrotor más la carga se deja para futuros proyectos, incluido el estudio de otros modos de vuelo como son el Acro o Circle.

En definitiva, los vehículos aéreos no tripulados (UAV's) ofrecen hoy en día una amplia variedad de campos de estudios donde poder emplearlos y normalizar su uso. Esto hace que la investigación sobre estos

vehículos sea algo novedoso y enormemente útil para la sociedad. El avance tecnológico actual y la tendencia a compactar y sintetizar sistemas, harán posible que los UAV's formen parte de nuestra vida cotidiana de aquí a unos años.

2 Modelado Matemático del Sistema Quadrotor más Carga

En este capítulo se van a derivar las ecuaciones cinemáticas y dinámicas del sistema, haciendo uso tanto de la formulación de Lagrange como la de Newton. En primer lugar, se detallan las fuerzas y pares actuando sobre el quadrotor, para continuar con el establecimiento de las hipótesis y parámetros necesarios para su obtención. Seguidamente, se plantean las ecuaciones cinemáticas y dinámicas y se realiza la comparación entre ambas formulaciones. Para finalizar con el capítulo, se presenta un modelo de quadrotor sin carga que será empleado para el control lineal en los siguientes capítulos.

2.1 Fuerzas y Pares actuando sobre el Quadrotor

Un quadrotor es un vehículo subactuado que presenta seis grados de libertad y sólo cuatro variables sobre las que poder actuar. Si a este le añadimos una carga colgante, el problema se complica pues pasamos de seis a ocho grados de libertad. Las velocidades de giro de sus rotores son las encargadas de generar los cambios necesarios en las condiciones de vuelo y las fuerzas de sustentación asociadas a la rotación de los mismos se pueden aproximar como sigue:

$$f_i^T = k_{th} \Omega_i^2. \quad (2.1)$$

Donde k_{th} representa el factor de empuje y Ω_i la velocidad de rotación del rotor “i”.

A partir del empuje ejercido por cada rotor, se establecen los pares que actúan en cada dirección del quadrotor. Los pares responsables de los giros de alabeo y cabeceo del drone (τ_{B_x} y τ_{B_y}) dependerán, además de la diferencia de empuje entre sus rotores opuestos, de la distancia de sus rotores al centro de gravedad del vehículo (a). Por el contrario, el par responsable de la variación en guiñada (τ_{B_z}) depende no sólo de la variación de empuje de los cuatro rotores sino también del cociente entre el coeficiente de resistencia (k_{dr}) y el de empuje (k_{th}). Este procedimiento para obtener las fuerzas y pares, sigue los mismos pasos que los adoptados por otros autores [1].

$$\begin{aligned} {}^{(B)}f^T &= f_1^T + f_2^T + f_3^T + f_4^T, \\ {}^{(B)}\tau_{B_x}^T &= a(f_2^T - f_4^T), \\ {}^{(B)}\tau_{B_y}^T &= a(f_1^T - f_3^T), \\ {}^{(B)}\tau_{B_z}^T &= \frac{k_{dr}}{k_{th}}(f_1^T - f_2^T + f_3^T - f_4^T). \end{aligned} \quad (2.2)$$

Las expresiones anteriores pueden ser también reescritas en forma matricial, con ${}^{(B)}\tau = [{}^{(B)}f^T, {}^{(B)}\tau_{B_x}^T, {}^{(B)}\tau_{B_y}^T, {}^{(B)}\tau_{B_z}^T]^T$ como el conocido vector de fuerzas generalizadas. En adelante, se tomará la variable c como el coeficiente entre el coeficiente de resistencia y el de empuje ($c = k_{dr}/k_{th}$).

$${}^{(B)}\tau = \begin{bmatrix} {}^{(B)}f^T \\ {}^{(B)}\tau_{B_x}^T \\ {}^{(B)}\tau_{B_y}^T \\ {}^{(B)}\tau_{B_z}^T \end{bmatrix} = M \begin{bmatrix} f_1^T \\ f_2^T \\ f_3^T \\ f_4^T \end{bmatrix}, \quad \text{con} \quad M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & a & 0 & -a \\ a & 0 & -a & 0 \\ c & -c & c & -c \end{bmatrix}. \quad (2.3)$$

Descrita la obtención de las fuerzas y pares a partir del empuje ejercido por cada rotor, la fuerza máxima de sustentación se calcula a partir de la máxima rotación permitida y será la condición que establece si los rotores se encuentran o no saturados.

$$f_{i_{max}}^T = k_{th} \Omega_{i_{max}}^2. \quad (2.4)$$

El empuje total correspondiente, será por tanto la suma de los empujes máximos ejercidos por los cuatro rotores.

$${}^{(B)}f_{max}^T = 4f_{i_{max}}^T. \quad (2.5)$$

Se ha optado por este modelo dado que se tienen datos tanto de la velocidad máxima de rotación como del coeficiente de empuje proporcionados por Erle-Copter en [34]. Y considerando que se desea comparar este modelo de quadrotor con el de Erle-Copter, será necesario el uso de las mismas variables características para describir ambos sistema.

2.2 Hipótesis y Parámetros

Con el fin de simplificar las ecuaciones que rigen el comportamiento dinámico del sistema formado por el quadrotor más la carga, se han tomado una serie de simplificaciones e hipótesis. Esto supone alejar el modelo de la realidad, con la consecuente pérdida de precisión, pero permite tener un modelo lo suficientemente preciso como para aplicar distintos esquemas de control.

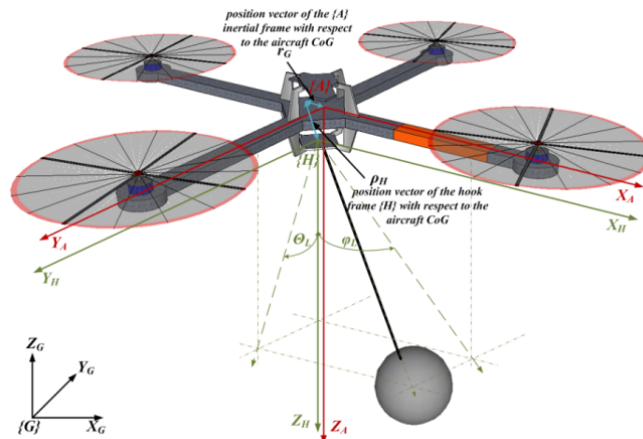


Figura 2.1 Quadrotor con Carga Colgante tomado de [30].

- La carga se modela como una masa puntual unida al quadrotor mediante un cable sin masa, no elástico y tirante en todo momento. Esta hipótesis ha sido considerada también en [30].
- La carga no se encuentra afectada por el giro en guiñada, aunque sí por los ángulos de alabeo y cabeceo.
- Las fuerzas aerodinámicas y de sustentación se suponen aplicadas al centro de gravedad del vehículo como en [22]. Se ha considerado también que el punto de unión del cable al vehículo coincide con su centro de gravedad.
- Se han tenido en cuenta los efectos de resistencia aerodinámica, donde se ha optado por una resistencia distinta para la dirección horizontal y vertical, y el efecto de la fricción en el punto de unión.
- Se considera la atmósfera en calma, es decir, no existen ráfagas de viento que puedan alterar el movimiento del vehículo.

Todas las variables necesarias para una descripción completa del sistema quadrotor más carga se recogen en la Tabla 2.1. La mayor parte de estos datos han sido obtenidos a partir de los proporcionados para Gazebo por Erle-Copter, [34].

Tabla 2.1 Variables características del sistema quadrotor más carga.

Variable	Unidad	Valor	Descripción
m_B	Kg	1.12	Masa del Quadrotor
m_L	Kg	0.118	Masa de la Carga Suspendida
I_B^{xx}	$Kg \cdot m^2$	0.04	Momento de Inercia en x
I_B^{yy}	$Kg \cdot m^2$	0.05	Momento de Inercia en y
I_B^{zz}	$Kg \cdot m^2$	0.1	Momento de Inercia en z
g	m/s^2	9.81	Gravedad
a	m	0.141	Distancia de los Rotores al c.g.
l	m	0.62	Longitud del Cable
k_{th}	-	$8.5486e - 6$	Coefficiente de Empuje
k_{dr}	-	$1.3678e - 7$	Coefficiente de Resistencia
Ω_{imax}^2	rad/s	838	Velocidad de Rotación Máxima
c	-	0.016	k_{dr}/k_{th}
b	$N \cdot m/(rad \cdot s)$	0.0011	Coefficiente de Fricción Unión
C_{Bxy}	$N/(m/s)^2$	1.2	Resistencia del Quadrotor en Dirección Horizontal
C_{Bz}	$N/(m/s)^2$	1.2	Resistencia del Quadrotor en Dirección Vertical
C_L	$N/(m/s)^2$	0.3	Resistencia de la Carga
C	Centígrados	$\frac{180}{\pi} * 100$	Constante de Radianes a Centígrados
T	s	0.01	Tiempo de Muestreo

2.3 Cinemática

En esta sección se pretende desarrollar las ecuaciones cinemáticas de la carga y del cuerpo para poder expresar las ecuaciones dinámicas del sistema en cualquiera de los ejes inerciales, cuerpo o carga.

2.3.1 Cuerpo

Para la obtención de las ecuaciones cinemáticas del cuerpo se considera conocida la posición absoluta del mismo, expresada en el propio sistema inercial (p_B). La velocidad y aceleración absolutas del vehículo se

obtienen derivando a su vez la posición y velocidad del mismo, p_B y $\dot{p}_B$.

$$p_B = {}^I p_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}, \quad \dot{p}_B = {}^I \dot{p}_B = \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix}, \quad \ddot{p}_B = {}^I \ddot{p}_B = \begin{bmatrix} \ddot{x}_B \\ \ddot{y}_B \\ \ddot{z}_B \end{bmatrix}. \quad (2.6)$$

La orientación del vehículo se describe usando la formulación de Tait-Bryan, es decir, considerando una secuencia de tres rotaciones consecutivas en tres ejes de referencia distintos y utilizando para ello los ángulos de Euler.

$$\{I\} \xrightarrow[{}^I u_z]{\psi_B} \{B_1\} \xrightarrow[B_1 u_y]{\theta_B} \{B_2\} \xrightarrow[B_2 u_x]{\phi_B} \{B\} \quad (2.7)$$

El primer de los giros, ψ_B , es una rotación alrededor del eje ${}^I u_z$ y su matriz de transformación vendrá dada por la expresión 2.8.

$$R_{\hat{z}, \psi_B} = \begin{bmatrix} \cos \psi_B & \sin \psi_B & 0 \\ -\sin \psi_B & \cos \psi_B & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.8)$$

La segunda rotación, θ_B , consiste en una variación angular alrededor del eje ${}^{B_1} u_y$ y su matriz se representa mediante 2.9.

$$R_{\hat{y}, \theta_B} = \begin{bmatrix} \cos \theta_B & 0 & -\sin \theta_B \\ 0 & 1 & 0 \\ \sin \theta_B & 0 & \cos \theta_B \end{bmatrix}. \quad (2.9)$$

Y el último de ellos, ϕ_B , es una rotación entorno al eje ${}^{B_2} u_x$, cuya matriz de giro viene dada por 2.10.

$$R_{\hat{x}, \phi_B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_B & \sin \phi_B \\ 0 & -\sin \phi_B & \cos \phi_B \end{bmatrix} \quad (2.10)$$

Finalmente, la matriz de rotación resultante de realizar los tres giros y que representa la actitud del quadrotor, viene dada por la expresión 2.11. Esta matriz transforma cualquier vector expresado en ejes inerciales en otro dado en coordenadas cuerpo.

$${}^B R_I = R_{\hat{x}, \phi_B} R_{\hat{y}, \theta_B} R_{\hat{z}, \psi_B} = \begin{bmatrix} c\psi_B c\theta_B & c\theta_B s\psi_B & -s\theta_B \\ c\psi_B s\phi_B s\theta_B - c\phi_B s\psi_B & c\phi_B c\psi_B + s\phi_B s\psi_B s\theta_B & c\theta_B s\phi_B \\ s\phi_B s\psi_B + c\phi_B c\psi_B s\theta_B & c\phi_B s\psi_B s\theta_B - c\psi_B s\phi_B & c\phi_B c\theta_B \end{bmatrix} \quad (2.11)$$

En donde se ha tomado $\cos \equiv c$ y $\sin \equiv s$, simplificaciones que serán usadas a lo largo de este trabajo para tener expresiones más simples. La matriz de transformación inversa para pasar de ejes cuerpo a inerciales, resulta de multiplicar las matrices ortogonales en orden contrario.

$${}^I R_B = R_{\hat{z}, \psi_B}^T R_{\hat{y}, \theta_B}^T R_{\hat{x}, \phi_B}^T = \begin{bmatrix} c\psi_B c\theta_B & c\psi_B s\phi_B s\theta_B - c\phi_B s\psi_B & s\phi_B s\psi_B + c\phi_B c\psi_B s\theta_B \\ c\theta_B s\psi_B & c\phi_B c\psi_B + s\phi_B s\psi_B s\theta_B & c\phi_B s\psi_B s\theta_B - c\psi_B s\phi_B \\ -s\theta_B & c\theta_B s\phi_B & c\phi_B c\theta_B \end{bmatrix} \quad (2.12)$$

De esta forma, conocida la posición de un punto cualquiera del quadrotor a su centro de masas $({}^{(B)} p_B')$,

representada en ejes cuerpo, se puede hallar fácilmente su expresión respecto al sistema inercial. Esta matriz de rotación es sólo aplicable a fuerzas y velocidades de traslación. La transformación equivalente para velocidades y aceleraciones angulares será más complicada y su representación no resulta trivial.

$$p'_B = {}^I p'_B = p_B + {}^I R_B^{(B)} p'_B \quad (2.13)$$

En donde el índice superior “B” indica que el vector está referido a coordenadas cuerpo.

Siguiendo con la formulación de Tait-Bryan, se obtiene la velocidad angular en ejes cuerpo como la suma de tres rotaciones en tres ejes de referencia distintos. Por tanto, habrá que multiplicar dichas rotaciones por las matrices de transformación correspondientes y expresarlas finalmente en un mismo sistema de referencia.

La variaciones temporales de los ángulos de Tait-Bryan se representan mediante la terna, ${}^I \dot{\eta}_B = [\dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B]^T$, y su velocidad angular expresada en coordenadas cuerpo como ${}^{(B)}\omega_B = [p, q, r]^T$.

$${}^{(B)}\omega_B = \dot{\phi}_B {}^B(B_2 u_x) + \dot{\theta}_B {}^B(B_1 u_y) + \dot{\psi}_B {}^B(I u_z) \quad (2.14)$$

Los vectores ${}^B(B_2 u_x)$, ${}^B(B_1 u_y)$ y ${}^B(I u_z)$ representan tres vectores unitarios respecto a los cuales se realizan los tres giros.

$${}^B(B_2 u_x) = R_{\hat{x}, \phi_B} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.15)$$

$${}^B(B_1 u_y) = R_{\hat{x}, \phi_B} R_{\hat{y}, \theta_B} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ c\phi_B \\ -s\phi_B \end{bmatrix} \quad (2.16)$$

$${}^B(I u_z) = R_{\hat{x}, \phi_B} R_{\hat{y}, \theta_B} R_{\hat{z}, \psi_B} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -s\theta_B \\ s\phi_B c\theta_B \\ c\phi_B c\theta_B \end{bmatrix} \quad (2.17)$$

Y su expresión en forma matricial queda expresada como sigue

$${}^{(B)}\omega_B = W(\eta_B) \dot{\eta}_B, \quad \text{con} \quad W(\eta_B) = \begin{bmatrix} 1 & 0 & -s\theta_B \\ 0 & c\phi_B & s\phi_B c\theta_B \\ 0 & -s\phi_B & c\phi_B c\theta_B \end{bmatrix}. \quad (2.18)$$

Para obtener la velocidad angular representada en ejes inerciales móviles, se realizará el proceso inverso. La inversa de la matriz $W(\eta_B)$ no es una matriz ortogonal y, por tanto, su inversa no coincidirá con su transpuesta.

$$\dot{\eta}_B = W^{-1}(\eta_B) {}^{(B)}\omega_B, \quad \text{con} \quad W^{-1}(\eta_B) = \begin{bmatrix} 1 & s\phi_B t g \theta_B & c\phi_B t g \theta_B \\ 0 & c\phi_B & -s\phi_B \\ 0 & s\phi_B / c\theta_B & c\phi_B / c\theta_B \end{bmatrix}. \quad (2.19)$$

La aceleración angular en coordenadas cuerpo se obtiene derivando con respecto al tiempo la expresión anterior.

$${}^{(B)}\dot{\alpha}_B = {}^{(B)}\dot{\omega}_B = \dot{W}(\eta_B) \dot{\eta}_B + W(\eta_B) \ddot{\eta}_B \quad (2.20)$$

con ${}^{(B)}\dot{\omega}_B = [\dot{p}, \dot{q}, \dot{r}]^T$ y $\ddot{\eta}_B = [\ddot{\phi}_B, \ddot{\theta}_B, \ddot{\psi}_B]^T$. Y su aceleración, expresada mediante la terna de Tait-Bryan, viene dada por la fórmula 2.21.

$$\ddot{\eta}_B = \dot{W}^{-1}(\eta_B)^{(B)}\dot{\omega}_B + W^{-1}(\eta_B)^{(B)}\ddot{\omega}_B. \quad (2.21)$$

Debido a la complejidad de estas expresiones, su desarrollo se realizará mediante el programa matemático MATLAB[®] que permitirá obtener su resultado de forma rápida y sencilla.

2.3.2 Carga

Haciendo uso de los ángulos de Euler, se define el giro ϕ_L como el ángulo formado entre la vertical que pasa por el c.g. del quadrotor y la proyección del cable en el plano vertical $x_B = 0$. Su matriz de rotación viene dada por la expresión 2.22 y su giro es positivo en el sentido contrario a las agujas del reloj.

$$R_{\hat{x}, \phi_L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_L & s\phi_L \\ 0 & -s\phi_L & c\phi_L \end{bmatrix} \quad (2.22)$$

El giro entorno a “y”, θ_L , se define como el ángulo formado entre la vertical y la proyección del cable en el plano $y_B = 0$. La expresión de su matriz se representa en 2.23 y es positivo si gira en dirección contraria a las agujas del reloj.

$$R_{\hat{y}, \theta_L} = \begin{bmatrix} c\theta_L & 0 & -s\theta_L \\ 0 & 1 & 0 \\ s\theta_L & 0 & c\theta_L \end{bmatrix} \quad (2.23)$$

Por tanto, la matriz para pasar de ejes inerciales fijos a ejes carga será resultado de la composición de dos giros.

$${}^L R_I = R_{\hat{x}, \phi_L} R_{\hat{y}, \theta_L} = \begin{bmatrix} c\theta_L & 0 & -s\theta_L \\ s\phi_L s\theta_L & c\phi_L & c\theta_L s\phi_L \\ c\phi_L s\theta_L & -s\phi_L & c\phi_L c\theta_L \end{bmatrix} \quad (2.24)$$

Y su matriz inversa, se obtiene cambiando de orden las matrices transpuestas.

$${}^I R_L = R_{\hat{y}, \theta_L}^T R_{\hat{x}, \phi_L}^T = \begin{bmatrix} c\theta_L & s\phi_L s\theta_L & c\phi_L s\theta_L \\ 0 & c\phi_L & -s\phi_L \\ -s\theta_L & c\theta_L s\phi_L & c\phi_L c\theta_L \end{bmatrix} \quad (2.25)$$

Así pues, se puede expresar la posición absoluta de la carga en ejes inerciales como la suma de la posición del c.g. del cuerpo más la distancia de la carga al c.g. del mismo. Dando por conocida la posición de la carga en sus respectivos ejes, sólo habrá que multiplicar por la matriz de transformación, ${}^I R_L$, para expresar dicha distancia en ejes inerciales.

$$p_L = {}^I p_L = p_B + {}^I R_L ({}^L p_L) = \begin{Bmatrix} x_B - lc\phi_L s\theta_L \\ y_B + ls\phi_L \\ z_B - lc\phi_L c\theta_L \end{Bmatrix} \quad (2.26)$$

Siendo ${}^{(L)}p_L = [0, 0, -L]^T$, la posición de la carga respecto al c.g. del quadrotor en coordenadas de la propia carga y ${}^{(L)}p_B = [0, 0, L]^T$, la posición del c.g. del cuerpo respecto a la carga expresada en sus propios

ejes.

La velocidad y aceleración de la carga se obtienen derivando su posición absoluta respecto al tiempo y sus expresiones se representan a continuación. En ellas, se ha omitido los términos correspondientes a ${}^{(L)}\dot{p}_L$, dado que representan términos nulos al ser ${}^{(L)}p_L$ constante en todo momento y no depender del tiempo.

$$\dot{p}_L = \dot{p}_B + {}^I \dot{R}_L^{(L)} p_L = \begin{Bmatrix} \dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L \\ \dot{y}_B + \dot{\phi}_L l c \phi_L \\ \dot{z}_B + \dot{\phi}_L l s \phi_L c \theta_L + \dot{\theta}_L l c \phi_L s \theta_L \end{Bmatrix} \quad (2.27)$$

$$\ddot{p}_L = \ddot{p}_B + {}^I \ddot{R}_L^{(L)} p_L = \begin{Bmatrix} \ddot{x}_B - \ddot{\theta}_L l c \phi_L c \theta_L + \ddot{\phi}_L l s \phi_L s \theta_L + \dot{\phi}_L^2 l c \phi_L s \theta_L + \dot{\theta}_L^2 l c \phi_L s \theta_L + 2 \dot{\phi}_L \dot{\theta}_L l s \phi_L c \theta_L \\ \ddot{y}_B + \ddot{\phi}_L l c \phi_L - \dot{\phi}_L^2 l s \phi_L \\ \ddot{z}_B + \ddot{\phi}_L l s \phi_L c \theta_L + \ddot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L^2 l c \phi_L c \theta_L + \dot{\theta}_L^2 l c \phi_L c \theta_L - 2 \dot{\phi}_L \dot{\theta}_L l s \phi_L s \theta_L \end{Bmatrix} \quad (2.28)$$

Estas ecuaciones serán útiles para expresar las ecuaciones dinámicas en coordenadas carga o en coordenadas cuerpo según interese, ya que sólo habrá que sustituir la coordenada de interés por su expresión correspondiente. Por ejemplo, si se quiere expresar las ecuaciones en coordenadas de la carga, bastará despejar p_B , $\dot{p}_B$ y $\ddot{p}_B$ de las expresiones 2.26, 2.27 y 2.28 y sustituirlas en las correspondientes ecuaciones dinámicas.

La actitud de la carga, a diferencia de la del cuerpo, tendrá una expresión más sencilla dado que se consideran rotaciones entorno a un sistema de referencia de fijo que se corresponderá con el sistema inercial. De esta forma, la velocidad angular de la carga, expresadas en ejes de la propia carga, queda como

$${}^{(L)}\omega_L = G_L(\eta_L) \dot{\eta}_L, \quad \text{siendo} \quad G_L(\eta_L) = \begin{bmatrix} c \theta_L & 0 & -s \theta_L \\ s \phi_L s \theta_L & c \phi_L & c \theta_L s \phi_L \\ c \phi_L s \theta_L & -s \phi_L & c \phi_L c \theta_L \end{bmatrix}. \quad (2.29)$$

En la fórmula 2.29, se ha considerado nulo el giro entorno al eje “z” y se utiliza el vector $\eta_L = [\phi_L, \theta_L, 0]^T$ para representar la actitud inercial de la carga. La aceleración inercial de la carga, expresada en ejes carga, se obtiene derivando la expresión anterior en el tiempo.

$${}^{(L)}\alpha_L = {}^{(L)}\dot{\omega}_L = \dot{G}_L(\eta_L) \dot{\eta}_L + G_L(\eta_L) \ddot{\eta}_L \quad (2.30)$$

2.4 Dinámica

La obtención de las ecuaciones dinámicas del quadrotor es un problema que ya ha sido abordado por multitud de autores como Bouabdallah et al. [9] y [10]. En esta sección no sólo se desarrollará un modelo dinámico del quadrotor sino también se modelará la dinámica de la carga con el fin de tener las ecuaciones completas del sistema quadrotor más carga. Para su obtención, se usará tanto la formulación de Newton como la formulación de Lagrange en coordenadas cuerpo o carga. Además, todas las magnitudes estarán referidas a los ejes inerciales al igual que la actitud tanto del quadrotor como de la carga.

$$\begin{bmatrix} m_B I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & {}^{(B)}I_B \end{bmatrix} \begin{bmatrix} \ddot{p}_B \\ {}^{(B)}\dot{\omega}_B \end{bmatrix} + \begin{bmatrix} {}^{(B)}\omega_B \times m_B \dot{p}_B \\ {}^{(B)}\omega_B \times {}^{(B)}I_B {}^{(B)}\omega_B \end{bmatrix} = \begin{bmatrix} f_B^T \\ \tau_B \end{bmatrix} \quad (2.31)$$

2.4.1 Formulación de Newton

Se comienza definiendo todas las fuerzas y momentos ejercidos sobre el vehículo y la carga. Una vez definidos, se establece el equilibrio de fuerzas en las tres direcciones del quadrotor para la posición, el balance de momentos entorno a los tres ejes para hallar las ecuaciones de la actitud y el equilibrio de momentos entorno al punto de cogida vehículo-cable para encontrar la actitud de la carga. El subíndice de las fuerzas y momentos indica el tipo de cuerpo, quadrotor o carga, al que están referidos. El superíndice establece el tipo de fuerza o momento ejercido sobre el sistema (Empuje, Resistencia, Gravedad, ...).

Equilibrio de Fuerzas

En primer lugar, se calculan las ecuaciones dinámicas correspondientes al movimientos de traslación del vehículo y de la carga. Para ello, se definen todas las fuerzas actuantes sobre el quadrotor y se plantea el equilibrio de fuerzas en el mismo. Igualmente, se establece el equilibrio de fuerzas para la carga y se despeja de esta última ecuación, la fuerza de interacción que ejerce el cuerpo sobre la carga. Finalmente, se sustituye esta fuerza en la primera ecuación y se obtiene la dinámica completa del movimiento de traslación.

- f_B : Fuerza total actuado sobre el quadrotor.
- f_B^I : Fuerza inercial actuando sobre el quadrotor. Se considera de sentido contrario al movimiento y, por tanto, a la aceleración del vehículo.

$$f_B^I = -m_B \ddot{p}_B = -m_B a_B \quad (2.32)$$

- f_B^D : Fuerza de resistencia aerodinámica del vehículo que se opone al movimiento del mismo. Se considera un modelo cuadrático, cuyo coeficiente aerodinámico será distinto en dirección horizontal ($C_{B_{xy}}$) y vertical (C_{B_z}).

$$f_B^D = -c_B |\dot{p}_B| \dot{p}_B = -c_B |v_B| v_B \quad (2.33)$$

- f_B^g : Fuerza debida a la acción de la gravedad sobre el vehículo. Su signo es opuesto a la dirección de ascenso del vehículo y expresada en ejes inerciales ($u_z = {}^I u_z$).

$$f_B^g = -m_B g u_z \quad (2.34)$$

- f_B^T : Fuerza ejercida por la planta propulsora del quadrotor. Se considera perpendicular al plano de los rotores por lo que habrá que transformarla de ejes cuerpo a ejes inerciales.

$$f_B^T = {}^I R_B^{(B)} f_B^T = {}^I R_B [0, 0, {}^{(B)} f_B^T]^T \quad (2.35)$$

- f_B^L : Fuerza de interacción que ejerce la carga colgante sobre el vehículo. Es una fuerza de acción y reacción que tendrá su fuerza equivalente, de sentido opuesto, actuando sobre la carga.

Definidas todas las fuerzas que actúan sobre el cuerpo, se procede a establecer el equilibrio de fuerzas en las tres direcciones. Para la dirección vertical habría que considerar, además del efecto de la gravedad, el efecto de un coeficiente aerodinámico distinto al de las direcciones en el plano horizontal.

$$f_B^I + f_B^g + f_B^D + f_B^T + f_B^L = 0 \quad (2.36)$$

$$m_B \ddot{p}_B = {}^I R_B^{(B)} f_B^T - m_B g u_z - c_B |\dot{p}_B| \dot{p}_B + f_B^L \quad (2.37)$$

Establecido el equilibrio de fuerzas en el vehículo, se procede de forma idéntica para la carga. Las fuerzas actuantes sobre la misma, serán de la misma índole que las vistas para el caso del cuerpo.

- f_L^I : Fuerza inercial actuando sobre la carga que se opone a su movimiento.

$$f_L^I = -m_L \ddot{p}_L = -m_L a_L \quad (2.38)$$

- f_L^D : Fuerza debida a la resistencia aerodinámica de la carga. En este caso, se considera un único coeficiente actuando en las tres direcciones de la carga dado que se modela como una masa puntual.

$$f_L^D = -c_L |\dot{p}_L| \dot{p}_L = -c_L |v_L| v_L \quad (2.39)$$

- f_L^g : Fuerza debida a la acción gravitatoria sobre la carga, expresada en el sistema inercial.

$$f_L^g = -m_L g u_z \quad (2.40)$$

- f_L^B : Fuerza de interacción que ejerce el vehículo sobre la carga expresada en el sistema inercial. Es de sentido opuesto a la ejercida por la carga sobre el cuerpo.

$$f_L^B = -f_B^L \quad (2.41)$$

Estableciendo el equilibrio de fuerzas en la carga y despejando posteriormente la fuerza de interacción, se obtiene una expresión de dicha fuerza en función de la velocidad y aceleración lineal de la carga.

$$f_L^I + f_L^g + f_L^D + f_L^B = 0 \quad (2.42)$$

$$f_L^B = m_L \ddot{p}_L + m_L g u_z + c_L |\dot{p}_L| \dot{p}_L \quad (2.43)$$

Finalmente, dado que $f_B^L = -f_L^B$, se sustituye dicha expresión en la ecuación 2.37 y se obtienen las ecuaciones dinámicas para el movimiento de traslación del quadrotor más la carga colgante.

$$m_B \ddot{p}_B + m_L \ddot{p}_L = {}^I R_B^{(B)} f_B^T - (m_B + m_L) g u_z - c_B |\dot{p}_B| \dot{p}_B - c_L |\dot{p}_L| \dot{p}_L \quad (2.44)$$

Estas ecuaciones se podrán expresar tanto en coordenadas del propio cuerpo como de la carga, sin más que sustituir las relaciones cinemáticas vistas en la expresiones 2.26, 2.27 y 2.28. Para ello, se despejaría la posición, velocidad y aceleración deseada según convenga.

Balance de Momentos

La obtención de las ecuaciones dinámicas del movimiento de rotación del quadrotor se realizará a través de un balance de momento expresado en ejes cuerpo, ya que la matriz de inercia es diagonal en estos ejes. Posteriormente, se transformará la velocidad y aceleración angular del cuerpo a velocidad y aceleración representada según los ángulos de Tait-Bryan mediante el programa matemático MATLAB[®]. Los momentos externos considerados serán los pares de actuación debidos a las fuerzas de propulsión y el par fricción en el punto de unión quadrotor-carga.

$${}^{(B)} I_B {}^{(B)} \dot{\omega}_B + {}^{(B)} \omega_B \times ({}^{(B)} I_B {}^{(B)} \omega_B) = {}^{(B)} \tau_B^T + {}^{(B)} \tau_B^f \quad (2.45)$$

- ${}^{(B)}I_B$: Matriz Inercial referida al cuerpo.

$${}^{(B)}I_B = \begin{bmatrix} I_B^{xx} & 0 & 0 \\ 0 & I_B^{yy} & 0 \\ 0 & 0 & I_B^{zz} \end{bmatrix} \quad (2.46)$$

- ${}^{(B)}\dot{\omega}_B$: Aceleración angular del vehículo expresada en el propio cuerpo.

$${}^{(B)}\dot{\omega}_B = {}^{(B)}\dot{\alpha}_B = [\dot{p}, \dot{q}, \dot{r}]^T \quad (2.47)$$

- ${}^{(B)}\omega_B$: Velocidad angular del vehículo expresada en el propio cuerpo.

$${}^{(B)}\omega_B = [p, q, r]^T \quad (2.48)$$

- ${}^{(B)}\tau_B^T$: Par de fuerza actuando sobre el vehículo, expresado en el propio cuerpo, y debido a las fuerzas de propulsión ejercidas por los rotores. Su resultado puede consultarse en la expresión 2.2.

$${}^{(B)}\tau_B^T = [{}^{(B)}\tau_{B_x}^T, {}^{(B)}\tau_{B_y}^T, {}^{(B)}\tau_{B_z}^T]^T \quad (2.49)$$

- ${}^{(B)}\tau_B^f$: Momento actuando sobre el vehículo, expresado en el propio cuerpo, y debido a la fricción en el punto de unión cuerpo-carga.

$${}^{(B)}\tau_B^f = [{}^{(B)}\tau_{B_x}^f, {}^{(B)}\tau_{B_y}^f, 0]^T = -b({}^{(B)}\omega_B^f - {}^{(B)}\omega_L^f) = -b({}^{(B)}\omega_B^f - {}^B R_L \dot{\eta}_L) \quad (2.50)$$

Donde b es el denominado coeficiente de fricción en el punto de unión, ${}^{(B)}\omega_L^f$ es la velocidad angular de la carga expresada en el cuerpo y $\dot{\eta}_L = [\dot{\phi}_L, \dot{\theta}_L, 0]^T$ es la velocidad de la carga representada en el sistema inercial de ejes fijos. Esta forma de representar la fricción ha sido considerada por otros autores como [35] y [24], donde la resistencia aerodinámica era modelada como el producto de un coeficiente por la velocidad angular del cuerpo.

Finalmente, dado que el par de fricción sólo afecta a los ejes de giro ${}^{(B)}u_x$ y ${}^{(B)}u_y$, pues dependerá de la diferencia entre la velocidad angular de alabeo y cabeceo del cuerpo y la carga, se representa para estas dos direcciones de interés no teniendo ningún efecto sobre la dirección de guiñada.

$${}^{(B)}\tau_B^f = \begin{bmatrix} {}^{(B)}\tau_{B_x}^f \\ {}^{(B)}\tau_{B_y}^f \\ 0 \end{bmatrix} = -b \begin{bmatrix} p - \dot{\phi}_L c \psi_B c \theta_B - \dot{\theta}_L c \theta_B s \psi_B \\ q - \dot{\phi}_L (c \psi_B s \phi_B s \theta_B - c \phi_B s \psi_B) - \dot{\theta}_L (c \phi_B c \psi_B + s \phi_B s \psi_B s \theta_B) \\ 0 \end{bmatrix} \quad (2.51)$$

Desarrollando la anterior expresión se llega a las denominadas ecuaciones de Euler, expresadas en los ejes cuerpo.

$$\begin{aligned} I_B^{xx} \dot{p} + q r (I_B^{zz} - I_B^{yy}) &= {}^{(B)}\tau_{B_x}^T + {}^{(B)}\tau_{B_x}^f \\ I_B^{yy} \dot{q} + p r (I_B^{xx} - I_B^{zz}) &= {}^{(B)}\tau_{B_y}^T + {}^{(B)}\tau_{B_y}^f \\ I_B^{zz} \dot{r} + p q (I_B^{yy} - I_B^{xx}) &= {}^{(B)}\tau_{B_z}^T \end{aligned} \quad (2.52)$$

La obtención de estas ecuaciones en el sistema inercial de ejes móviles se ha realizado mediante el programa de cálculo MATLAB® y se representarán más adelante en forma matricial. Su cálculo es sencillo, pues dadas

las ecuaciones dinámicas expresadas en el cuerpo, sólo habría que sustituir la velocidad y aceleración angular por las expresiones cinemáticas 2.18 y 2.20 correspondientes.

Equilibrio de Momentos entorno al punto de unión

La ecuaciones dinámicas del movimiento de rotación de la carga se plantean como un equilibrio de momentos entorno al punto de unión cuerpo-carga. Los momentos serán el producto vectorial de la distancia cuerpo-carga, expresada en el sistema inercial, por la correspondiente fuerza ejercida sobre la misma. Siguiendo el procedimiento anterior, las fuerzas consideradas serán las debidas al efecto de la gravedad, la resistencia aerodinámica, la fricción e inerciales. El resultado final, son dos nuevas ecuaciones dinámicas para las variables ϕ_L y θ_L que representan los ángulos de alabeo y cabeceo de la carga respectivamente. Por tanto, sólo las dos primeras componentes de los momentos calculados serán de interés para el control de la carga y se desprecia la última componente como ocurrió para el momento de fricción en el punto de unión.

- τ_L^I : Momento inercial correspondiente a las fuerzas inerciales sobre la carga.

$$\tau_L^I = p_L \times f_L^I = ({}^I R_L^{(L)} p_L) \times (-m_L \ddot{p}_L) \quad (2.53)$$

- τ_L^D : Momento de resistencia correspondiente a las fuerzas de resistencia aerodinámica sobre la carga.

$$\tau_L^D = p_L \times f_L^D = ({}^I R_L^{(L)} p_L) \times (-c_L |\dot{p}_L| \dot{p}_L) \quad (2.54)$$

- τ_L^g : Momento de gravedad debido a las fuerzas gravitatorias que actúan sobre la carga.

$$\tau_L^g = p_L \times f_L^g = ({}^I R_L^{(L)} p_L) \times (-m_L g u_z) \quad (2.55)$$

- τ_L^f : Momento de fricción en el punto de unión, expresado en el sistema inercial de ejes fijos. Para ello, se transforma la velocidad angular del cuerpo, expresada en sus propios ejes, a velocidad dada en ejes inerciales fijos. Ahora bien, teniendo en cuenta la expresión ${}^{(B)}\omega_B = W(\eta_B)\dot{\eta}_B$, se puede representar la velocidad en función de las variaciones temporales de los ángulos de Tait-Bryan.

$$\tau_L^f = -b(\dot{\eta}_L - {}^I R_B^{(B)} \omega_B) = -b(\dot{\eta}_L - {}^I R_B W(\eta_B) \dot{\eta}_B) \quad (2.56)$$

Donde, $\dot{\eta}_L = [\dot{\phi}_L, \dot{\theta}_L, 0]^T$ es la velocidad angular de la carga dada para el sistema inercial fijo.

Finalmente, se representa el vector de fricción resultante que afecta al libre movimiento de la carga.

$$\tau_L^f = -b \left\{ \begin{array}{l} \dot{\phi}_L - \dot{\phi}_B c \psi_B c \theta_B + \dot{\theta}_B s \psi_B \\ \dot{\theta}_L - \dot{\phi}_B c \theta_B s \psi_B - \dot{\theta}_B c \psi_B \end{array} \right\} \quad (2.57)$$

Una vez definidas las fuerzas y momentos necesarios se procede a realizar el balance de momentos entorno al punto de unión. No se han considerado las fuerzas que actúan sobre el quadrotor dado que todas ellas pasan por el c.g del mismo y, además, el punto de unión se ha elegido para que coincida con el c.g. del vehículo.

$$\tau_L^I + \tau_L^D + \tau_L^g + \tau_L^f = 0 \quad (2.58)$$

$${}^I R_L \begin{bmatrix} 0 \\ 0 \\ -l \end{bmatrix} \times (m_L \ddot{p}_L) = {}^I R_L \begin{bmatrix} 0 \\ 0 \\ -l \end{bmatrix} \times \left(-m_L g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) + {}^I R_L \begin{bmatrix} 0 \\ 0 \\ -l \end{bmatrix} \times (-c_L |\dot{p}_L| \dot{p}_L) + \tau_L^f = 0 \quad (2.59)$$

2.4.2 Formulación de Lagrange

Obtenidas las ecuaciones dinámicas mediante la formulación de Newton, se procede a la búsqueda de las mismas mediante la formulación de Lagrange. La idea es comprobar que ambas formulaciones coinciden, es decir que generan las mismas ecuaciones dinámicas, y reforzar así la idea de que el modelo de quadrotor más carga es válido. Para su obtención, se empezará definiendo la ecuación de Lagrange y se continuará formulando las expresiones de energía cinética, potencial y fuerzas generalizadas del sistema. Los cálculos matemáticos se desarrollarán mediante MATLAB[®], evitando un desarrollo tedioso a lo largo del documento.

Para el establecimiento de la ecuación de Lagrange se definen el Lagrangiano (L), como la diferencia entre la energía cinética y potencial del sistema, y las coordenadas generalizadas que serán distintas si se usan las coordenadas cuerpo ($q_B = [x_B, y_B, z_B, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$) o coordenadas carga ($q_L = [x_L, y_L, z_L, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$). Por tanto, la ecuación de Lagrange, incluyendo la energía de disipación de Rayleigh, quedaría como en 2.61.

$$L = T_{sistema} - V_{sistema}, \quad (2.60)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial v}{\partial \dot{q}_i} = Q_i \quad (2.61)$$

Energía Cinética

La energía cinética del quadrotor constará de dos términos, el primero debido al movimiento de traslación del cuerpo, y un segundo término dependiente de la velocidad angular del mismo. Este último término se expresará en el propio cuerpo dado que la matriz de inercia es diagonal en estas coordenadas, por lo que habrá que realizar la transformación correspondiente para representarlo según Tait-Bryan.

$$T_{cuerpo} = \frac{1}{2} \dot{p}_B^T m_B \dot{p}_B + \frac{1}{2} {}^{(B)}\omega_B^T I_B {}^{(B)}\omega_B \quad (2.62)$$

Sustituyendo ${}^{(B)}\omega_B = W(\eta_B) \dot{\eta}_B$ en la ecuación 2.62, se obtiene la fórmula de la energía cinética expresada en el sistema inercial de ejes móviles.

$$T_{cuerpo} = \frac{1}{2} \dot{p}_B^T m_B \dot{p}_B + \frac{1}{2} \dot{\eta}_B^T W^T(\eta_B) I_B W(\eta_B) \dot{\eta}_B \quad (2.63)$$

Desarrollando ahora la parte correspondiente a la actitud del quadrotor y reagrupando sus términos de forma que quede una expresión mucho más compacta, se obtiene una fórmula idéntica a la ya obtenida por otros autores, [8], para el movimiento de rotación.

$$T_{C_{Trasl}} = \frac{1}{2} \dot{p}_B^T m_B \dot{p}_B \quad (2.64)$$

$$T_{C_{Rot}} = \frac{1}{2} I_B^{xx} (\dot{\phi}_B - \dot{\psi}_B s \theta_B)^2 + \frac{1}{2} I_B^{yy} (\dot{\theta}_B c \phi_B + \dot{\psi}_B c \theta_B s \phi_B)^2 + \frac{1}{2} I_B^{zz} (\dot{\theta}_B s \phi_B - \dot{\psi}_B c \phi_B c \theta_B)^2 \quad (2.65)$$

$$T_{cuerpo} = T_{C_{Trasl}} + T_{C_{Rot}} \quad (2.66)$$

La energía cinética de la carga, sin embargo, sólo constará de un término debido al movimiento de traslación de la misma, pues se trata de un objeto pequeño (una bola) y se modela como una masa puntual.

$$T_{carga} = \frac{1}{2} \dot{p}_L^T m_L \dot{p}_L \quad (2.67)$$

Y, por tanto, la energía cinética del sistema completo, será la suma tanto de la energía del cuerpo como de la carga.

$$T_{sistema} = T_{cuerpo} + T_{carga} \quad (2.68)$$

Al igual que en la formulación de Newton, estas ecuaciones se podrían presentar tanto en coordenadas del cuerpo como de la carga, tan sólo habría que sustituir las relaciones cinemáticas correspondientes 2.26, 2.27 y 2.28. Una vez sustituidas la posición, aceleración y velocidad del cuerpo o de la carga según se desee en la expresión 2.68, la ecuación de Lagrange se derivaría para estas variables y se podría efectuar el control directo de las mismas.

Energía Potencial

Tanto en el cuerpo como en la carga, la única energía potencial posible es la debida al efecto de la gravedad sobre ambos. Esto se debe a que se considera el cuerpo como un sólido rígido y se establece la hipótesis de cable sin masa y rígido. Por tanto, la energía potencial tendrá únicamente sentido en la dirección vertical del movimiento.

$$V_{cuerpo} = m_B g z_B = m_B g (p_B \cdot u_z) \quad (2.69)$$

$$V_{carga} = m_L g z_L = m_L g (p_L \cdot u_z) \quad (2.70)$$

$$V_{sistema} = V_{cuerpo} + V_{carga} \quad (2.71)$$

De forma idéntica a la energía cinética, la energía potencial puede presentarse en función de la posición del cuerpo o de la carga, sustituyendo tan sólo las variables z_B o z_L por sus expresiones correctas.

Fuerzas Generalizadas

Entre las fuerzas generalizadas o externas al sistema se encuentran tanto fuerzas disipadoras, como fuerzas y momentos generados por la propulsión del vehículo.

En primer lugar, se analizan las fuerzas y pares debidos al sistema de propulsión generadas por la diferencia de rotación entre los cuatro rotores. Tanto la resultante de fuerza sobre el quadrotor, ${}^{(B)}f_B^T$, como la resultante de momentos, ${}^{(B)}\tau_B^T$, están expresadas en el propio cuerpo y habrá que transformarlas al propio sistema inercial.

Para expresar la fuerza de sustentación en el sistema inercial, sólo habrá que multiplicar por la matriz de transformación, ${}^I R_B$, para pasar de ejes cuerpo a ejes tierra.

$$f_B^T = \begin{bmatrix} f_{B_x}^T \\ f_{B_y}^T \\ f_{B_z}^T \end{bmatrix} = {}^I R_B {}^{(B)}f_B^T = \begin{bmatrix} {}^{(B)}f^T (s\phi_B s\psi_B + c\phi_B c\psi_B s\theta_B) \\ {}^{(B)}f^T (c\phi_B s\psi_B s\theta_B - c\psi_B s\phi_B) \\ {}^{(B)}f^T c\phi_B c\theta_B \end{bmatrix} \quad (2.72)$$

La obtención del momento de sustentación, expresado en el sistema inercial móvil, es más complicado debido a que las respectivas componentes del momento se tienen que transformar a los distintos ejes de referencia. El momento en el eje $^{(B)}u_x$ será transformado por la matriz de giro $R_{\hat{x},\phi_B}^{-1}$, a su vez el momento aplicado en $^{(B)}u_y$, será multiplicado por las matrices $R_{\hat{y},\theta_B}^{-1} R_{\hat{x},\phi_B}^{-1}$, y al momento contenido en el eje $^{(B)}u_z$, se le aplicará el cambio $R_{\hat{z},\psi_B}^{-1} R_{\hat{y},\theta_B}^{-1} R_{\hat{x},\phi_B}^{-1}$.

$$\tau_B^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} R_{\hat{x},\phi_B}^{-1} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} R_{\hat{y},\theta_B}^{-1} R_{\hat{x},\phi_B}^{-1} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} R_{\hat{z},\psi_B}^{-1} R_{\hat{y},\theta_B}^{-1} R_{\hat{x},\phi_B}^{-1} \quad {}^{(B)}\tau_B^T \quad (2.73)$$

$$\tau_B^T = WI(\eta_B)^{(B)} \tau_B^T = \begin{bmatrix} \tau_{B_x}^T \\ \tau_{B_y}^T \\ \tau_{B_z}^T \end{bmatrix} \quad \text{con} \quad WI = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_B & -s\phi_B \\ -s\theta_B & c\theta_B s\phi_B & c\phi_B c\theta_B \end{bmatrix}. \quad (2.74)$$

Aplicando ahora el principio de los trabajos virtuales, se establecen las fuerzas generalizadas debidas a la fuerzas y momentos de propulsión. Estas serán distintas según se considere coordenadas cuerpo ($q_B = [x_B, y_B, z_B, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$) o coordenadas carga ($q_L = [x_L, y_L, z_L, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$).

El trabajo virtual para la fuerza de sustentación es

$$\delta w^T = \sum_{i=1}^{i=8} Q_{B_i}^T \partial q_i, \quad \text{con} \quad Q_{B_i}^T = \sum_{j=1}^{j=8} F_{B_j}^T \frac{\partial r_j}{\partial q_i}. \quad (2.75)$$

Y las respectivas fuerzas generalizadas en coordenadas cuerpo y carga son respectivamente

$$Q_B^T = \begin{bmatrix} Q_{x_B}^T \\ Q_{y_B}^T \\ Q_{z_B}^T \\ Q_{\phi_B}^T \\ Q_{\theta_B}^T \\ Q_{\psi_B}^T \\ Q_{\phi_L}^T \\ Q_{\theta_L}^T \end{bmatrix} = \begin{bmatrix} f_{B_x}^T \\ f_{B_y}^T \\ f_{B_z}^T \\ \tau_{B_x}^T \\ \tau_{B_y}^T \\ \tau_{B_z}^T \\ 0 \\ 0 \end{bmatrix} \quad \text{y} \quad Q_L^T = \begin{bmatrix} Q_{x_L}^T \\ Q_{y_L}^T \\ Q_{z_L}^T \\ Q_{\phi_B}^T \\ Q_{\theta_B}^T \\ Q_{\psi_B}^T \\ Q_{\phi_L}^T \\ Q_{\theta_L}^T \end{bmatrix} = \begin{bmatrix} f_{B_x}^T \\ f_{B_y}^T \\ f_{B_z}^T \\ \tau_{B_x}^T \\ \tau_{B_y}^T \\ \tau_{B_z}^T \\ -ls\phi_L s\theta_L f_{B_x}^T - lc\phi_L f_{B_y}^T - ls\phi_L c\theta_L f_{B_z}^T \\ lc\phi_L c\theta_L f_{B_x}^T - lc\phi_L s\theta_L f_{B_z}^T \end{bmatrix}. \quad (2.76)$$

La diferencia entre considerar coordenadas cuerpo o coordenadas carga reside en que habrá que considerar los términos, $Q_{\phi_L}^T$ y $Q_{\theta_L}^T$, en la expresión de las fuerzas generalizadas para el caso de la carga. Estos términos son debidos a que la resultante de fuerzas de propulsión se supone aplicada sobre el c.g. del vehículo y habrá que desplazar su efecto al c.g. de la carga.

Definidas las fuerzas generalizadas correspondientes al empuje de los rotores, se procede a definir las fuerzas generalizadas debidas a la resistencia aerodinámica. Usando la función de disipación de Rayleigh, la resistencia aerodinámica se puede aproximar como una función cúbica de la velocidad lineal del cuerpo y la carga. El uso de esta función supone la inclusión de un término adicional en las ecuaciones de Lagrange, y el conocimiento de esta función de disipación, v .

$$Q_{B_i}^{fr} = -\frac{\partial v}{\partial \dot{q}_i} \quad (2.77)$$

$$v = \frac{1}{3}c_B|\dot{\phi}_B|^3 + \frac{1}{3}c_L|\dot{\phi}_L|^3 \quad (2.78)$$

Aunque se ha representado una resistencia global para el cuerpo, hay que tener en cuenta que sería distinta para dirección horizontal y vertical.

Por último, se obtienen las fuerzas generalizadas correspondientes a la fricción en el punto de unión. Partiendo de la ecuación (2.51), expresada en ejes cuerpo, se calcula el momento de fricción expresado en el sistema inercial móvil mediante la matriz de transformación vista para el momento de sustentación, $WI(\eta_B)$.

$$\tau_B^f = WI(\eta_B)^{(B)} \tau_B^f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi_B & -s\phi_B \\ -s\theta_B & c\theta_B s\phi_B & c\phi_B c\theta_B \end{bmatrix}^{(B)} \tau_B^f = \begin{bmatrix} \tau_{B_x}^f \\ \tau_{B_y}^f \\ 0 \end{bmatrix} \quad (2.79)$$

En donde se desprecia el último término dado que se considera que el giro en guiñada no afecta al movimiento de la carga. Utilizando ahora el principio de los trabajos virtuales para el momento de fricción, se tiene las siguientes fuerzas generalizadas para la actitud del quadrotor. Estas serán válidas tanto para coordenadas cuerpo como coordenadas carga pues la actitud del quadrotor se calcula de forma idéntica en ambas expresiones.

$$\begin{aligned} Q_{\phi_B}^f &= \tau_{B_x}^f \\ Q_{\theta_B}^f &= \tau_{B_y}^f \end{aligned} \quad (2.80)$$

Las ecuaciones de la carga también tendrán componentes debidas a la fricción pero su cálculo no es trivial. Estas ecuaciones están referidas al sistema inercial de ejes fijos y no al de ejes móviles como la actitud del vehículo, por lo que habrá que transformar el momento debido a la fricción al sistema inercial de ejes fijos.

$$\tau_L^f = -b(\dot{\eta}_L - {}^I R_B W(\eta_B) \dot{\eta}_B) = \begin{bmatrix} \tau_{L_x}^f \\ \tau_{L_y}^f \\ 0 \end{bmatrix} \quad (2.81)$$

Donde se ha despreciado el término correspondiente a la guiñada, pues se considera que no influye sobre las ecuaciones del movimiento. Aplicando ahora el principio de los trabajos virtuales, se obtienen las fuerzas generalizadas de la carga debido a la fricción en el punto de unión. Para ello, se ha tomado la variación del ángulo de la carga como el cociente entre el desplazamiento horizontal de la misma y la distancia vertical al c.g del quadrotor.

$$\begin{aligned} \partial \phi_L &= \frac{\partial y_L}{lc\phi_L c\theta_L} \\ \partial \theta_L &= -\frac{\partial x_L}{lc\phi_L c\theta_L} \end{aligned} \quad (2.82)$$

Estas fuerzas tendrán la misma expresión tanto en coordenadas cuerpo como en coordenadas carga al no depender de la posición y si de la variación angular.

$$\begin{aligned} Q_{\phi_L}^f &= lc\phi_L \frac{\tau_{L_x}^f}{lc\phi_L c\theta_L} - ls\phi_L s\theta_L \frac{\tau_{L_y}^f}{lc\phi_L c\theta_L} \\ Q_{\theta_L}^f &= lc\phi_L c\theta_L \frac{\tau_{L_y}^f}{lc\phi_L c\theta_L} \end{aligned} \quad (2.83)$$

Finalmente, la fuerza resultante para cada variable característica es la suma de las fuerzas generalizadas de fricción y de propulsión.

$$Q = \begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_{\phi_B} \\ Q_{\theta_B} \\ Q_{\psi_B} \\ Q_{\phi_L} \\ Q_{\theta_L} \end{bmatrix} = \begin{bmatrix} Q_x^T \\ Q_y^T \\ Q_z^T \\ Q_{\phi_B}^T + Q_{\phi_B}^f \\ Q_{\theta_B}^T + Q_{\theta_B}^f \\ Q_{\psi_B}^T \\ Q_{\phi_L}^T + Q_{\phi_L}^f \\ Q_{\theta_L}^T + Q_{\theta_L}^f \end{bmatrix} \quad (2.84)$$

Una vez obtenidas las ecuaciones del movimiento para ambas formulaciones, se despejan las aceleraciones de las mismas y se compararan para ambos modelos. El resultado es una diferencia en aceleración nula, verificando así la validez de las ecuaciones obtenidas para describir el movimiento y el comportamiento dinámico del vehículo.

2.5 Modelo Matricial

Para visualizar las ecuaciones del movimiento de una forma más clara, se ha optado por representar sus resultados en forma matricial. Esta representación tiene la ventaja de tratar las ecuaciones como el producto de una matriz por un vector de coordenadas generalizadas, $q_B = [x_B, y_B, z_B, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$ para el cuerpo y $q_L = [x_L, y_L, z_L, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L]^T$ para la carga.

$$Q = M(q)\ddot{q} + v(q, \dot{q})\dot{q} + D(q, \dot{q})\dot{q} + g(q) \quad (2.85)$$

Donde están representadas las siguientes variables:

- $M(q)$: Matriz de inercia o masa.
- $v(q, \dot{q})$: Matriz de fuerzas centrífugas y de coriolis.
- $D(q, \dot{q})$: Matriz de resistencia.
- $g(q)$: Matriz de fuerzas gravitatorias.
- Q : Vector de fuerzas generalizadas.

2.5.1 Coordenadas Cuerpo

En esta subsección se presentan las matrices correspondientes al modelo dinámico, expresado en coordenadas cuerpo.

$$M(q) = \begin{bmatrix} m_B + m_L & 0 & 0 & 0 & 0 \\ 0 & m_B + m_L & 0 & 0 & 0 \\ 0 & 0 & m_B + m_L & 0 & 0 \\ 0 & 0 & 0 & I_B^{xx} & 0 \\ 0 & 0 & 0 & 0 & I_B^{zz} + (I_B^{yy} - I_B^{zz})c^2\phi_B \\ 0 & 0 & 0 & -I_B^{xx}s\theta_B & (I_B^{yy} - I_B^{zz})c\phi_Bc\theta_Bs\phi_B \\ lm_Ls\phi_Ls\theta_L & lm_Lc\phi_L & lm_Lc\theta_Ls\phi_L & 0 & 0 \\ -lm_Lc\phi_Lc\theta_L & 0 & lm_Lc\phi_Ls\theta_L & 0 & 0 \end{bmatrix} \dots \quad (2.86)$$

$$\begin{bmatrix} 0 & lm_Ls\phi_Ls\theta_L & -lm_Lc\phi_Lc\theta_L \\ 0 & lm_Lc\phi_L & 0 \\ 0 & lm_Lc\theta_Ls\phi_L & lm_Lc\phi_Ls\theta_L \\ -I_B^{xx}s\theta_B & 0 & 0 \\ (I_B^{yy} - I_B^{zz})c\phi_Bc\theta_Bs\phi_B & 0 & 0 \\ I_B^{xx} + (I_B^{yy} - I_B^{xx})c^2\theta_B + (I_B^{zz} - I_B^{yy})c^2\phi_Bc^2\theta_B & 0 & 0 \\ 0 & l^2m_L & 0 \\ 0 & 0 & l^2m_Lc^2\phi_L \end{bmatrix}$$

$$g(q) = \begin{bmatrix} 0 \\ 0 \\ g(m_B + m_L) \\ 0 \\ 0 \\ 0 \\ glm_Lc\theta_Ls\phi_L \\ glm_Lc\phi_Ls\theta_L \end{bmatrix} \quad (2.87)$$

$$Q = \begin{bmatrix} {}^{(B)}f^T(s\phi_Bs\psi_B + c\phi_Bc\psi_Bs\theta_B) \\ {}^{(B)}f^T(c\phi_Bs\psi_Bs\theta_B - c\psi_Bs\phi_B) \\ {}^{(B)}f^T(c\phi_Bc\theta_B) \\ {}^{(B)}\tau_{B_x}^T \\ {}^{(B)}\tau_{B_y}^Tc\phi_B - {}^{(B)}\tau_{B_z}^Ts\phi_B \\ -{}^{(B)}\tau_{B_x}^Ts\theta_B + {}^{(B)}\tau_{B_y}^Tc\theta_Bs\phi_B + {}^{(B)}\tau_{B_z}^Tc\phi_Bc\theta_B \\ 0 \\ 0 \end{bmatrix} \quad (2.88)$$

$$\begin{aligned}
v(q, \dot{q}) = & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(I_B^{xx} - I_B^{yy} + I_B^{zz})\dot{\psi}_B c\theta_B + (I_B^{zz} - I_B^{yy})\dot{\theta}_B s\phi_B c\phi_B + (I_B^{yy} - I_B^{zz})\dot{\psi}_B c^2\phi_B c\theta_B & \dots \\ 0 & 0 & 0 & \frac{1}{2}(I_B^{zz} - I_B^{xx} - I_B^{yy})\dot{\theta}_B c\theta_B + (I_B^{yy} - I_B^{zz})\dot{\theta}_B c^2\phi_B c\theta_B + (I_B^{yy} - I_B^{zz})\dot{\psi}_B c\phi_B c^2\theta_B s\phi_B & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} (I_B^{yy} - I_B^{zz})\dot{\theta}_B s\phi_B c\phi_B + \frac{1}{2}(I_B^{yy} - I_B^{zz} - I_B^{xx})\dot{\psi}_B c\theta_B + (I_B^{zz} - I_B^{yy})\dot{\psi}_B c^2\phi_B c\theta_B \\ (I_B^{zz} - I_B^{yy})\dot{\phi}_B s\phi_B c\phi_B \\ \frac{1}{2}(I_B^{zz} - I_B^{xx} - I_B^{yy})\dot{\phi}_B c\theta_B + (I_B^{yy} - I_B^{zz})\dot{\phi}_B c^2\phi_B c\theta_B + (I_B^{yy} - I_B^{zz})\dot{\psi}_B s\theta_B c\theta_B + (I_B^{zz} - I_B^{yy})\dot{\theta}_B c\phi_B s\phi_B s\theta_B \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} \frac{1}{2}(I_B^{yy} - I_B^{zz} - I_B^{xx})\dot{\theta}_B c\theta_B + (I_B^{zz} - I_B^{yy})\dot{\psi}_B c\phi_B c^2\theta_B s\phi_B + (I_B^{zz} - I_B^{yy})\dot{\theta}_B c^2\phi_B c\theta_B \\ (I_B^{yy} - I_B^{zz})\dot{\psi}_B s\theta_B c\theta_B + \frac{1}{2}(I_B^{xx} - I_B^{yy} + I_B^{zz})\dot{\phi}_B c\theta_B + (I_B^{zz} - I_B^{yy})\dot{\psi}_B c^2\phi_B c\theta_B s\theta_B \\ (I_B^{xx} - I_B^{yy})\dot{\theta}_B s\theta_B c\theta_B + (I_B^{yy} - I_B^{zz})\dot{\phi}_B c\phi_B c^2\theta_B s\phi_B + (I_B^{yy} - I_B^{zz})\dot{\theta}_B c^2\phi_B c\theta_B s\theta_B \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} lm_L(c\phi_L s\theta_L \dot{\phi}_L + c\theta_L s\phi_L \dot{\theta}_L) & lm_L(c\phi_L s\theta_L \dot{\theta}_L + c\theta_L s\phi_L \dot{\phi}_L) \\ -lm_L s\phi_L \dot{\phi}_L & 0 \\ lm_L(c\phi_L c\theta_L \dot{\phi}_L - s\phi_L s\theta_L \dot{\theta}_L) & lm_L(c\phi_L c\theta_L \dot{\theta}_L - s\phi_L s\theta_L \dot{\phi}_L) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & l^2 m_L c\phi_L s\phi_L \dot{\theta}_L \\ -l^2 m_L c\phi_L s\phi_L \dot{\theta}_L & -l^2 m_L c\phi_L s\phi_L \dot{\phi}_L \end{bmatrix} \\
& + (I_B^{yy} - I_B^{zz})\dot{\phi}_B c^2\phi_B c\theta_B \quad (2.89)
\end{aligned}$$

$$D(q, \dot{q}) = \begin{bmatrix} C_{Bxy}|\dot{x}_B| + C_L|\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & C_{Bxy}|\dot{y}_B| + C_L|\dot{y}_B + \dot{\phi}_L l c \phi_L| & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ C_L l s \phi_L s \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & C_L l c \phi_L |\dot{y}_B + \dot{\phi}_L l c \phi_L| & 0 \\ -C_L l c \phi_L c \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & 0 & 0 \\ \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ C_{Bz}|\dot{z}_B| + C_L|\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \\ 0 & -b s \theta_B & 0 \\ C_L l s \phi_L c \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & -b(c \psi_B \frac{c \theta_B}{c \theta_L} - c \theta_B s \psi_B t g \phi_L t g \theta_L) & 0 \\ C_L l c \phi_L s \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & -b c \theta_B s \psi_B & 0 \\ \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -b s \theta_B & 0 \\ b c^2 \phi_B & b c \phi_B c \theta_B s \phi_B & 0 \\ b c \phi_B c \theta_B s \phi_B & b(1 - c^2 \phi_B c^2 \theta_B) & 0 \\ b(\frac{s \psi_B}{c \theta_L} + c \psi_B t g \phi_L t g \theta_L) & 0 & \frac{b}{c \theta_L} + C_L l^2 c^2 \phi_L |\dot{y}_B + \dot{\phi}_L l c \phi_L| + C_L l^2 c^2 \theta_L s^2 \phi_L |\dot{z}_B \\ -b c \psi_B & 0 & -C_L l^2 c \phi_L c \theta_L s \phi_L s \theta_L |\dot{x}_B + \dot{\phi}_L l s \theta_L s \phi_L - \dot{\theta}_L l c \phi_L c \theta_L| \\ \\ c_L l s \phi_L s \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & & \\ c_L l c \phi_L |\dot{y}_B + \dot{\phi}_L l c \phi_L| & & \\ c_L l s \phi_L c \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & & \\ -b c \psi_B c \theta_B & & \\ b(c^2 \phi_B s \psi_B - c \phi_B c \psi_B s \phi_B s \theta_B) & & \\ b(c \phi_B c \theta_B s \phi_B s \psi_B + c^2 \phi_B c \psi_B c \theta_B s \theta_B) & & \\ + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L + C_L l^2 s^2 \phi_L s^2 \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & -b t g \phi_L t g \theta_L & \\ + C_L l^2 c \phi_L s \phi_L c \theta_L s \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & b + C_L l^2 c^2 \phi_L c^2 \theta_L |\dot{x}_B & \\ \\ -c_L l c \phi_L c \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & & \\ 0 & & \\ c_L l c \phi_L s \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & & \\ -b s \psi_B c \theta_B & & \\ -b(c^2 \phi_B c \psi_B + c \phi_B s \phi_B s \psi_B s \theta_B) & & \\ -b(c \phi_B c \psi_B c \theta_B s \phi_B - c^2 \phi_B c \theta_B s \psi_B s \theta_B) & & \\ + C_L l^2 c \phi_L c \theta_L s \phi_L s \theta_L |\dot{z}_B + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| - C_L l^2 c \phi_L c \theta_L s \phi_L s \theta_L |\dot{x}_B - \dot{\theta}_L l c \phi_L c \theta_L + \dot{\phi}_L l s \phi_L s \theta_L| & & \\ + \dot{\phi}_L l s \theta_L s \phi_L - \dot{\theta}_L l c \phi_L c \theta_L + C_L l^2 c^2 \phi_L s^2 \theta_L |\dot{z}_B + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l s \phi_L c \theta_L| & & \end{bmatrix} \quad (2.90)$$

2.5.2 Coordenadas Carga

La matrices de masa, de fuerza gravitatoria y de fuerza centrífuga, expresadas en coordenadas carga, tienen los mismos términos que las formuladas en coordenadas cuerpo para la posición y actitud del vehículo. Sin

embargo, cambian aquellos términos que dependen de los ángulos de la carga (ϕ_L y θ_L) y que tienen en cuenta la dinámica oscilante de la misma. Para evitar una representación tediosa, sólo estos últimos términos serán representados y se refiere al lector a la subsección anterior para conocer el resto. Esta representación en coordenadas de la propia carga permite efectuar un control óptimo sobre la misma y ha sido considerada también por otros autores como [36].

$$M(q) = \begin{bmatrix} m_B + m_L & \cdots & \cdots & -lm_B s\phi_L s\theta_L & lm_B c\phi_L c\theta_L \\ \vdots & \ddots & & -lm_B c\phi_L & 0 \\ & & & -lm_B c\theta_L s\phi_L & -lm_B c\phi_L s\theta_L \\ \vdots & & & \vdots & \vdots \\ -lm_B s\phi_L s\theta_L & -lm_B c\phi_L & -lm_B c\theta_L s\phi_L & \cdots & l^2 m_B & 0 \\ lm_B c\phi_L c\theta_L & 0 & -lm_B c\phi_L s\theta_L & \cdots & 0 & l^2 m_B c^2 \phi_L \end{bmatrix} \quad (2.91)$$

$$g(q) = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ -glm_B c\theta_L s\phi_L \\ -glm_B c\phi_L s\theta_L \end{bmatrix} \quad (2.92)$$

$$Q = \begin{bmatrix} {}^{(B)}f^T (s\phi_B s\psi_B + c\phi_B c\psi_B s\theta_B) \\ \vdots \\ \vdots \\ {}^{(B)}f^T l (c\phi_L c\psi_B s\phi_B - c\phi_B c\theta_L c\theta_B s\phi_L - c\phi_B c\phi_L s\psi_B s\theta_B - s\phi_B s\phi_L s\psi_B s\theta_L - c\phi_B c\psi_B s\phi_L s\theta_L s\theta_B) \\ {}^{(B)}f^T l (-c\phi_B c\phi_L c\theta_B s\theta_L + c\phi_L c\theta_L s\phi_B s\psi_B + c\phi_B c\phi_L c\psi_B c\theta_L s\theta_B) \end{bmatrix} \quad (2.93)$$

$$v(q, \dot{q}) = \begin{bmatrix} 0 & \cdots & \cdots & -lm_B (c\phi_L s\theta_L \dot{\phi}_L + c\theta_L s\phi_L \dot{\theta}_L) & -lm_B (c\phi_L s\theta_L \dot{\theta}_L + c\theta_L s\phi_L \dot{\phi}_L) \\ \vdots & \ddots & & lm_B s\phi_L \dot{\phi}_L & 0 \\ & & & lm_B (-c\phi_L c\theta_L \dot{\phi}_L + s\phi_L s\theta_L \dot{\theta}_L) & lm_B (s\phi_L s\theta_L \dot{\phi}_L - c\phi_L c\theta_L \dot{\theta}_L) \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & l^2 m_B c\phi_L s\phi_L \dot{\theta}_L \\ 0 & 0 & 0 & \cdots & -l^2 m_B c\phi_L s\phi_L \dot{\theta}_L & -l^2 m_B \dot{\phi}_L c\phi_L s\phi_L \end{bmatrix} \quad (2.94)$$

Para la matriz de resistencia no sólo cambian los términos asociados a los ángulos de la carga sino también los asignados a la posición de la misma debido a las fuerzas de resistencia aerodinámica sobre el cuerpo y la carga. Los únicos que no varían son los asociados a la fricción en el punto de unión que serán idénticos tanto

para coordenadas carga como coordenadas cuerpo.

$$\begin{aligned}
 D(q, \dot{q}) = & \begin{bmatrix} C_L |\dot{x}_L| + C_{B_{xy}} |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| & 0 & & & \\ & 0 & C_L |\dot{y}_L| + C_{B_{xy}} |\dot{y}_L - \dot{\phi}_L l c \phi_L| & & \\ & 0 & 0 & & \\ & \vdots & \vdots & & \dots \\ -C_{B_{xy}} l s \phi_L s \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| & -C_{B_{xy}} l c \phi_L |\dot{y}_L - \dot{\phi}_L l c \phi_L| & & & \\ C_{B_{xy}} l c \phi_L c \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| & 0 & & & \\ & 0 & & & \dots \\ & 0 & & & \dots \\ C_L |\dot{z}_L| + C_{B_z} |-\dot{z}_L + \dot{\theta}_L l c \phi_L s \theta_L + \dot{\phi}_L l c \theta_L s \phi_L| & & & & \dots \\ & \vdots & & & \\ -C_{B_z} l c \theta_L s \phi_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & \dots & \frac{b}{c \theta_L} + C_{B_{xy}} l^2 c^2 \phi_L |\dot{y}_L - \dot{\phi}_L l c \phi_L| + C_{B_{xy}} l^2 s^2 \phi_L & & \dots \\ -C_{B_z} l c \phi_L s \theta_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & \dots & -C_{B_{xy}} l^2 c \phi_L c \theta_L s \phi_L s \theta_L |\dot{x}_L + & & \\ & & & & \\ & & & & \\ -C_{B_{xy}} l s \phi_L s \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| & & & & \\ & & & & \\ -C_{B_{xy}} l c \phi_L |\dot{y}_L - \dot{\phi}_L l c \phi_L| & & & & \\ -C_{B_z} l c \theta_L s \phi_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & & & & \\ & & & & \vdots \\ & & & & \\ s^2 \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| + C_{B_z} l^2 c^2 \theta_L s^2 \phi_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & -b t g \phi_L t g \theta_L & & & \dots \\ + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| + C_{B_z} l^2 c \phi_L c \theta_L s \phi_L s \theta_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & & & & \\ & & & & \\ & & & & \\ C_{B_{xy}} l c \phi_L c \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| & & & & \\ & & & & \\ -C_{B_z} l c \phi_L s \theta_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & & & & \\ & & & & \vdots \\ -C_{B_{xy}} l^2 c \phi_L c \theta_L s \phi_L s \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| + C_{B_z} l^2 c \theta_L c \phi_L s \phi_L s \theta_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & & & & \\ b + C_{B_{xy}} l^2 c^2 \phi_L c^2 \theta_L |\dot{x}_L + \dot{\theta}_L l c \phi_L c \theta_L - \dot{\phi}_L l s \phi_L s \theta_L| + C_{B_z} l^2 c^2 \phi_L s^2 \theta_L |-\dot{z}_L + \dot{\phi}_L l c \theta_L s \phi_L + \dot{\theta}_L l c \phi_L s \theta_L| & & & & \end{bmatrix} \\
\end{aligned} \tag{2.95}$$

2.6 Ecuaciones del Quadrotor sin Carga

Para terminar con este capítulo y dado que se busca aplicar los esquemas de control desarrollados por ArduPilot a nuestro modelo matemático en MATLAB[®], se presentan las ecuaciones del mismo sin considerar el efecto de la carga. Esta eliminación se realiza dado que el modelo en Gazebo de Erle-Copter no presenta ninguna carga colgante y se desea dos vehículos similares para poder compararlos.

Tanto la posición como la altitud del vehículo estarán expresadas en el propio sistema inercial de acuerdo con el modelo de control de ArduPilot para Gazebo. Sin embargo, las velocidades angulares del dron vendrán expresadas para el cuerpo pues son medidas mediante IMUs y, por consiguiente, su actitud será controlada también dichos ejes. Se tiene pues un sistema de seis ecuaciones con seis grados de libertad. Las tres primeras representan el comportamiento dinámico de la posición y están expresadas en ejes inerciales. Las tres últimas

describen la actitud y vienen dadas en el sistema de referencia cuerpo.

$$\begin{aligned}
 m_B \ddot{x}_B &= {}^{(B)}f^T (s\phi_B s\psi_B + c\phi_B c\psi_B s\theta_B) \\
 m_B \ddot{y}_B &= {}^{(B)}f^T (c\phi_B s\psi_B s\theta_B - c\psi_B s\phi_B) \\
 m_B \ddot{z}_B &= {}^{(B)}f^T c\phi_B c\theta_B - gm_B
 \end{aligned} \tag{2.96}$$

$$\begin{aligned}
 I_B^{xx} \dot{p} + qr(I_B^{zz} - I_B^{yy}) &= {}^{(B)}\tau_{B_x}^T \\
 I_B^{yy} \dot{q} + pr(I_B^{xx} - I_B^{zz}) &= {}^{(B)}\tau_{B_y}^T \\
 I_B^{zz} \dot{r} + pq(I_B^{yy} - I_B^{xx}) &= {}^{(B)}\tau_{B_z}^T
 \end{aligned}$$

Para la simulación del sistema en MATLAB[®] SIMULINK[®] no se ha tenido en cuenta el efecto de la resistencia aerodinámica, interviniendo sólo la sustentación y la gravedad. Estas ecuaciones son bastante más simples que las representadas para las variaciones de los ángulos de Tait-Bryan, donde aparecían términos acoplados debido al efecto de las fuerzas centrífugas y de coriolis.

3 Linealización

En este capítulo se van a linealizar las ecuaciones del movimiento del sistema formado por el vehículo más la carga a partir del sistema matricial obtenido en el capítulo anterior. También se calcularán sus expresiones en el dominio de Laplace y su linealización en el espacio de estados. Estas transformaciones se aplicarán tanto a coordenadas cuerpo como a coordenadas carga y se tendrá en cuenta dos casos de estudio: velocidad nula (hovering) y velocidad constante.

3.1 Velocidad Nula

3.1.1 Coordenadas Cuerpo

La obtención de las ecuaciones linealizadas requiere del desarrollo de Taylor entorno a un punto de equilibrio o hovering. Este punto ha sido bastante recurrente en la literatura y empleado, por ejemplo, en [1]. Es el resultado de establecer un equilibrio de fuerzas y momentos en las ecuaciones dinámicas, cuando las aceleraciones y velocidades son nulas. Los valores tomados para este punto de operación se establecen en 3.1 para las variables de estado y en 3.2 para las variables de actuación.

$$\begin{aligned}\bar{x}_B &= \bar{y}_B = \bar{z}_B = cte. \\ \bar{\phi}_B &= \bar{\theta}_B = \bar{\psi}_B = 0 \\ \bar{\phi}_L &= \bar{\theta}_L = 0\end{aligned}\tag{3.1}$$

$$\bar{q}_B = [\bar{x}_B, \bar{y}_B, \bar{z}_B, 0, 0, 0, 0, 0]^T$$

Donde el valor de ψ_B se ha tomado nulo por comodidad pero puede ser un valor arbitrario constante.

$$\begin{aligned}{}^{(B)}\bar{f}^T &= mg \\ {}^{(B)}\bar{\tau}_{B_x}^T &= {}^{(B)}\bar{\tau}_{B_y}^T = {}^{(B)}\bar{\tau}_{B_z}^T = 0\end{aligned}\tag{3.2}$$

$$\bar{\tau} = [{}^{(B)}\bar{f}^T, 0, 0, 0]^T$$

Una vez definido el punto de equilibrio, se introducen valores incrementales alrededor del mismo. Las nuevas variables de estado tendrán la forma $x_B(t) = \bar{x}_B + x_B^\delta(t), \dots, \theta_L(t) = \bar{\theta}_L + \theta_L^\delta(t)$ y las variables de

El vector de fuerzas gravitatorias recoge el efecto de la atracción de la tierra sobre el vehículo, acoplándose tanto a las ecuaciones de la posición como de la carga.

$$g(q^\delta) = \begin{bmatrix} -gm\theta_B^\delta \\ gm\phi_B^\delta \\ 0 \\ 0 \\ 0 \\ 0 \\ glm_L\phi_L^\delta \\ glm_L\theta_L^\delta \end{bmatrix} \quad (3.6)$$

El vector de fuerzas generalizadas contiene los términos relativos al empuje y los pares generados por los rotores.

$$Q = \begin{bmatrix} 0 \\ 0 \\ {}^{(B)}f^T{}^\delta \\ {}^{(B)}\tau_{B_x}^T{}^\delta \\ {}^{(B)}\tau_{B_y}^T{}^\delta \\ {}^{(B)}\tau_{B_z}^T{}^\delta \\ 0 \\ 0 \end{bmatrix} \quad (3.7)$$

La linealización de las ecuaciones conlleva también la desaparición de todos los términos correspondientes a la matriz de fuerzas centrífugas y de Coriolis y sólo quedan los términos asociados a la fricción en el punto de unión en la matriz de resistencia.

$$v(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 \end{bmatrix}_{8 \times 8} \quad (3.8)$$

$$D(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b & 0 & 0 & -b & 0 \\ 0 & 0 & 0 & 0 & b & 0 & 0 & -b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -b & 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 & -b & 0 & 0 & b \end{bmatrix} \quad (3.9)$$

Transformada de Laplace

A partir del modelo linealizado en coordenadas cuerpo y despreciando los términos de fricción, se realiza la transformación de las ecuaciones al dominio de la frecuencia. Para ello, se emplea el método de la transformada de Laplace y se transforman tanto las variables de estado como las variables de entrada al

domino de la frecuencia ($f(s) = \mathcal{L}[f^\delta(t)]$).

$$\begin{aligned} \mathcal{L}[x_B^\delta(t)] &= x_B(s); & \mathcal{L}[\dot{x}_B^\delta(t)] &= s x_B(s) - x_B(0); & \mathcal{L}[\ddot{x}_B^\delta(t)] &= s^2 x_B(s) - s x_B(0) - \dot{x}_B(0); \\ \vdots & & \vdots & & \vdots & \\ \mathcal{L}[\theta_L^\delta(t)] &= \theta_L(s); & \mathcal{L}[\dot{\theta}_L^\delta(t)] &= s \theta_L(s) - \theta_L(0); & \mathcal{L}[\ddot{\theta}_L^\delta(t)] &= s^2 \theta_L(s) - s \theta_L(0) - \dot{\theta}_L(0). \end{aligned} \quad (3.10)$$

$$\begin{aligned} \mathcal{L}[(B)f^T(t)] &= (B)f^T(s); \\ \vdots & \\ \mathcal{L}[(B)\tau_{B_z}^T(t)] &= (B)\tau_{B_z}^T(s). \end{aligned} \quad (3.11)$$

Sustituyendo las expresiones 3.10 y 3.11 en las correspondientes ecuaciones linealizadas y despejando de estas última las variables de salida del sistema, se llega a las ecuaciones deseadas en el domino de Laplace que aportan información adicional sobre la dinámica.

$$\begin{aligned} x_B(s) &= \frac{g}{s^2} \theta_B(s) + \frac{l m_L}{m} \theta_L(s) \\ y_B(s) &= -\frac{g}{s^2} \phi_B(s) - \frac{l m_L}{m} \phi_L(s) \\ z_B(s) &= \frac{(B)f^T(s)}{m s^2} \\ \phi_B(s) &= \frac{1}{I_B^{xx} s^2} (B)\tau_{B_x}^T(s) \\ \theta_B(s) &= \frac{1}{I_B^{yy} s^2} (B)\tau_{B_y}^T(s) \\ \psi_B(s) &= \frac{1}{I_B^{zz} s^2} (B)\tau_{B_z}^T(s) \\ \phi_L(s) &= -\frac{s^2}{(l s^2 + g)} y_B(s) \\ \theta_L(s) &= \frac{s^2}{(l s^2 + g)} x_B(s) \end{aligned} \quad (3.12)$$

A la vista de las ecuaciones anteriores, la posición horizontal del quadrotor o dirección de avance puede ser controlada mediante el ángulo de cabeceo demandado y, este a su vez, mediante el par de actuación correspondiente $(B)\tau_{B_y}^T$. Además, esta posición horizontal se encuentra también influenciada por el movimiento de oscilación de la carga y, en concreto, por el giro en cabeceo de la misma (θ_L). De igual forma, la dirección opuesta será controlada por el ángulo de alabeo (ϕ_B) y condicionada por el respectivo ángulo de oscilación (ϕ_L). Por otro lado, el empuje total del quadrotor manda directamente sobre la altitud de mismo y de forma idéntica ocurre con los pares de alabeo, cabeceo y guiñada y sus respectivos ángulos. Estos pares y el empuje total vendrán generados por cambios en la velocidad de rotación de los cuatro rotores presentes en el vehículo.

Tanto la posición horizontal como los ángulos de la carga podrían ser controlados de forma directa sin más que reescribir las ecuaciones 3.12 en forma de entrada-salida. Estas vendrán dadas por las expresiones descritas a continuación.

$$\begin{aligned}
x_B(s) &= \frac{g}{I_B^{yy} s^4} \frac{(1 + \frac{l}{g} s^2)}{(1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_y}^T \\
y_B(s) &= -\frac{g}{I_B^{xx} s^4} \frac{(1 + \frac{l}{g} s^2)}{(1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_x}^T \\
\phi_L(s) &= \frac{1}{I_B^{xx} s^2 (1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_x}^T \\
\theta_L(s) &= \frac{1}{I_B^{yy} s^2 (1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_y}^T
\end{aligned} \tag{3.13}$$

Expresión Linealizada en el Espacio de Estados

Para obtener el modelo LTI a partir de las ecuaciones linealizadas, se expanden las variables de estado y se desprecia los términos correspondientes a la fricción en el punto de unión. Para ello, se utilizarán además de las variables de estado referidas a posición y ángulo, las equivalentes a velocidad lineal y angular.

$$\begin{aligned}
\frac{d}{dt}x_B &= \dot{x}_B, & \frac{d}{dt}y_B &= \dot{y}_B, & \frac{d}{dt}z_B &= \dot{z}_B, \\
\frac{d}{dt}\phi_B &= \dot{\phi}_B, & \frac{d}{dt}\theta_L &= \dot{\theta}_L, & \frac{d}{dt}\psi_L &= \dot{\psi}_L, \\
\frac{d}{dt}\phi_L &= \dot{\phi}_L, & \frac{d}{dt}\theta_L &= \dot{\theta}_L.
\end{aligned} \tag{3.14}$$

$$x = [x_B, y_B, z_B, \phi_B, \theta_B, \psi_B, \phi_L, \theta_L, \dot{x}_B, \dot{y}_B, \dot{z}_B, \dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B, \dot{\phi}_L, \dot{\theta}_L]^T$$

Las variables sobre las que se actúa para cambiar la posición y orientación de vehículo, no se verán afectadas en la obtención del modelo LTI.

$$u = [{}^{(B)}f^T, {}^{(B)}\tau_{B_x}^T, {}^{(B)}\tau_{B_y}^T, {}^{(B)}\tau_{B_z}^T] \tag{3.15}$$

Definidos el vector de estados y de variables de actuación, falta establecer el vector salida del sistema que cierra el problema.

$$y = [x_B, y_B, z_B, \phi_L, \theta_L] \tag{3.16}$$

Y, finalmente, una representación en el espacio de estados en forma matricial es dada.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \tag{3.17}$$

Donde la derivada del vector de estados se define como $\dot{x} = \frac{d}{dt}x$, y las matrices en derivadas parciales son $A = \left. \frac{\partial f(x,u)}{\partial x} \right|_{\bar{x}, \bar{u}}$ y $B = \left. \frac{\partial f(x,u)}{\partial u} \right|_{\bar{x}, \bar{u}}$.

$$A = \left[\begin{array}{cccc|cccc} & & & & 0_{8 \times 8} & & & I_{8 \times 8} \\ \hline 0 & 0 & 0 & 0 & g \frac{m}{m_B} & 0 & 0 & -g \frac{m_L}{m_B} \\ 0 & 0 & 0 & -g \frac{m}{m_B} & 0 & 0 & g \frac{m_L}{m_B} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g \frac{m}{l m_B} & 0 & 0 & -g \frac{m}{l m_B} & 0 \\ 0 & 0 & 0 & 0 & g \frac{m}{l m_B} & 0 & 0 & -g \frac{m}{l m_B} \end{array} \right] \quad 0_{8 \times 8} \quad (3.18)$$

$$B = \left[\begin{array}{cccc} & & & 0_{8 \times 4} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_B^{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_B^{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_B^{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right] \quad (3.19)$$

$$C = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad 0_{5 \times 8} \quad (3.20)$$

3.1.2 Coordenadas Carga

Para linealizar las ecuaciones dinámicas, expresadas en coordenadas carga, se eligió también como punto de operación aquel donde el vehículo se encontraba en reposo y la carga suspendida sin oscilación.

$$\begin{aligned} \bar{x}_L &= \bar{y}_L = \bar{z}_L = cte. \\ \bar{\phi}_B &= \bar{\theta}_B = \bar{\psi}_B = 0 \\ \bar{\phi}_L &= \bar{\theta}_L = 0 \end{aligned} \quad (3.21)$$

$$\bar{q}_L = [\bar{x}_L, \bar{y}_L, \bar{z}_L, 0, 0, 0, 0, 0]^T$$

En el vector de fuerzas gravitatorias, sólo cambian los términos asociados a los ángulos de la carga, añadiendo ahora el acoplamiento de los ángulos de inclinación del quadrotor.

$$g(q^\delta) = \begin{bmatrix} -gm\theta_B^\delta \\ gm\phi_B^\delta \\ 0 \\ 0 \\ 0 \\ 0 \\ glm_L\phi_L^\delta - glm\phi_B^\delta \\ glm_L\theta_L^\delta - glm\theta_B^\delta \end{bmatrix} \quad (3.26)$$

El vector de fuerzas generalizadas, que depende de las fuerzas y pares de sustentación, es idéntico para ambas formulaciones.

$$Q = \begin{bmatrix} 0 \\ 0 \\ {}^{(B)}f^T \delta \\ {}^{(B)}\tau_{B_x}^T \delta \\ {}^{(B)}\tau_{B_y}^T \delta \\ {}^{(B)}\tau_{B_z}^T \delta \\ 0 \\ 0 \end{bmatrix} \quad (3.27)$$

Tanto la matriz de fuerzas centrífugas y de coriolis como la de resistencia son nulas, salvo los términos asociados a la fricción en el punto de unión en esta última.

$$v(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 \end{bmatrix}_{8 \times 8} \quad (3.28)$$

$$D(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b & 0 & 0 & -b & 0 \\ 0 & 0 & 0 & 0 & b & 0 & 0 & -b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -b & 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 & -b & 0 & 0 & b \end{bmatrix} \quad (3.29)$$

Transformada de Laplace

Partiendo del modelo linealizado en coordenadas carga, representado en la expresión 3.23, se realiza la transformación de las ecuaciones al dominio de la frecuencia mediante la transformada de Laplace. En ellas, se ha considerado despreciable el efecto de la fricción sobre el punto de unión cuerpo-carga.

$$\begin{aligned}
x_L(s) &= \frac{g}{s^2} \theta_B(s) - \frac{lm_B}{m} \theta_L(s) \\
y_L(s) &= -\frac{g}{s^2} \phi_B(s) + \frac{lm_B}{m} \phi_L(s) \\
z_L(s) &= \frac{{}^{(B)}f^T(s)}{ms^2} \\
\phi_B(s) &= \frac{1}{I_B^{xx} s^2} {}^{(B)}\tau_{B_x}^T(s) \\
\theta_B(s) &= \frac{1}{I_B^{yy} s^2} {}^{(B)}\tau_{B_y}^T(s) \\
\psi_B(s) &= \frac{1}{I_B^{zz} s^2} {}^{(B)}\tau_{B_z}^T(s) \\
\phi_L(s) &= \frac{m}{m_L} \frac{(\phi_B(s) + \frac{m_B}{gm} s^2 y_L(s))}{(1 + \frac{lm_B}{gm_L} s^2)} \\
\theta_L(s) &= \frac{m}{m_L} \frac{(\theta_B(s) - \frac{m_B}{gm} s^2 x_L(s))}{(1 + \frac{lm_B}{gm_L} s^2)}
\end{aligned} \tag{3.30}$$

Al igual que en coordenadas cuerpo, se puede controlar directamente la altitud del vehículo y su actitud en alabeo, cabeceo y guiñada mediante el empuje total y los pares correspondientes. La posición horizontal de la carga dependerá del ángulo de inclinación correspondiente y se verá afectada por la oscilación de la misma, estando la dirección “x” acoplada con el ángulo de cabeceo y la dirección “y” con el ángulo de alabeo.

Finalmente, se presenta la posibilidad de controlar la posición horizontal de la carga o los ángulos de inclinación de la misma a través de los momentos de alabeo y cabeceo generados por los rotores.

$$\begin{aligned}
x_L(s) &= \frac{g}{I_B^{yy} s^4} \frac{1}{(1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_y}^T \\
y_L(s) &= -\frac{g}{I_B^{xx} s^4} \frac{1}{(1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_x}^T \\
\phi_L(s) &= \frac{1}{I_B^{xx} s^2 (1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_x}^T \\
\theta_L(s) &= \frac{1}{I_B^{yy} s^2 (1 + \frac{lm_B}{gm} s^2)} {}^{(B)}\tau_{B_y}^T
\end{aligned} \tag{3.31}$$

Expresión Linealizada en el Espacio de Estados

El modelo LTI en coordenadas carga se obtiene tras expandir las variables de estado a partir de las ecuaciones vistas en 3.23. Al igual que en el dominio de Laplace, se desprecia el efecto de la fricción en el punto de unión para generar un modelo más simplificado con el que poder operar posteriormente. Estas nuevas variables incluyen además la velocidad lineal y angular de la carga o el vehículo.

$$x = [x_L, y_L, z_L, \phi_B, \theta_B, \psi_B, \dot{\phi}_L, \dot{\theta}_L, \dot{z}_L, \dot{\phi}_B, \dot{\theta}_B, \dot{\psi}_B, \dot{\phi}_L, \dot{\theta}_L]^T \tag{3.32}$$

El vector de actuación y el vector de salida no se expandirán y la única variación que experimentan es el cambio de coordenadas cuerpo a carga.

$$u = [{}^{(B)}f^T, {}^{(B)}\tau_{B_x}^T, {}^{(B)}\tau_{B_y}^T, {}^{(B)}\tau_{B_z}^T] \quad (3.33)$$

$$y = [x_L, y_L, z_L, \phi_L, \theta_L] \quad (3.34)$$

Descritas todas las variables que intervienen en el problema, la representación en el espacio de estados es posible y se visualiza a continuación.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (3.35)$$

Las matrices A, B y C, se calculan como las derivadas parciales de las ecuaciones dinámicas respecto a los vectores de estado y de actuación, llevadas al punto de equilibrio.

$$A = \left[\begin{array}{c|c} 0_{8 \times 8} & I_{8 \times 8} \\ \hline \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & g \\ 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & g \frac{m}{lm_B} & 0 & 0 & -g \frac{m}{lm_B} & 0 \\ 0 & 0 & 0 & 0 & g \frac{m}{lm_B} & 0 & 0 & -g \frac{m}{lm_B} \end{array} & 0_{8 \times 8} \end{array} \right] \quad (3.36)$$

$$B = \left[\begin{array}{c|c} 0_{8 \times 4} & \\ \hline \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{I_B^{xx}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_B^{yy}} & 0 \\ 0 & 0 & 0 & \frac{1}{I_B^{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} & \end{array} \right] \quad (3.37)$$

$$C = \left[\begin{array}{c|c} \begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} & 0_{5 \times 8} \end{array} \right] \quad (3.38)$$

3.2 Velocidad Constante

En esta sección se pretende linealizar las ecuaciones del movimiento considerando que el vehículo se desplaza a una velocidad constante ya sea vertical u horizontal. Esta linealización complementa a la de velocidad nula y ha sido tomada en cuenta por otros autores como [27], donde se describe una linealización para velocidad nula, velocidad vertical constante y ángulo de inclinación constante. Para su cálculo, se parte de las ecuaciones formuladas a partir de Lagrange y se considera el efecto de la resistencia aerodinámica. Este si será importante para el problema de velocidad constante a diferencia del de velocidad nula, donde su efecto desaparecería al linealizar las ecuaciones. En cambio, la resistencia debida a la fricción en el punto de unión no será tomada en cuenta por el momento.

3.2.1 Velocidad Vertical Constante

Coordenadas Cuerpo

La linealización de las ecuaciones para velocidad vertical constante, se realiza eligiendo como punto de equilibrio aquel en el que la velocidad vertical del vehículo es constante, mientras que la velocidad horizontal del mismo permanece nula. Se suponen también despreciables el resto de velocidades angulares del vehículo, incluida las de oscilación de la carga.

$$\begin{aligned}\dot{x}_B &= \dot{y}_B = 0, & \dot{z}_B &= cte. \\ \dot{\phi}_B &= \dot{\theta}_B = \dot{\psi}_B = 0 \\ \dot{\phi}_L &= \dot{\theta}_L = 0 \\ \dot{q}_B &= [0, 0, \dot{z}_B, 0, 0, 0, 0, 0]^T\end{aligned}\tag{3.39}$$

Sustituyendo los valores de velocidad establecidos y anulando las aceleraciones en las ecuaciones dinámicas, se consiguen los siguientes valores para la posición de equilibrio.

$$\begin{aligned}\bar{x}_B &= \bar{y}_B = cte, & \bar{z}_B(t) &= \dot{z}_B t + \bar{z}_B(0) \\ \bar{\phi}_B &= \bar{\theta}_B = \bar{\psi}_B = 0 \\ \bar{\phi}_L &= \bar{\theta}_L = 0 \\ \bar{q}_B &= [\bar{x}_B, \bar{y}_B, \bar{z}_B(t), 0, 0, 0, 0, 0]^T\end{aligned}\tag{3.40}$$

Donde el valor de ψ_B es un valor arbitrario al igual que la posición inicial pero se ha tomado nulo por comodidad y porque no afecta de forma significativa al movimiento de ascenso y descenso del vehículo. En cambio, los ángulos de inclinación son nulos para que pueda cumplirse el equilibrio en las direcciones horizontales y lo mismo sucede con los ángulos de la carga que se mantiene suspendida del vehículo sin apenas oscilación. Los valores en el equilibrio para las variables de entrada o actuación experimentan también variaciones con respecto al modelo de velocidad nula.

$$\begin{aligned}{}^{(B)}\bar{f}^T &= gm + (C_{B_z} + C_L)|\dot{z}_B|\dot{z}_B \\ {}^{(B)}\bar{\tau}_{B_x}^T &= {}^{(B)}\bar{\tau}_{B_y}^T = {}^{(B)}\bar{\tau}_{B_z}^T = 0 \\ \bar{\tau} &= [{}^{(B)}\bar{f}^T, 0, 0, 0]^T\end{aligned}\tag{3.41}$$

Introduciendo ahora pequeñas variaciones alrededor de la posición de equilibrio.

$$\begin{aligned}
 x_B(t) &= \bar{x}_B + x_B^\delta(t), & y_B(t) &= \bar{y}_B + y_B^\delta(t), & z_B(t) &= \dot{\bar{z}}_B t + \bar{z}_B(0) + z_B^\delta(t). \\
 \phi_B(t) &= \bar{\phi}_B + \phi_B^\delta(t), & \theta_B(t) &= \bar{\theta}_B + \theta_B^\delta(t), & \psi_B(t) &= \bar{\psi}_B + \psi_B^\delta(t). \\
 \phi_L(t) &= \bar{\phi}_L + \phi_L^\delta(t), & \theta_L(t) &= \bar{\theta}_L + \theta_L^\delta(t).
 \end{aligned} \tag{3.42}$$

Y lo mismo para las variables de entrada al sistema.

$$\begin{aligned}
 {}^{(B)}f^T(t) &= gm + (C_{Bz} + C_L)|\dot{\bar{z}}_B|\dot{\bar{z}}_B + {}^{(B)}f^{T\delta}(t) \\
 {}^{(B)}\tau_{B_x}^T(t) &= {}^{(B)}\bar{\tau}_{B_x}^T + {}^{(B)}\tau_{B_x}^{T\delta}(t) \\
 {}^{(B)}\tau_{B_y}^T(t) &= {}^{(B)}\bar{\tau}_{B_y}^T + {}^{(B)}\tau_{B_y}^{T\delta}(t) \\
 {}^{(B)}\tau_{B_z}^T(t) &= {}^{(B)}\bar{\tau}_{B_z}^T + {}^{(B)}\tau_{B_z}^{T\delta}(t)
 \end{aligned} \tag{3.43}$$

Se obtienen finalmente las ecuaciones del movimiento linealizadas para velocidad vertical constante, aplicado el desarrollo de Taylor entorno al punto de operación elegido.

$$\begin{aligned}
 m\ddot{x}_B^\delta - lm_L\ddot{\theta}_L^\delta - gm\theta_B^\delta - (C_{Bz} + C_L)|\dot{\bar{z}}_B|\dot{\bar{z}}_B\theta_B^\delta &= 0 \\
 m\ddot{y}_B^\delta + lm_L\ddot{\phi}_L^\delta + gm\phi_B^\delta + (C_{Bz} + C_L)|\dot{\bar{z}}_B|\dot{\bar{z}}_B\phi_B^\delta &= 0 \\
 m\ddot{z}_B^\delta - ({}^{(B)}f^T)^\delta + 2(C_{Bz} + C_L)|\dot{\bar{z}}_B|\dot{\bar{z}}_B^\delta &= 0 \\
 I_B^{xx}\ddot{\phi}_B^\delta - {}^{(B)}\tau_{B_x}^{T\delta} &= 0 \\
 I_B^{yy}\ddot{\theta}_B^\delta - {}^{(B)}\tau_{B_y}^{T\delta} &= 0 \\
 I_B^{zz}\ddot{\psi}_B^\delta - {}^{(B)}\tau_{B_z}^{T\delta} &= 0 \\
 lm_L\ddot{y}_B^\delta + l^2m_L\ddot{\phi}_L^\delta + glm_L\phi_L^\delta + (C_Ll|\dot{\bar{z}}_B|\dot{\bar{z}}_B)\phi_L^\delta &= 0 \\
 -lm_L\ddot{x}_B^\delta + l^2m_L\ddot{\theta}_L^\delta + glm_L\theta_L^\delta + (C_Ll|\dot{\bar{z}}_B|\dot{\bar{z}}_B)\theta_L^\delta &= 0
 \end{aligned} \tag{3.44}$$

Comparando las ecuaciones linealizadas para velocidad constante con las obtenidas en 3.3 para velocidad nula, la principal diferencia se encuentra en la adición de un término de resistencia aerodinámica en dirección vertical. Este término se introduce como consecuencia del equilibrio de fuerzas en el punto de operación, teniendo la fuerza de sustentación que compensar no sólo la inercia del vehículo sino también la resistencia aerodinámica del cuerpo. En menor media, se encuentran también afectadas las direcciones horizontales a través de los ángulos de inclinación correspondientes y, por acoplamiento, los ángulos de la carga.

Las matrices de masa y gravedad coinciden con las expresiones 3.5 y 3.6 para el modelo linealizado considerando velocidad nula y expresado en coordenadas cuerpo. Las matrices de fuerzas generalizas y resistencia, por el contrario, si experimentan cambios respecto a dicho modelo y se dibujarán a continuación.

$$Q = \begin{bmatrix} (C_{Bz} + C_L)|\dot{z}_B|\dot{z}_B\theta_B^\delta \\ -(C_{Bz} + C_L)|\dot{z}_B|\dot{z}_B\phi_B^\delta \\ {}^{(B)}f^T \\ {}^{(B)}\tau_{Bx}^T \\ {}^{(B)}\tau_{By}^T \\ {}^{(B)}\tau_{Bz}^T \\ -C_L l |\dot{z}_B|\dot{z}_B\phi_L^\delta \\ -C_L l |\dot{z}_B|\dot{z}_B\theta_L^\delta \end{bmatrix} \quad (3.45)$$

$$D(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(C_{Bz} + C_L)|\dot{z}_B| & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.46)$$

Coordenadas Carga

El punto de operación tomado para linealizar las ecuaciones expresadas en coordenadas carga es idéntico al elegido en el apartado anterior. Su posición estará expresada ahora en coordenadas de la carga en lugar de coordenadas cuerpo, manteniéndose la actitud del vehículo y los ángulos de la carga como variables de estado.

$$\begin{aligned} \dot{x}_L = \dot{y}_L = 0, \quad \dot{z}_L = cte. \\ \dot{\phi}_B = \dot{\theta}_B = \dot{\psi}_B = 0 \\ \dot{\phi}_L = \dot{\theta}_L = 0 \end{aligned} \quad (3.47)$$

$$\dot{q}_L = [0, 0, \dot{z}_L, 0, 0, 0, 0, 0]^T$$

Los nuevos valores para la posición y los ángulos en el equilibrio serán idénticos a los ya visto en coordenadas cuerpo, cambiando la posición del cuerpo por la de la carga.

$$\begin{aligned} \bar{x}_L = \bar{y}_L = cte, \quad \bar{z}_L(t) = \dot{z}_L t + \bar{z}_L(0) \\ \bar{\phi}_B = \bar{\theta}_B = \bar{\psi}_B = 0 \\ \bar{\phi}_L = \bar{\theta}_L = 0 \end{aligned} \quad (3.48)$$

$$\bar{q}_L = [\bar{x}_L, \bar{y}_L, \bar{z}_L(t), 0, 0, 0, 0, 0]^T$$

Igualmente, se toma un valor de ψ_B nulo para simplificar aún mas las ecuaciones del movimiento ya que no afecta al movimiento en dirección vertical. El único valor de entrada al sistema que se ve afectado es el empuje total ejercido por los rotores que incluye ahora la resistencia aerodinámica tanto del cuerpo como de la carga.

$$\begin{aligned}
{}^{(B)}\bar{f}^T &= gm + (C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L \\
{}^{(B)}\bar{\tau}_{B_x}^T &= {}^{(B)}\bar{\tau}_{B_y}^T = {}^{(B)}\bar{\tau}_{B_z}^T = 0
\end{aligned} \tag{3.49}$$

$$\bar{\tau} = [{}^{(B)}\bar{f}^T, 0, 0, 0]^T$$

Introduciendo valores incrementales entorno al punto de equilibrio calculado y desarrollando dichas ecuaciones mediante series de Taylor, se calcula las ecuaciones linealizadas deseadas en coordenadas carga.

$$\begin{aligned}
m\ddot{x}_L^\delta + lm_B\ddot{\theta}_L^\delta - gm\theta_B^\delta - (C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L\theta_B^\delta &= 0 \\
m\ddot{y}_L^\delta - lm_B\ddot{\phi}_L^\delta + gm\phi_B^\delta + (C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L\phi_B^\delta &= 0 \\
m\ddot{z}_L^\delta - ({}^{(B)}\bar{f}^T)^\delta + 2(C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L^\delta &= 0 \\
I_B^{xx}\ddot{\phi}_B^\delta - ({}^{(B)}\bar{\tau}_{B_x}^T)^\delta &= 0 \\
I_B^{yy}\ddot{\theta}_B^\delta - ({}^{(B)}\bar{\tau}_{B_y}^T)^\delta &= 0 \\
I_B^{zz}\ddot{\psi}_B^\delta - ({}^{(B)}\bar{\tau}_{B_z}^T)^\delta &= 0
\end{aligned} \tag{3.50}$$

$$\begin{aligned}
-lm_B\ddot{y}_L^\delta + l^2m_B\ddot{\phi}_L^\delta - glm\phi_B^\delta + glm_L\phi_L^\delta - (C_L + C_{Bz})l|\dot{z}_L|\dot{z}_L\phi_B^\delta + (C_Ll|\dot{z}_L|\dot{z}_L)\phi_L^\delta &= 0 \\
lm_B\ddot{x}_L^\delta + l^2m_B\ddot{\theta}_L^\delta - glm\theta_B^\delta + glm_L\theta_L^\delta - (C_L + C_{Bz})l|\dot{z}_L|\dot{z}_L\theta_B^\delta + (C_Ll|\dot{z}_L|\dot{z}_L)\theta_L^\delta &= 0
\end{aligned}$$

Las tres ecuaciones referidas a la actitud del vehículo son idénticas y dará igual expresarlas en coordenadas cuerpo o coordenadas carga. Sin embargo, las tres ecuaciones referidas a la posición de la carga si son distintas pues presentan la ventaja de poder controlar directamente la posición de la carga y no la del vehículo. Lo mismo sucede con las dos ecuaciones que describen el movimiento de oscilación de la carga, cuyos términos inerciales son distintos e incorporan la influencia de los ángulos de inclinación del cuerpo.

Tanto la matriz de masa como la matriz de fuerzas generalizadas coinciden con las matrices calculadas para velocidad nula en coordenadas carga. Estas venían dadas por las expresiones 3.25 y 3.26 respectivamente. Las matrices de fuerzas generalizadas y resistencia si se verán afectadas por la inclusión de términos relacionados con la resistencia aerodinámica en dirección vertical.

$$Q = \begin{bmatrix} (C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L\theta_B^\delta \\ -(C_{Bz} + C_L)|\dot{z}_L|\dot{z}_L\phi_B^\delta \\ ({}^{(B)}\bar{f}^T)^\delta \\ ({}^{(B)}\bar{\tau}_{B_x}^T)^\delta \\ ({}^{(B)}\bar{\tau}_{B_y}^T)^\delta \\ ({}^{(B)}\bar{\tau}_{B_z}^T)^\delta \\ -C_Ll|\dot{z}_L|\dot{z}_L\phi_L^\delta + (C_{Bz} + C_L)l|\dot{z}_L|\dot{z}_L\phi_B^\delta \\ -C_Ll|\dot{z}_L|\dot{z}_L\theta_L^\delta + (C_{Bz} + C_L)l|\dot{z}_L|\dot{z}_L\theta_B^\delta \end{bmatrix} \tag{3.51}$$

$$D(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2(C_{Bz} + C_L)|\dot{z}_L| & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.52)$$

3.2.2 Velocidad Horizontal Constante

Coordenadas Cuerpo

El punto de operación elegido para este caso cumple con que velocidad horizontal tanto en dirección “x” como “y” es constante, siendo nula el resto de velocidades asociadas a las variables de estudio. Es decir, tenemos un vehículo desplazándose horizontalmente a una altitud constante y con un determinado ángulo de inclinación determinado por la velocidad de desplazamiento.

$$\begin{aligned} \dot{x}_B &= \dot{y}_B = cte, \quad \dot{z}_B = 0, \\ \dot{\phi}_B &= \dot{\theta}_B = \dot{\psi}_B = 0, \\ \dot{\phi}_L &= \dot{\theta}_L = 0, \\ \dot{q}_B &= [\dot{x}_B, \dot{y}_B, 0, 0, 0, 0, 0, 0]^T. \end{aligned} \quad (3.53)$$

Establecido el equilibrio dinámico para velocidad horizontal constante, se obtienen los valores característicos de las variables en el punto de operación. El acoplamiento entre las direcciones horizontales y los ángulos de inclinación se vuelve a poner de manifiesto en el valor tomado por estos últimos para compensar las fuerzas de resistencia aerodinámica en dirección horizontal. Dado que la actitud de guiñada es independiente del resto, se toma un valor nulo para la misma simplificando enormemente el cálculo del resto de variables.

$$\begin{aligned} \bar{x}_B(t) &= \dot{x}_B t + \bar{x}_B(0), & \bar{y}_B(t) &= \dot{y}_B t + \bar{y}_B(0), & \bar{z}_B &= cte, \\ \bar{\phi}_B &= \arctan\left(-\frac{(C_{Bxy} + C_L)|\dot{y}_B|\dot{x}_B \cos \bar{\theta}_B}{gm}\right), & \bar{\theta}_B &= \arctan\left(\frac{(C_{Bxy} + C_L)|\dot{x}_B|\dot{y}_B}{gm}\right), & \bar{\psi}_B &= 0, \\ \bar{\phi}_L &= \arctan\left(-\frac{C_L l |\dot{y}_B|\dot{x}_B}{(C_L l \sin \bar{\theta}_L |\dot{x}_B|\dot{y}_B + g l m_L \cos \bar{\theta}_L)}\right), & \bar{\theta}_L &= \arctan\left(\frac{C_L l |\dot{x}_B|\dot{y}_B}{g l m_L}\right). \end{aligned} \quad (3.54)$$

$$\bar{q}_B = [\dot{x}_B t + \bar{x}_B(0), \dot{y}_B t + \bar{y}_B(0), \bar{z}_B, \bar{\phi}_B, \bar{\theta}_B, 0, \bar{\phi}_L, \bar{\theta}_L]^T \quad (3.55)$$

Para las variables de entrada al sistema en el punto de operación, se tienen los siguientes resultados.

$$\begin{aligned} {}^{(B)}\bar{f}^T &= \frac{gm}{\cos \bar{\phi}_B \cos \bar{\theta}_B} \\ {}^{(B)}\bar{\tau}_{B_x}^T &= {}^{(B)}\bar{\tau}_{B_y}^T = {}^{(B)}\bar{\tau}_{B_z}^T = 0 \\ \bar{\tau} &= [{}^{(B)}\bar{f}^T, 0, 0, 0]^T \end{aligned} \quad (3.56)$$

De forma similar al caso de velocidad vertical constante, se introducen pequeñas oscilaciones entorno a los puntos de equilibrio obtenidos previamente y se linealiza las ecuaciones dinámicas mediante series de Taylor alrededor de dicho punto. Con el fin de evitar unas ecuaciones muy extensas y difíciles de visualizar, se presenta el modelo matricial lineal en coordenadas cuerpo para el caso de velocidad horizontal constante.

$$M(q^\delta)\ddot{q}^\delta + v(q^\delta, \dot{q}^\delta)\dot{q}^\delta + D(q^\delta, \dot{q}^\delta)\dot{q}^\delta + g(q^\delta) = Q^\delta \quad (3.57)$$

La matriz de masa (M) contiene los términos constantes asociados a la masa o inercia del sistema. Sus valores están calculados para el punto de equilibrio y vienen determinados a partir de las expresiones recogidas en 3.54.

$$M(q^\delta) = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 \\ 0 & 0 & 0 & I_B^{xx} & 0 \\ 0 & 0 & 0 & 0 & I_B^{zz} + (I_B^{yy} - I_B^{zz})c^2\bar{\phi}_B \\ 0 & 0 & 0 & -I_B^{xx}s\bar{\theta}_B & (I_B^{yy} - I_B^{zz})c\bar{\phi}_Bc\bar{\theta}_Bs\bar{\phi}_B \\ lm_Ls\bar{\phi}_Ls\bar{\theta}_L & lm_Lc\bar{\phi}_L & lm_Ls\bar{\phi}_Lc\bar{\theta}_L & 0 & 0 \\ -lm_Lc\bar{\phi}_Lc\bar{\theta}_L & 0 & lm_Lc\bar{\phi}_Ls\bar{\theta}_L & 0 & 0 \end{bmatrix} \quad (3.58)$$

$$\begin{bmatrix} 0 & lm_Ls\bar{\phi}_Ls\bar{\theta}_L & -lm_Lc\bar{\phi}_Lc\bar{\theta}_L \\ 0 & lm_Lc\bar{\phi}_L & 0 \\ 0 & lm_Ls\bar{\phi}_Lc\bar{\theta}_L & lm_Lc\bar{\phi}_Ls\bar{\theta}_L \\ -I_B^{xx}s\bar{\theta}_B & 0 & 0 \\ (I_B^{yy} - I_B^{zz})c\bar{\phi}_Bc\bar{\theta}_Bs\bar{\phi}_B & 0 & 0 \\ (I_B^{xx} + (I_B^{yy} - I_B^{xx})c^2\bar{\theta}_B + (I_B^{zz} - I_B^{yy})c^2\bar{\phi}_Bc^2\bar{\theta}_B) & 0 & 0 \\ 0 & l^2m_L & 0 \\ 0 & 0 & l^2m_Lc^2\bar{\phi}_L \end{bmatrix}$$

El vector gravedad (g) contiene todos los elementos linealizados asociados a las fuerzas de atracción gravitatoria, que dependerán a su vez de la actitud del vehículo para el caso de la posición y de la carga para los ángulos de la misma.

$$g(q^\delta) = \begin{bmatrix} (gm \tan \bar{\phi}_B \tan \bar{\theta}_B)\phi_B^\delta - (gm)\theta_B^\delta - (gm \frac{\tan \bar{\phi}_B}{c\bar{\theta}_B})\psi_B^\delta \\ (\frac{gm}{c\bar{\theta}_B})\phi_B^\delta - (gm \tan \bar{\theta}_B)\psi_B^\delta \\ (gm \tan \bar{\phi}_B)\phi_B^\delta + (gm \tan \bar{\theta}_B)\theta_B^\delta \\ 0 \\ 0 \\ 0 \\ (glm_Lc\bar{\theta}_Lc\bar{\phi}_L)\phi_L^\delta - (glm_Ls\bar{\theta}_Ls\bar{\phi}_L)\theta_L^\delta \\ -(glm_Ls\bar{\phi}_Ls\bar{\theta}_L)\phi_L^\delta + (glm_Lc\bar{\phi}_Lc\bar{\theta}_L)\theta_L^\delta \end{bmatrix} \quad (3.59)$$

El vector de fuerzas generalizadas recoge las variables relacionadas con el empuje y los pares creados por los rotores. Incluye, además, términos de resistencia aerodinámica dependientes de los ángulos de oscilación de la carga que serán incluidos aquí porque la matriz de resistencia tiene sólo en cuenta valores relacionados con la velocidad angular.

$$Q = \begin{bmatrix} c\bar{\phi}_B s\bar{\theta}_B^{(B)} f^T)^{\delta} \\ -s\bar{\phi}_B^{(B)} f^T)^{\delta} \\ c\bar{\phi}_B c\bar{\theta}_B^{(B)} f^T)^{\delta} \\ {}^{(B)}\tau_{B_x}^T)^{\delta} \\ c\bar{\phi}_B^{(B)} \tau_{B_y}^T)^{\delta} - s\bar{\phi}_B^{(B)} \tau_{B_z}^T)^{\delta} \\ -s\bar{\theta}_B^{(B)} \tau_{B_x}^T)^{\delta} + c\bar{\theta}_B s\bar{\phi}_B^{(B)} \tau_{B_y}^T)^{\delta} + c\bar{\phi}_B c\bar{\theta}_B^{(B)} \tau_{B_z}^T)^{\delta} \\ C_L l s\bar{\phi}_L |\dot{\hat{y}}_B| \dot{\hat{y}}_B \phi_L^{\delta} - C_L l c\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| \dot{\hat{x}}_B \phi_L^{\delta} - C_L l s\bar{\phi}_L c\bar{\theta}_L |\dot{\hat{x}}_B| \dot{\hat{x}}_B \theta_L^{\delta} \\ -C_L l s\bar{\phi}_L c\bar{\theta}_L |\dot{\hat{x}}_B| \dot{\hat{x}}_B \phi_L^{\delta} - C_L l c\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| \dot{\hat{x}}_B \theta_L^{\delta} \end{bmatrix} \quad (3.60)$$

Esta linealización supone también que todos los los términos asociados a la matriz de fuerzas centrífugas y de Coriolis sean nulos pero no sucede así con los incluidos en la matriz de resistencia, que dependen de la resistencia aerodinámica y de la velocidad de operación.

$$v(q^{\delta}, \dot{q}^{\delta}) = \begin{bmatrix} 0 \end{bmatrix}_{8 \times 8} \quad (3.61)$$

$$D(q^{\delta}, \dot{q}^{\delta}) = \begin{bmatrix} 2(C_{B_{xy}} + C_L) |\dot{\hat{x}}_B| & 0 & 0 & 0 & 0 & 0 \\ 0 & 2(C_{B_{xy}} + C_L) |\dot{\hat{y}}_B| & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2C_L l s\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| & 2C_L l c\bar{\phi}_L |\dot{\hat{y}}_B| & 0 & 0 & 0 & 0 \\ -2C_L l c\bar{\phi}_L c\bar{\theta}_L |\dot{\hat{x}}_B| & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \dots \quad (3.62)$$

$$\begin{bmatrix} 2C_L l s\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| & -2C_L l c\bar{\phi}_L c\bar{\theta}_L |\dot{\hat{x}}_B| \\ 2C_L l c\bar{\phi}_L |\dot{\hat{y}}_B| & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2C_L l^2 s^2 \bar{\phi}_L s^2 \bar{\theta}_L |\dot{\hat{x}}_B| + 2C_L l^2 c^2 \bar{\phi}_L |\dot{\hat{y}}_B| & -2C_L l^2 c\bar{\phi}_L c\bar{\theta}_L s\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| \\ -2C_L l^2 c\bar{\phi}_L c\bar{\theta}_L s\bar{\phi}_L s\bar{\theta}_L |\dot{\hat{x}}_B| & 2C_L l^2 c^2 \bar{\phi}_L c^2 \bar{\theta}_L |\dot{\hat{x}}_B| \end{bmatrix}$$

Coordenadas Carga

Se toma el mismo punto de operación que el empleado para coordenadas cuerpo con la diferencia de que la posición característica será la de la carga y no la del cuerpo. Este equilibrio puede verse como un desplazamiento constante de la carga en dirección horizontal sin variaciones en su altitud.

$$\begin{aligned} \dot{\hat{x}}_L &= \dot{\hat{y}}_L = cte, \quad \dot{\hat{z}}_L = 0, \\ \dot{\hat{\phi}}_B &= \dot{\hat{\theta}}_B = \dot{\hat{\psi}}_B = 0, \\ \dot{\hat{\phi}}_L &= \dot{\hat{\theta}}_L = 0, \\ \dot{\hat{q}}_L &= [\dot{\hat{x}}_L, \dot{\hat{y}}_L, 0, 0, 0, 0, 0]^T. \end{aligned} \quad (3.63)$$

Tanto los ángulos de inclinación del vehículo como los de la carga quedarán fijados a un determinado

valor que vendrá determinado por la velocidad horizontal de diseño y por la resistencia aerodinámica en su respectiva dirección.

$$\begin{aligned}\bar{x}_L(t) &= \dot{\bar{x}}_L t + \bar{x}_L(0), & \bar{y}_L(t) &= \dot{\bar{y}}_L t + \bar{y}_L(0), & \bar{z}_L &= cte, \\ \bar{\phi}_B &= \arctan\left(-\frac{(C_{B_{xy}}+C_L)|\dot{\bar{y}}_L|\dot{\bar{y}}_L \cos \bar{\theta}_B}{gm}\right), & \bar{\theta}_B &= \arctan\left(\frac{(C_{B_{xy}}+C_L)|\dot{\bar{x}}_L|\dot{\bar{x}}_L}{gm}\right), & \bar{\psi}_B &= 0, \\ \bar{\phi}_L &= \arctan\left(-\frac{C_L l |\dot{\bar{y}}_L|\dot{\bar{y}}_L}{(C_L l \sin \bar{\theta}_L |\dot{\bar{x}}_L|\dot{\bar{x}}_L + g l m_L \cos \bar{\theta}_L)}\right), & \bar{\theta}_L &= \arctan\left(\frac{C_L l |\dot{\bar{x}}_L|\dot{\bar{x}}_L}{g l m_L}\right).\end{aligned}\quad (3.64)$$

$$\bar{q}_L = [\dot{\bar{x}}_L t + \bar{x}_L(0), \dot{\bar{y}}_L t + \bar{y}_L(0), \bar{z}_L, \bar{\phi}_B, \bar{\theta}_B, 0, \bar{\phi}_L, \bar{\theta}_L]^T \quad (3.65)$$

El empuje total en el equilibrio será idéntico al ya obtenido para coordenadas cuerpo y los pares de alabeo, cabeceo y guiñada serán nulos.

$$\begin{aligned}{}^{(B)}\bar{f}^T &= \frac{gm}{\cos \bar{\phi}_B \cos \bar{\theta}_B} \\ {}^{(B)}\bar{\tau}_{B_x}^T &= {}^{(B)}\bar{\tau}_{B_y}^T = {}^{(B)}\bar{\tau}_{B_z}^T = 0 \\ \bar{\tau} &= [{}^{(B)}\bar{f}^T, 0, 0, 0]^T\end{aligned}\quad (3.66)$$

Obtenido el punto de operación para velocidad horizontal constante, se linealizan las ecuaciones mediante series de Taylor alrededor de dicho punto y se presenta su resultado en forma matricial para una mejor visualización del mismo.

$$M(q^\delta)\ddot{q}^\delta + v(q^\delta, \dot{q}^\delta)\dot{q}^\delta + D(q^\delta, \dot{q}^\delta)\dot{q}^\delta + g(q^\delta) = Q^\delta \quad (3.67)$$

La matriz de masa (M) recoge todos los términos relacionados con las fuerzas inerciales que actúan sobre el sistema. Sus valores vendrán dados para el punto de equilibrio calculado en 3.64. De forma similar al modelo no lineal, sólo los términos asociados a los ángulos de la carga (ϕ_L y θ_L) se verán afectados al expresar

las matrices en coordenadas carga en lugar de coordenadas cuerpo.

$$M(q^\delta) = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_B^{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_B^{zz} + (I_B^{yy} - I_B^{zz})c^2\bar{\phi}_B & \dots \\ 0 & 0 & 0 & -I_B^{xx}s\bar{\theta}_B & (I_B^{yy} - I_B^{zz})c\bar{\phi}_Bc\bar{\theta}_Bs\bar{\phi}_B & \\ -lm_Bs\bar{\phi}_Ls\bar{\theta}_L & -lm_Bc\bar{\phi}_L & -lm_Bs\bar{\phi}_Lc\bar{\theta}_L & 0 & 0 & 0 \\ lm_Bc\bar{\phi}_Lc\bar{\theta}_L & 0 & -lm_Bc\bar{\phi}_Ls\bar{\theta}_L & 0 & 0 & 0 \end{bmatrix} \quad (3.68)$$

$$\begin{bmatrix} 0 & -lm_Bs\bar{\phi}_Ls\bar{\theta}_L & lm_Bc\bar{\phi}_Lc\bar{\theta}_L \\ 0 & -lm_Bc\bar{\phi}_L & 0 \\ 0 & -lm_Bs\bar{\phi}_Lc\bar{\theta}_L & -lm_Bc\bar{\phi}_Ls\bar{\theta}_L \\ -I_B^{xx}s\bar{\theta}_B & 0 & 0 \\ (I_B^{yy} - I_B^{zz})c\bar{\phi}_Bc\bar{\theta}_Bs\bar{\phi}_B & 0 & 0 \\ (I_B^{xx} + (I_B^{yy} - I_B^{xx})c^2\bar{\theta}_B + (I_B^{zz} - I_B^{yy})c^2\bar{\phi}_Bc^2\bar{\theta}_B & 0 & 0 \\ 0 & l^2m_B & 0 \\ 0 & 0 & l^2m_Bc^2\bar{\phi}_L \end{bmatrix}$$

Por su parte, el vector gravedad (g) contiene los valores dependientes de las fuerzas de atracción gravitatoria. Al igual que en la matriz de masa, sólo los términos asociados a los ángulos de la carga varían al pasar de coordenadas cuerpo a coordenadas carga. Sus componentes estarán formadas por el producto escalar de un valor constante, determinado por el punto de operación, y una variación angular.

$$g(q^\delta) = \begin{bmatrix} (gm \tan \bar{\phi}_B \tan \bar{\theta}_B) \phi_B^\delta - (gm) \theta_B^\delta - (gm \frac{\tan \bar{\phi}_B}{c \bar{\theta}_B}) \psi_B^\delta \\ (\frac{gm}{c \bar{\theta}_B}) \phi_B^\delta - (gm \tan \bar{\theta}_B) \psi_B^\delta \\ (gm \tan \bar{\phi}_B) \phi_B^\delta + (gm \tan \bar{\theta}_B) \theta_B^\delta \\ 0 \\ 0 \\ 0 \\ (I) \\ (II) \end{bmatrix} \quad (3.69)$$

Donde (I) y (II) representan los términos correspondientes al ángulo de alabeo y cabeceo de la carga respectivamente.

$$\begin{aligned} (I) : & -gml(\frac{c\bar{\phi}_L}{c\bar{\theta}_B} + \tan \bar{\phi}_B c \bar{\theta}_L s \bar{\phi}_L + \tan \bar{\phi}_B \tan \bar{\theta}_B s \bar{\phi}_L s \bar{\theta}_L) \phi_B^\delta + glm(s \bar{\phi}_L s \bar{\theta}_L - \tan \bar{\theta}_B c \bar{\theta}_L s \bar{\phi}_L) \theta_B^\delta \\ & + gml(\tan \bar{\theta}_B c \bar{\phi}_L + \frac{\tan \bar{\phi}_B}{c \bar{\theta}_B} s \bar{\phi}_L s \bar{\theta}_L) \psi_B^\delta + (glm_L c \bar{\theta}_L c \bar{\phi}_L + glm(\tan \bar{\phi}_B \frac{s \bar{\phi}_L}{c \bar{\theta}_B} + \tan \bar{\theta}_B c \bar{\phi}_L s \bar{\theta}_L)) \phi_L^\delta \\ & + (-glm_L s \bar{\theta}_L s \bar{\phi}_L + gml \tan \bar{\theta}_B s \bar{\phi}_L c \bar{\theta}_L) \theta_L^\delta \end{aligned} \quad (3.70)$$

$$\begin{aligned}
(II) : gml(\tan \bar{\phi}_B \tan \bar{\theta}_B c \bar{\phi}_L c \bar{\theta}_L - \tan \bar{\phi}_B c \bar{\phi}_L s \bar{\theta}_L) \phi_B^\delta - gml(\tan \bar{\theta}_B c \bar{\phi}_L s \bar{\theta}_L + c \bar{\phi}_L c \bar{\theta}_L) \theta_B^\delta \\
- gml(\tan \bar{\phi}_B \frac{c \bar{\phi}_L c \bar{\theta}_L}{c \bar{\theta}_B}) \psi_B^\delta + (-glm_L s \bar{\phi}_L s \bar{\theta}_L + glm \tan \bar{\theta}_B s \bar{\phi}_L c \bar{\theta}_L) \phi_L^\delta + (gm_L l c \bar{\phi}_L c \bar{\theta}_L \\
+ gml \tan \bar{\theta}_B c \bar{\phi}_L s \bar{\theta}_L) \theta_L^\delta
\end{aligned} \quad (3.71)$$

El vector de fuerzas generalizadas también experimenta cambios respecto a coordenadas cuerpo en sus términos asociados a los ángulos de la carga. Estos tendrán coeficientes dependientes no sólo de la resistencia aerodinámica sino también de la sustentación total del quadrotor.

$$Q = \begin{bmatrix} c \bar{\phi}_B s \bar{\theta}_B {}^{(B)} f^T \\ -s \bar{\phi}_B {}^{(B)} f^T \\ c \bar{\phi}_B c \bar{\theta}_B {}^{(B)} f^T \\ {}^{(B)} \tau_{B_x}^T \\ c \bar{\phi}_B {}^{(B)} \tau_{B_y}^T - s \bar{\phi}_B {}^{(B)} \tau_{B_z}^T \\ -s \bar{\theta}_B {}^{(B)} \tau_{B_x}^T + c \bar{\theta}_B s \bar{\phi}_B {}^{(B)} \tau_{B_y}^T + c \bar{\phi}_B c \bar{\theta}_B {}^{(B)} \tau_{B_z}^T \\ C_{B,xy} l (c \bar{\phi}_L s \bar{\theta}_L |\dot{x}_L| \dot{x}_L - s \bar{\phi}_L |\dot{y}_L| \dot{y}_L) \phi_L^\delta + (C_{B,xy} l s \bar{\phi}_L c \bar{\theta}_L |\dot{x}_L| \dot{x}_L) \theta_L^\delta - l (-c \bar{\phi}_L s \bar{\phi}_B \\ C_{B,xy} l s \bar{\phi}_L c \bar{\theta}_L |\dot{x}_L| \dot{x}_L \phi_L^\delta + C_{B,xy} l c \bar{\phi}_L s \bar{\theta}_L |\dot{x}_L| \dot{x}_L \theta_L^\delta - l (c \bar{\phi}_B c \bar{\phi}_L c \bar{\theta}_B s \bar{\theta}_L \\ + c \bar{\phi}_B c \bar{\theta}_L c \bar{\theta}_B s \bar{\phi}_L + c \bar{\phi}_B s \bar{\phi}_L s \bar{\theta}_L s \bar{\theta}_B) {}^{(B)} f^T \\ - c \bar{\phi}_B c \bar{\phi}_L c \bar{\theta}_L s \bar{\theta}_B) {}^{(B)} f^T \end{bmatrix} \quad (3.72)$$

Finalmente, se tiene una matriz de fuerzas centrífugas nula y una matriz de resistencia que tendrá términos distintos de cero tanto en la ecuaciones que rigen el movimiento horizontal de la carga como en las que describen el comportamiento angular de la misma.

$$v(q^\delta, \dot{q}^\delta) = \begin{bmatrix} 0 \end{bmatrix}_{8 \times 8} \quad (3.73)$$

$$\begin{aligned}
D(q^\delta, \dot{q}^\delta) = & \begin{bmatrix} 2(C_{B_{xy}} + C_L)|\dot{\bar{x}}_L| & 0 & 0 & 0 & 0 & 0 \\ 0 & 2(C_{B_{xy}} + C_L)|\dot{\bar{y}}_L| & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2C_{B_{xy}}l s \bar{\phi}_L s \bar{\theta}_L |\dot{\bar{x}}_L| & -2C_{B_{xy}}l c \bar{\phi}_L |\dot{\bar{y}}_L| & 0 & 0 & 0 & 0 \\ 2C_{B_{xy}}l c \bar{\phi}_L c \bar{\theta}_L |\dot{\bar{x}}_L| & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \dots \\
& \begin{bmatrix} -2C_{B_{xy}}l s \bar{\phi}_L s \bar{\theta}_L |\dot{\bar{x}}_L| & 2C_{B_{xy}}l c \bar{\phi}_L c \bar{\theta}_L |\dot{\bar{x}}_L| \\ -2C_{B_{xy}}l c \bar{\phi}_L |\dot{\bar{y}}_L| & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 2(C_{B_{xy}}l^2 s^2 \bar{\phi}_L s^2 \bar{\theta}_L |\dot{\bar{x}}_L| + C_{B_{xy}}l^2 c^2 \bar{\phi}_L |\dot{\bar{y}}_L| & -2C_{B_{xy}}l^2 c \bar{\phi}_L c \bar{\theta}_L s \bar{\phi}_L s \bar{\theta}_L |\dot{\bar{x}}_L| \\ -2C_{B_{xy}}l^2 c \bar{\phi}_L c \bar{\theta}_L s \bar{\phi}_L s \bar{\theta}_L |\dot{\bar{x}}_L| & 2C_{B_{xy}}l^2 c^2 \bar{\phi}_L c^2 \bar{\theta}_L |\dot{\bar{x}}_L| \end{bmatrix}
\end{aligned} \tag{3.74}$$

4 Modo Stabilize

En este capítulo y en los sucesivos se desarrollarán los esquemas control propuestos por ArduCopter para estabilizar los ángulos, la altura y la posición del quadrotor. Estos se corresponden con la versión *Copter 3.3.1 26-Oct-2015* para Gazebo y se puede acceder a ellos a través del enlace [17]. El objetivo es estudiar cómo se realiza el control de los ángulos mediante una serie de controladores PID en cascada y reproducirlos en MATLAB® para poder aplicarlos posteriormente al modelo de quadrotor desarrollado en el Capítulo 2. Se busca así establecer una equivalencia entre el modelo en Gazebo de Erle Copter y el modelo desarrollado en este trabajo. Para ello, se realizarán distintas pruebas de cara a las diferentes direcciones de actitud y se establecerá una comparación gráfica entre ambos modelos.

En el modo Stabilize, el piloto controla directamente la actitud del vehículo mediante los mandos de vuelo y lo mismo deberá hacer con la altitud, ver [4]. Esto exige cambiar continuamente la palanca para mantener la posición y una atención constante sobre el direccionamiento para evitar que pueda caer y estrellarse. Constituyen el control primario del vehículo y están incluidos en otros modos de vuelo como son Alt-Hold y Loiter. De hecho, en las pruebas realizadas en Gazebo, se tomó el modo Alt-Hold para que se mantuviera la altura, cambiando los ángulos del vehículo a través del modo Stabilize.

El esquema para poder controlar la actitud del vehículo en el modo Stabilize constará de varias partes o lazos de control. El lazo interno controlará la velocidad angular del quadrotor y generará una señal de actuación sobre el par correspondiente. A su vez, el lazo externo calculará a partir del error en ángulo la velocidad angular de referencia que alimentará al lazo interno. Tanto el control de la actitud como el de la velocidad angular se realizará en ejes cuerpo, puesto que las ecuaciones de la dinámica de la actitud estarán expresadas en dichos ejes. Estas pueden consultarse en la Sección 2.6.

Antes de comenzar con el control de los ángulos roll y pitch, la referencia será transformada para obtener los ángulos de inclinación deseados en centígrados. Para ello, tras verificar la validez del máximo ángulo permitido, se escalan las entradas en alabeo y cabeceo al rango de valores establecido por ArduCopter. Estas operaciones se llevarán a cabo en el bloque conocido como *get_pilot_desired_lean_angles*.

$$\begin{aligned} \text{scaler} &= \text{ANGLE_MAX} / 3000 \\ \phi_{Bin} &= \phi_{Bin} * \text{scaler} \\ \theta_{Bin} &= \theta_{Bin} * \text{scaler} \end{aligned} \tag{4.1}$$

Además de la restricción lineal, se aplica un límite circular a las entradas en alabeo y cabeceo, de forma que la raíz cuadrada de la suma de ambas entradas al cuadrado no supere el máximo ángulo fijado. En caso de ser mayor que el valor fijado, se disminuyen ambas entradas mediante un coeficiente de reducción. Una

vez escaladas dichas entradas, se introducen en el sistema como ángulos de inclinación deseados.

$$total_{in} = \sqrt{\phi_{Bin}^2 + \theta_{Bin}^2}$$

$$if \quad total_{in} > ANGLE_MAX$$

$$ratio = ANGLE_MAX / total_{in}$$

$$\phi_{Bin} = \phi_{Bin} * ratio$$

$$\theta_{Bin} = \theta_{Bin} * ratio$$
(4.2)

4.1 Control del Roll

4.1.1 Esquema de Control para Roll

Los controladores para la dirección roll constarán de: un bloque a la entrada que filtra el ángulo deseado para que el error en alabeo no sea muy elevado, de un controlador proporcional que genera una respuesta cuadrática cuando el error sobrepasa un determinado valor de diseño y finalmente de un controlador PID para el control de la velocidad angular. Su esquema en SIMULINK® puede ser consultado en la Figura 4.1.

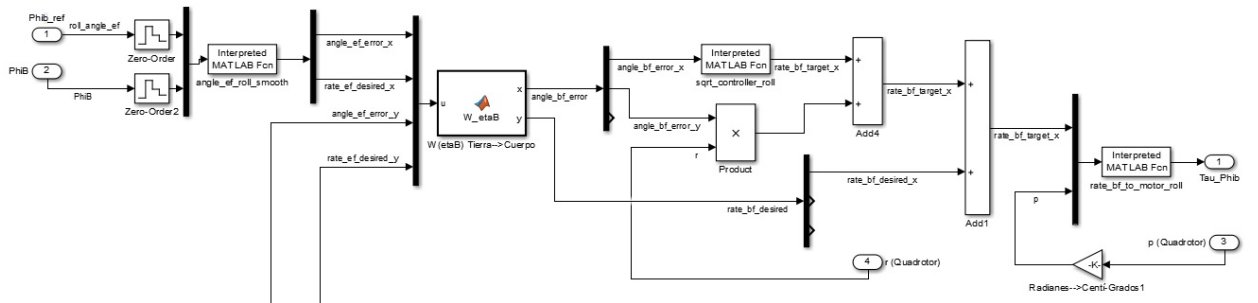


Figura 4.1 Esquema para el control del Roll en SIMULINK®.

La entrada al sistema viene dada por el ángulo de alabeo deseado que será distinto del ángulo objetivo calculado cuando la velocidad prealimentada esté activa. Se busca tener así una trayectoria de subida o bajada hacia la referencia más suave. En general, para pasar de una señal continua a otra discreta compatible con el controlador, se usará un mantenedor de orden cero.

$$roll_angle_ef_k = \phi_{Bin}$$
(4.3)

Si la velocidad de prealimentación está activa, entonces se ejecuta el bloque *angle_ef_roll_smooth* para calcular una velocidad prealimentada y un ángulo objetivo. La variable interna, *rate_bf_ff_enabled*, determina cuando se activa o no la prealimentación en velocidad y estará activa por defecto en el modo Stabilize. El valor de la aceleración máxima es un parámetro establecido como modificable en el APMPlanner y su valor se puede consultar en la Tabla 4.1. Lo mismo ocurre para la ganancia de suavizado, *smoothing_gain*,

cuyo valor será distinto al de la constante STB_RLL_P empleada en el control de actitud.

$$\begin{aligned} accel_roll_max &= ATC_ACCEL_R_MAX \\ \text{if } accel_roll_max \geq 0 \ \& \ rate_bf_ff_enabled == 1 \\ smoothing_gain &= 2 + RC_FEEL_RP/10 = 4.5 \end{aligned} \quad (4.4)$$

La velocidad angular límite se obtiene a partir de la aceleración máxima, tras multiplicar dicha aceleración por el tiempo de muestreo ($T = 0.01$). Esta velocidad será empleada para limitar la velocidad prealimentada calculada.

$$rate_change_limit = accel_roll_max * T \quad (4.5)$$

La velocidad prealimentada se obtiene a partir del error entre el ángulo deseado y el ángulo objetivo mediante el controlador $sqrt_controller$. El código del mismo se detalla en 4.1 y es idéntico al empleado en el lazo de control externo para la actitud, cuya tendencia puede verse en la Figura 4.2. Obviamente, su comportamiento será distinto, no solo cambia el error, sino también la ganancia y aceleración de entrada al mismo. El ángulo objetivo, $angle_ef_target.x_k$, vendrá definido más abajo y su expresión se puede consultar en 4.10.

$$\begin{aligned} error_k &= roll_angle_ef_k - angle_ef_target.x_{k-1} \\ rate_ef_desired_k &= sqrt_controller(error_k, smoothing_gain, accel_roll_max) \end{aligned} \quad (4.6)$$

Una vez obtenida la velocidad prealimentada, se limita dicha velocidad usando para ello la velocidad angular límite calculada anteriormente. Otra forma de verlo, sería verificar que la nueva velocidad calculada no genere una aceleración mayor que la permitida.

$$\begin{aligned} rate_ef_desired.x_{k-1} - rate_change_limit &\leq rate_ef_desired_k \dots \\ \dots &\leq rate_ef_desired.x_{k-1} + rate_change_limit \end{aligned} \quad (4.7)$$

$$rate_ef_desired.x_k = rate_ef_desired_k \quad (4.8)$$

Esta velocidad prealimentada se utiliza, a su vez, para actualizar el ángulo objetivo que será distinto del deseado y para alimentar al controlador PID de velocidad angular. El error en alabeo se obtiene como la diferencia entre este ángulo objetivo y el ángulo de alabeo del drone real y pasará posteriormente al controlador $sqrt_controller$ de actitud. Para que este error tenga un valor razonable, se realiza la equivalencia a un ángulo comprendido entre $\pm 180^\circ$ y se trunca su resultado teniendo en cuenta la sobreoscilación máxima permitida por ArduCopter. Este valor solo será alcanzado en casos extremos porque existen otras limitaciones que actúan con anterioridad. La diferencia angular vendrá dada en centígrados, ya que todos los controladores trabajan en dichas unidades. El parámetro para pasar de radianes a centígrados vendrá dado por: $C = \frac{180}{\pi} 100 = 5729.57795$.

$$\begin{aligned} angle_ef_error.x_k &= angle_ef_target.x_{k-1} - C * \phi_{Bk} \\ -30000 &\leq angle_ef_error.x_k \leq 30000 \end{aligned} \quad (4.9)$$

Donde ϕ_{Bk} es el ángulo de alabeo actual del vehículo expresado en ejes tierra y discretizado en el tiempo

para que el controlador pueda operar con él. A continuación, se actualiza el ángulo roll objetivo teniendo en cuenta el máximo ángulo permitido.

$$\begin{aligned} angle_ef_target.x_k &= C * \phi_{Bk} + angle_ef_error.x_k \\ angle_ef_target.x_k &= angle_ef_target.x_k + rate_ef_desired.x_k * T \end{aligned} \quad (4.10)$$

En el caso de no estar activado el cálculo de la velocidad prealimentada, el ángulo objetivo sería igual al ángulo deseado a la entrada del sistema. El error en alabeo sería la diferencia entre este ángulo deseado y el ángulo real del dron y la velocidad prealimentada tendría un valor nulo.

$$\begin{aligned} angle_ef_target.x_k &= roll_angle_ef_k \\ angle_ef_error.x_k &= angle_ef_target.x_k - C * \phi_{Bk} \\ rate_ef_desired.x_k &= 0 \end{aligned} \quad (4.11)$$

Una vez calculado el error en alabeo, ya sea por uno u otro método, se restringe dicho valor mediante el ángulo máximo permitido (ANGLE_MAX). Su valor puede consultarse en la Tabla 4.2.

$$\begin{aligned} angle_max &= ANGLE_MAX \\ -angle_max &\leq angle_ef_error.x_k \leq angle_max \end{aligned} \quad (4.12)$$

Obtenidos el error en el ángulo de alabeo ($angle_ef_error.x_k$) y la velocidad prealimentada ($rate_ef_desired.x_k$) en ejes tierra, se procede a realizar la transformación matricial $W(\eta_B)$. Esta matriz cambia las variaciones en los ángulos de Euler, expresadas en el sistema inercial, a un vector de velocidad angular expresado en los ejes del propio cuerpo. Tal cambio es necesario dado que el control de la actitud está referido a las coordenadas del cuerpo y, por tanto, las velocidades angulares del vehículo también. Cabe destacar que el carácter “ef” hacer referencia a ejes tierra, mientras que el carácter “bf” a ejes del propio cuerpo.

$$angle_bf_error = W(\eta_B) * angle_ef_error; \quad \text{con} \quad angle_ef_error = \begin{bmatrix} angle_ef_error.x \\ angle_ef_error.y \\ angle_ef_error.z \end{bmatrix}. \quad (4.13)$$

$$rate_bf_desired = W(\eta_B) * rate_ef_desired; \quad \text{con} \quad rate_ef_desired = \begin{bmatrix} rate_ef_desired.x \\ rate_ef_desired.y \\ rate_ef_desired.z \end{bmatrix}. \quad (4.14)$$

Una vez convertido el error en ángulo a ejes cuerpo ($angle_bf_error.x_k$), se procede a calcular la velocidad objetivo mediante el controlador $sqrt_controller$, contenido en el bloque $sqrt_controller_roll$. Este controlador tiene la característica de calcular una respuesta lineal cuando se está cerca del objetivo o la raíz cuadrada de la misma cuando se encuentra lejos. Los parámetros que condicionan su respuesta están recogidos en la Tabla 4.1 y son la ganancia proporcional propia de un controlador P (STB_RLL_P) y la aceleración angular máxima permitida en alabeo ($ATC_ACCEL_R_MAX$).

En primer lugar, se calcula la aceleración máxima permitida en roll que, junto con la ganancia proporcional, permiten el cálculo de la distancia lineal que determinará cuando nos encontramos en una situación o en otra. Además, se comprueba la validez de dicha aceleración, asegurando que se encuentre entre los parámetros

Tabla 4.1 Ganancias características del controlador en Roll y valores máximos.

Parámetro	Valor	Unidad
STB_RLL_P	11.25	-
ANGLE_MAX	3000	Centígrados
ATC_ACCEL_R_MAX	110000	(Centígrados/s ²)
RC_FEEL_RP	25	-

internos proporcionados por ArduPilot o asignándole estos en caso de superar su valor.

$$\begin{aligned}
 accel_max &= (ATC_ACCEL_R_MAX)/2 \\
 kp &= STB_RLL_P \\
 Asegura \quad que \quad 36000 &\leq accel_max \leq 72000 \\
 rate_bf_target.x_k &= sqrt_controller(angle_bf_error.x_k, kp, accel_max)
 \end{aligned} \tag{4.15}$$

Donde $accel_max$ está expresada en centígrados en concordancia con el error en ángulo. Tomando como variables de entrada el error en ángulo de alabeo, la ganancia proporcional (STB_RLL_P) y la aceleración máxima previamente calculada, el controlador $sqrt_controller$ calcula la respuesta tal y como se aprecia en el Listing 4.1.

Código 4.1 Código del controlador $sqrt_controller$ para roll y pitch.

```

function rate_target = sqrt_controller(error, P, accel_max)

    % Calcula la distancia lineal por debajo de la cual devuelve la
    % respuesta lineal.
    LD = accel_max/(P^2);

    % Si se esta lejos, devuelve la raíz cuadrada de la respuesta.
    if error > LD
        rate_target = sqrt(2*accel_max*(error-(LD/2)));
        % Idéntico para el caso negativo.
    elseif error < -LD
        rate_target = -sqrt(2*accel_max*(-error-(LD/2)));
        % Si se esta cerca, devuelve la respuesta lineal.
    else
        rate_target = error*P;
    end

```

Con $P = STB_RLL_P$ la ganancia proporcional de alabeo, $accel_max$ la aceleración máxima, $rate_target$ la velocidad objetivo o repuesta del controlador y LD la distancia lineal. Otra forma de ver la distancia lineal sería multiplicar su valor por la ganancia proporcional al cuadrado e igualar su resultado a la aceleración máxima prefijada, es decir, $accel_max = P * vel_max = P^2 * LD$. El cálculo de la raíz cuadrada de la respuesta no sería mas que la relación física $v^2 = 2 * a * d$ adaptada a la actitud.

Para ver el comportamiento de este controlador se realizó una prueba en bucle abierto, consistente en introducir una serie de errores en ángulo que iban desde -30° a 30° , utilizando para ello los parámetros recogidos en Tabla 4.1. La respuesta del mismo se recoge en la Figura 4.2 y se observa claramente como la respuesta es lineal para errores por debajo de $\pm 5^\circ$ y cuadrática para valores superiores. Señalar también el comportamiento simétrico del controlador, como cabría esperar, y los valores máximos de velocidad angular alcanzables $\pm 175^\circ/s$

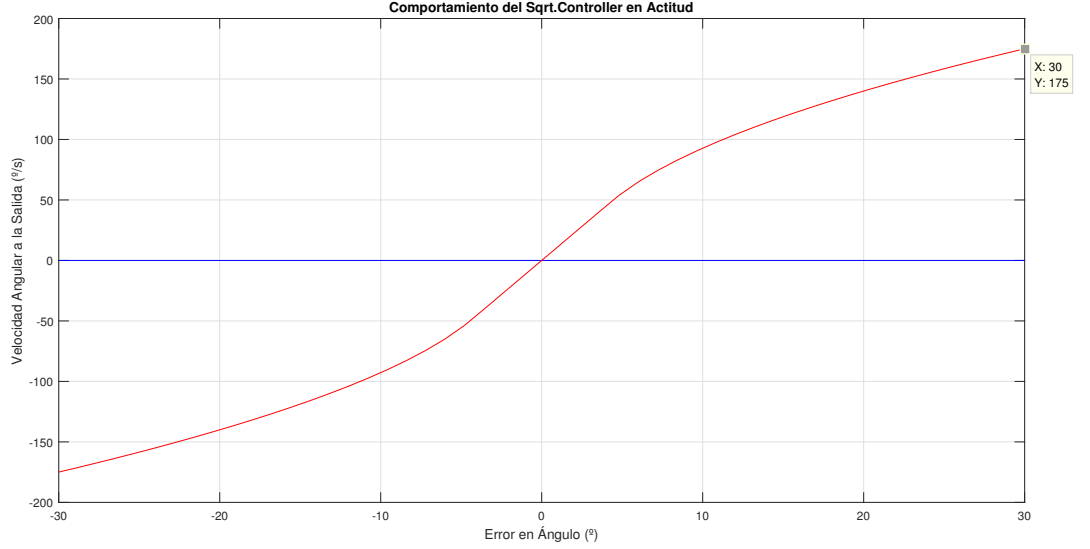


Figura 4.2 Respuesta del *Sqrt_Controller* ante entrada rampa del error en alabeo.

Una vez ejecutado el *sqrt_controller* en roll, a la velocidad de salida del mismo se le suma la componente de velocidad angular de alabeo que tiene en cuenta al efecto de la precesión del eje de rotación perpendicular al cuerpo.

$$rate_bf_target.x = rate_bf_target.x + angle_bf_error.y * r \quad (4.16)$$

La componente resultante de la suma anterior es la velocidad objetivo en roll (*rate_bf_target.x*), a la que se le añade ahora la velocidad de prealimetación en esa dirección. Esta velocidad angular ha sido previamente calculada mediante el controlador *sqrt_controller* usando para ello una ganancia proporcional de suavizado distinta a la utilizada en la obtención de la velocidad angular objetivo.

$$rate_bf_target.x = rate_bf_target.x + rate_bf_desired.x \quad (4.17)$$

Una vez obtenida la velocidad objetivo de alabeo (*rate_bf_target.x*), expresada en ejes cuerpo, se calcula el error en velocidad angular que alimenta al controlador PID del lazo interno. Este controlador se encuentra en el bloque *rate_bf_to_motor_roll* y genera a partir del error la señal de actuación necesaria o momento de alabeo que será introducido en las ecuaciones dinámicas desarrolladas en el Capítulo 2. Como todo controlador PID, consta de una parte proporcional, una parte integral y otra derivativa.

$$rate_error_roll = rate_bf_target.x - p \quad (4.18)$$

La parte proporcional es el resultado de multiplicar el error en velocidad por una ganancia proporcional cuyo valor se recoge en la Tabla 4.2. Para simplificar las expresiones 4.19 - 4.21, se sustituirá en dichas ecuaciones $rate_error_roll_k = e_k$.

$$p_k = Kp * e_k \quad (4.19)$$

La parte integral se calcula integrando dicho error de forma discreta y multiplicándolo por una ganancia integral (KI). Este resultado es posteriormente truncado tanto por arriba como por abajo (IMAX), estando su valor expresado en la Tabla 4.2. Su función es evitar un valor excesivo a la salida del controlador y es idéntico, tanto para velocidades angulares positivas como negativas, cumpliéndose así la simetría de giro en

ambas direcciones.

$$i_k = i_{k-1} + K_i * e_k * T \quad (4.20)$$

Siendo $T = 0.01$ el tiempo de muestreo del controlador discreto.

La parte derivativa incluye un filtro de paso bajo a la entrada, tras el cual se procede a derivar la entrada al controlador y multiplicarla posteriormente por la ganancia derivativa correspondiente (K_d). El uso del filtro tiene como objetivo suavizar el efecto de la señal error a la entrada del controlador.

$$\begin{aligned} der_k &= (e_k - e_{k-1})/T \\ filt &= \frac{T}{T + (2 * \pi * FILT)} \\ derf_k &= derf_{k-1} + filt * [der_k - derf_{k-1}] \\ d_k &= K_d * derf_k \end{aligned} \quad (4.21)$$

Con der_k haciendo referencia a la derivada del error y $derf_k$ calculada como la derivada del error una vez filtrada.

Tabla 4.2 Ganancias características del controlador RATE_RLL.

Parámetro	Valor	Observación
RATE_RLL_P	0.2025	Kp
RATE_RLL_I	0.2025	Ki
RATE_RLL_D	0.0125	Kd
RATE_RLL_IMAX	1000	IMAX
RATE_RLL_FILT_HZ (Hz)	20	FILT

Finalmente, se suman las distintas partes y se obtiene la salida en roll correspondiente.

$$salida_roll = p_k + i_k + d_k \quad (4.22)$$

4.1.2 Análisis de la Respuesta en Roll

Para estudiar el comportamiento del controlador PID en velocidad angular, se realizó una prueba consistente en introducir una entrada escalón de $1^\circ/s$ en la dirección de alabeo, dejando el resto de entradas al sistema nulas. La prueba realizada en bucle cerrado se ha representado en la Figura 4.3 y muestra la rapidez de la velocidad de alabeo para alcanzar la velocidad angular deseada, aunque oscilando ligeramente en torno a ella. Tanto el tiempo de subida como la sobreoscilación alcanzan valores pequeños, entorno a $0.4s$ y un 10% respectivamente. También se ha representado el cambio en el ángulo de alabeo, en consonancia con la velocidad angular en esa dirección, y la caída considerable de la altitud debida a la variación creciente del ángulo de alabeo. Estos resultados demuestran que no sería posible controlar la velocidad angular del quadrotor únicamente con un controlador PID, pues sería necesario disponer también de un controlador para la actitud del mismo y así evitar una caída brusca en la altitud.

No se ha representado el ángulo de cabeceo ni su velocidad angular correspondiente debido a que la dinámica en ambas direcciones está desacoplada, es decir, la introducción de una velocidad angular escalón en roll no tiene efecto alguno en dirección pitch cuando el ángulo de guiñada es nulo. Sin embargo, cuando dicho ángulo sea distinto de cero, si se tendrá una perturbación despreciable en el ángulo de cabeceo pero importante en la posición horizontal del vehículo. Esto es consecuencia de que el ángulo de guiñada afecta a la dinámica de la posición horizontal del vehículo.

Finalmente, se presentan las pruebas realizadas en SIMULINK[®] para ver el comportamiento del ángulo de

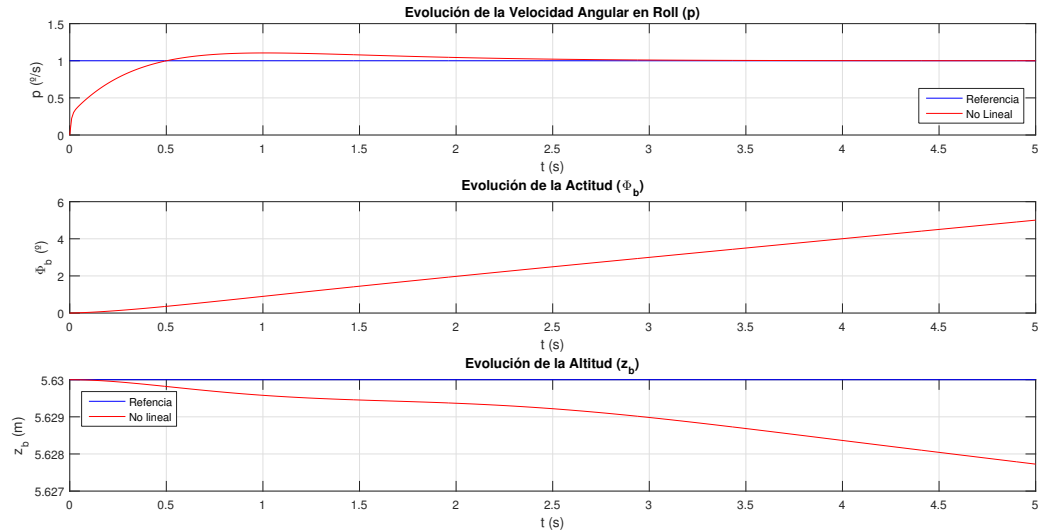


Figura 4.3 Respuesta de p ante entrada escalón de $1^\circ/s$.

alabeo del quadrotor cuando se inclina un cierto ángulo y el efecto que tiene dicha variación sobre la actitud y altitud del vehículo. Sus resultados se comparan con los obtenidos en Gazebo para distintos valores de la emisora que controla el roll. Así pues, se introdujeron los PWM establecidos en la Tabla 4.3 y se obtuvieron los correspondientes ángulos de referencia.

Tabla 4.3 Valor PWM en Gazebo y Ángulo ϕ_B Deseado.

Canal (rc)	PWM	$\phi_{Bin} (^\circ)$	Tiempo (s)
1	1550	1.05	0
1	1600	4.65	17.7
1	1650	8.75	26.7
1	1500	0	33.2

Para la prueba en MATLAB[®] se emplearon los ángulos presentados en la tabla anterior, activándose dichas señales cuando correspondía. Dado que se ha intentado reproducir los esquemas de control desarrollados por ArduPilot en el modelo en MATLAB[®], cabría esperar que el ángulo de alabeo tenga un comportamiento similar en ambos modelos. La respuesta del ángulo roll se presenta en la Figura 4.4 y se puede ver como efectivamente alcanza el ángulo deseado en muy poco tiempo, aunque el modelo en Gazebo lo hará más rápidamente y, en consecuencia, oscilando más en torno al ángulo deseado. Igualmente, la velocidad angular en ejes cuerpo oscilará más en el modelo en Gazebo y experimentará valores más grandes como consecuencia de haber considerado otros efectos dinámicos que no se tienen en cuenta en el modelo de quadrotor desarrollado aquí. Estas diferencias que tienen lugar durante el transitorio, desaparecen una vez que se ha alcanzado el ángulo deseado y la velocidad angular es nula. Los tiempos de subida, establecimiento de la señal y sobreoscilación de la misma, pueden consultarse en la Tabla 4.4.

Como consecuencia de cambiar el ángulo de alabeo, se produce una ligera oscilación en la altitud del vehículo que viene a ser del mismo orden en ambos modelos. Para evitar esta caída, se activan los controladores del modo Alt-Hold cuando se encuentren implementados o, en caso contrario, el usuario tendrá que manejar el mando del throttle para evitarlo. En MATLAB[®], el control de la altitud se realizará solamente mediante los controladores de altitud anidados.

Aunque no se han representado aquí, tanto el ángulo de cabeceo como el de guiñada no se verán afectados por cambios en el ángulo de alabeo. Esto no es totalmente cierto en Gazebo, pues dichos ángulos sufrirán

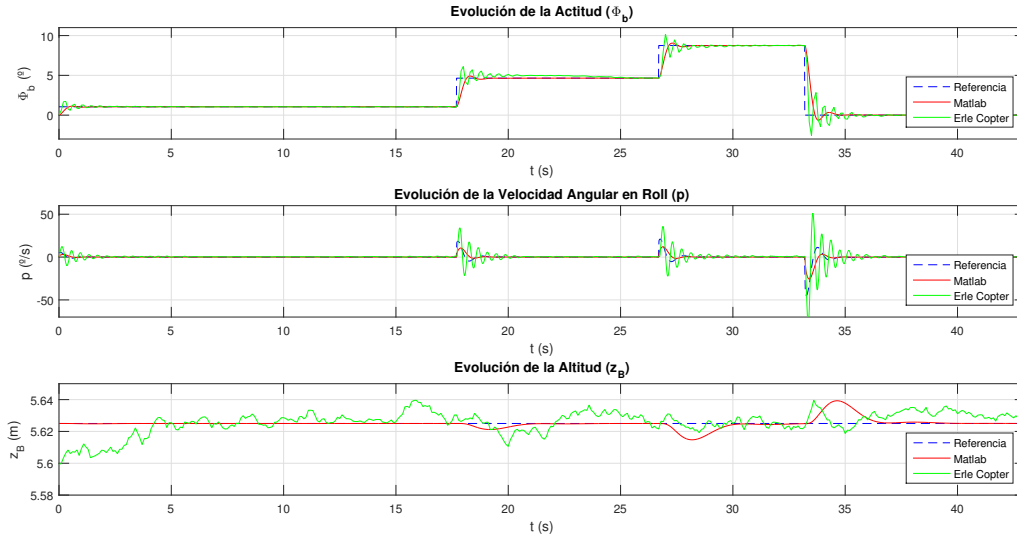


Figura 4.4 Respuesta de ϕ_B ante varias entradas escalón en ángulo.

pequeñas perturbaciones que se deberán no solo a la dinámica del vehículo, sino también a los ruidos procedentes de los aparatos de medida como las IMU's o barómetros.

Tabla 4.4 Análisis de la respuesta temporal de ϕ_B .

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	0.4	1.3	7
Erle Copter	0.2	16	60

Los errores en ángulo, velocidad angular y altitud se dibujan en la Figura 4.5 y confirman las diferencias existentes entre ambos modelos durante el transitorio. Así, la mayor diferencia en ángulo tiene lugar para 7° y lo mismo ocurre para la velocidad angular con errores de aproximadamente $50^\circ/s$ durante el cambio de alabeo.

Las fuerzas experimentadas por los rotores del quadrotor para cambiar el ángulo de alabeo del mismo se recogen en la Figura 4.6. En ella, puede observarse como los rotores “2” y “4” son los encargados de tal cambio y cómo sus valores se encuentran dentro de los límites razonables sin llegar a saturarse en ningún momento. También puede verse como los rotores “1” y “3” sufren pequeñas variaciones en su empuje para compensar la pérdida de altitud. En cualquier caso, para los máximos ángulos establecidos de 30° , tampoco se llegarían a saturar los rotores con la consiguiente pérdida de control sobre el mismo.

A la vista de los resultados, el modelo teórico propuesto en el Capítulo 2 sigue en líneas generales la tendencia del modelo dado por Erle Copter. Sin embargo, las oscilaciones presentes en este último modelo son bastante grandes en el caso de la velocidad angular de alabeo. Esto es debido fundamentalmente a que el modelo teórico propuesto aquí, está realizado a partir de un número considerable de simplificaciones e hipótesis que tal vez no han sido tenidas en cuenta en el modelo de Erle Copter. Los resultados expuestos hasta ahora son idénticos para ángulos negativos, cumpliéndose así la simetría en ambas direcciones del movimiento.

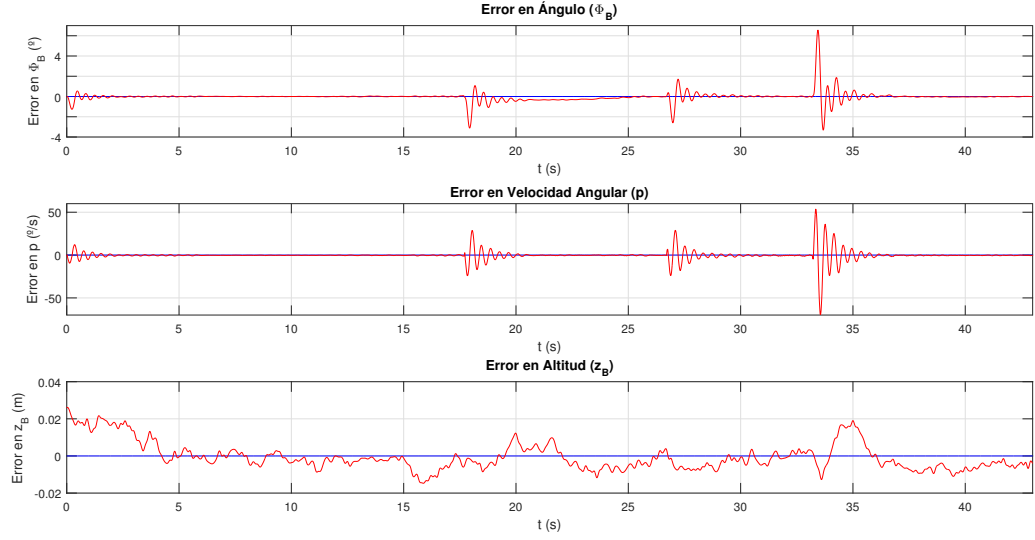


Figura 4.5 Error entre el modelo en Matlab y Gazebo para ϕ_B .

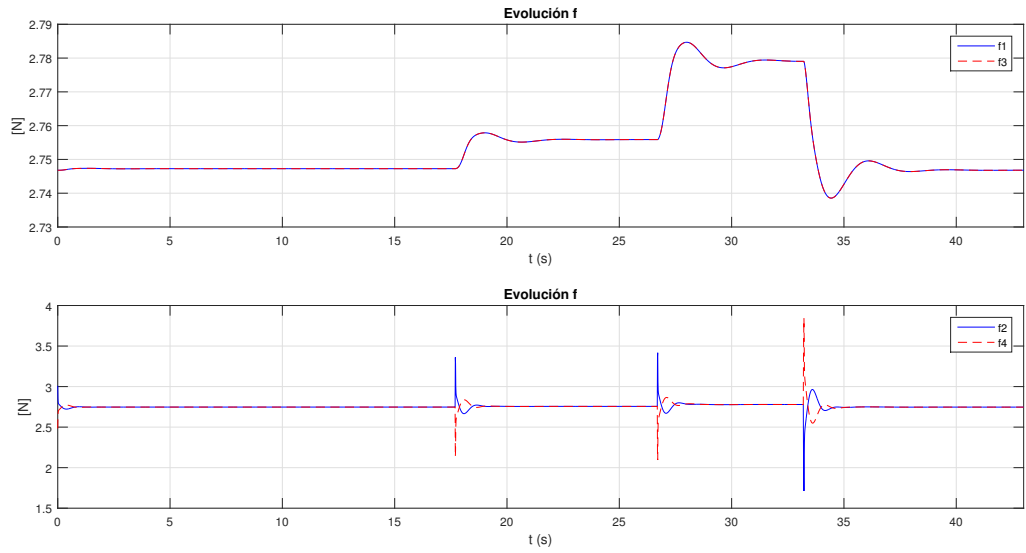


Figura 4.6 Variación de Empuje de los Rotores ante cambios en ϕ_B .

4.2 Control del Pitch

4.2.1 Esquema de Control para Pitch

El diseño de los controladores para el ángulo de cabeceo es idéntico al ya establecido para el ángulo de alabeo en la Subsección 4.1.1. De idéntica forma, estos constarán de tres partes diferenciadas: un prefiltro, un controlador *sqrt_controller* y un PID para velocidad angular. Su esquema de control en SIMULINK[®] se representa en la Figura 4.7.

La entrada al sistema es el ángulo de cabeceo deseado que cambia con respecto al ángulo objetivo calculado en la expresión 4.30. Igualmente, para pasar de señales continuas en el tiempo a señales discretas compatibles

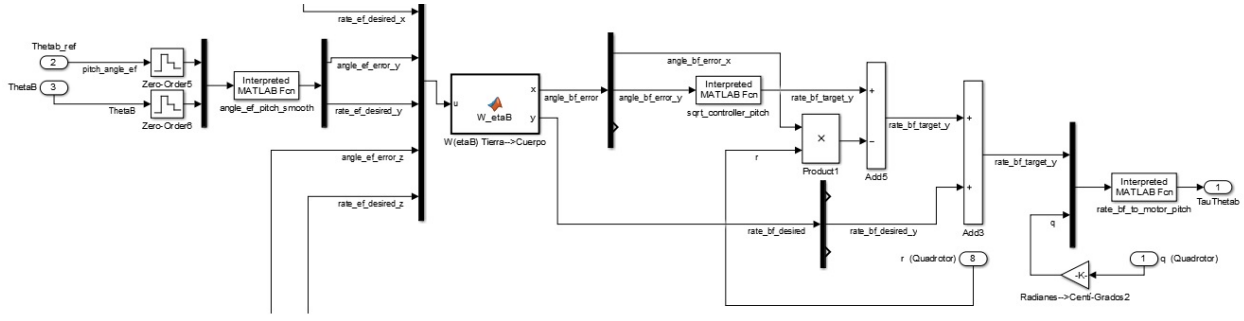


Figura 4.7 Esquema para el control del Pitch en SIMULINK®.

con los controladores, se usará un mantenedor de orden cero.

$$pitch_angle_ef_k = \theta_{Bin} \quad (4.23)$$

Cuando la velocidad de prealimentación esté activa y la aceleración en pitch sea mayor que cero, el bloque establecido como *angle_ef_pitch_smooth* será el encargado de calcular el ángulo objetivo y la velocidad de prealimentación necesaria a través del controlador *sqr_controller*. Los valores de las principales variables quedan recogidos en la Tabla 4.5 y, como puede observarse, se han tomado los mismos en ambas direcciones horizontales.

$$\begin{aligned} accel_pitch_max &= ATC_ACCEL_P_MAX \\ if \ accel_pitch_max &\geq 0 \ \& \ rate_bf_ff_enabled == 1 \\ smoothing_gain &= 2 + RC_FEEL_RP/10 = 4.5 \end{aligned} \quad (4.24)$$

Para limitar la velocidad prealimentada en cabeceo se hace uso de la velocidad angular límite.

$$rate_change_limit = accel_pitch_max * T \quad (4.25)$$

El controlador *sqr_controller* empleado para obtener la velocidad prealimentada, es idéntico al ya utilizado para el control de la actitud del vehículo y su código se encuentra recogido en 4.1. Aunque su comportamiento será distinto al controlador usado para la actitud, su tendencia será similar, constando de una respuesta lineal y otra cuadrática.

$$\begin{aligned} error_k &= pitch_angle_ef_k - angle_ef_target.y_{k-1} \\ rate_ef_desired_k &= sqr_controller(error_k, smoothing_gain, accel_pitch_max) \end{aligned} \quad (4.26)$$

Calculada la velocidad prealimentada, se establecen límites para dicha velocidad usando la velocidad máxima obtenida en 4.25.

$$\begin{aligned} rate_ef_desired.y_{k-1} - rate_change_limit &\leq rate_ef_desired_k \dots \\ \dots &\leq rate_ef_desired.y_{k-1} + rate_change_limit \end{aligned} \quad (4.27)$$

$$rate_ef_desired.y_k = rate_ef_desired_k \quad (4.28)$$

Esta velocidad deseada es utilizada para incrementar el ángulo objetivo y es enviada posteriormente al controlador PID de velocidad angular en pitch. El error en ángulo de cabeceo se obtiene como la diferencia entre este ángulo objetivo y el ángulo de cabeceo del drone real y será introducido posteriormente en el controlador *sqr_controller* de actitud. Para que esta variación angular tenga un valor razonable, se efectúa la conversión a un ángulo equivalente comprendido entre $\pm 180^\circ$ y se limita su resultado teniendo en cuenta la sobreoscilación máxima permitida.

$$\begin{aligned} angle_ef_error.y_k &= angle_ef_target.y_{k-1} - C * \theta_{Bk} \\ -30000 &\leq angle_ef_error.y_k \leq 30000 \end{aligned} \quad (4.29)$$

$$\begin{aligned} angle_ef_target.y_k &= C * \theta_{Bk} + angle_ef_error.y_k \\ angle_ef_target.y_k &= angle_ef_target.y_k + rate_ef_desired.y_k * T \end{aligned} \quad (4.30)$$

Donde θ_{Bk} es el ángulo de cabeceo actual del vehículo expresado en ejes tierra, el cual se ha discretizado para poder realizar las operaciones necesarias en tiempo discreto. La constante C pasará los ángulos y velocidades de radianes a centígrados en correspondencia con las unidades adoptadas durante todo el proceso.

Si la velocidad prealimentada está desactivada, el ángulo objetivo sería igual al ángulo deseado, el error en cabeceo sería la diferencia entre este ángulo deseado y el ángulo real del drone y la velocidad prealimentada tendría un valor nulo.

$$\begin{aligned} angle_ef_target.y_k &= pitch_angle_ef_k \\ angle_ef_error.y_k &= angle_ef_target.y_k - C * \theta_{Bk} \\ rate_ef_desired.y_k &= 0 \end{aligned} \quad (4.31)$$

Obtenido el error en cabeceo, expresado en ejes tierra, se procede a restringir su valor mediante el ángulo máximo proporcionado por ArduCopter (ANGLE_MAX), cuyo valor se encuentra en la Tabla 4.5.

$$\begin{aligned} angle_max &= ANGLE_MAX \\ -angle_max &\leq angle_ef_error.y_k \leq angle_max \end{aligned} \quad (4.32)$$

Una vez que se han calculado el error en cabeceo ($angle_ef_error.y_k$) y la velocidad prealimentada correspondiente ($rate_ef_desired.y_k$) en ejes tierra, se realiza el cambio a ejes cuerpo.

$$angle_bf_error = W(\eta_B) * angle_ef_error \quad (4.33)$$

$$rate_bf_desired = W(\eta_B) * rate_ef_desired \quad (4.34)$$

De esta forma, se obtiene el error en cabeceo ($angle_bf_error.y_k$), expresado en ejes cuerpo, que será introducido en el bloque *sqr_controller_pitch*. Este controlador será el mismo que el empleado para el cálculo de la velocidad objetivo en alabeo y sus valores característicos se recogen en la Tabla 4.5. Dado que posee la misma ganancia proporcional e idéntica aceleración máxima permitida en dirección “y”, su comportamiento será idéntico al representado en la Figura 4.2 para roll. Así pues, no se volverá a realizar una descripción detallada de dicho controlador y el lector es referido al Código 4.1.

Tabla 4.5 Ganancias características del controlador en Pitch y valores máximos.

Parámetro	Valor	Unidad
STB_PIT_P	11.25	-
ANGLE_MAX	3000	Centígrados
ATC_ACCEL_P_MAX	110000	Centígrados/s ²
RC_FEEL_RP	25	-

Las entradas al controlador *sqrt_controller* volverán a ser las mismas que en el caso anterior, salvo el error en cabeceo.

$$\begin{aligned}
 accel_max &= (ATC_ACCEL_P_MAX)/2 \\
 kp &= STB_PIT_P \\
 Asegura \quad que \quad 36000 &\leq accel_max \leq 72000 \\
 rate_bf_target.y_k &= sqrt_controller(angle_bf_error.y_k, kp, accel_max)
 \end{aligned} \tag{4.35}$$

A la velocidad objetivo en dirección “y”, se le resta ahora la velocidad angular que tiene en cuenta el efecto de la precesión en el eje de guiñada.

$$rate_bf_target.y = rate_bf_target.y - angle_bf_error.x * r \tag{4.36}$$

A este resultado, se le suma a su vez la velocidad prealimentada (*rate_bf_desired.y*), obtenida previamente y expresada en ejes cuerpo tras la transformación correspondiente.

$$rate_bf_target.y = rate_bf_target.y + rate_bf_desired.y \tag{4.37}$$

Obtenida finalmente la velocidad objetivo en pitch, se calcula el error en velocidad angular como la diferencia entre la primera y la velocidad actual del quadrotor en esa dirección. Este valor será introducido a su vez en el controlador PID del lazo interno, contenido en el bloque *rate_bf_to_motor_pitch*.

$$rate_error_pitch = rate_bf_target.y - q \tag{4.38}$$

Este controlador, que actúa sobre la velocidad angular de cabeceo del quadrotor, es también diseñado como un PID, cuyas ganancias se recogen en la Tabla 4.6. Constará de una parte proporcional, integral y derivativa. Además, la parte derivativa estará filtrada por un filtro de paso bajo para suavizar el error de entrada al controlador. Estos valores son ligeramente distintos a los ya vistos para el ángulo roll, dado que la dinámica en esta dirección es también distinta. Por último, la salida de la parte integral del controlador es truncada tanto en dirección positiva como negativa por el valor IMAX, e igualmente el par en cabeceo resultante será saturado en ambas direcciones por un valor establecido en el código interno de ArduPilot de ± 5000 .

Tabla 4.6 Ganancias características del controlador RATE_PIT.

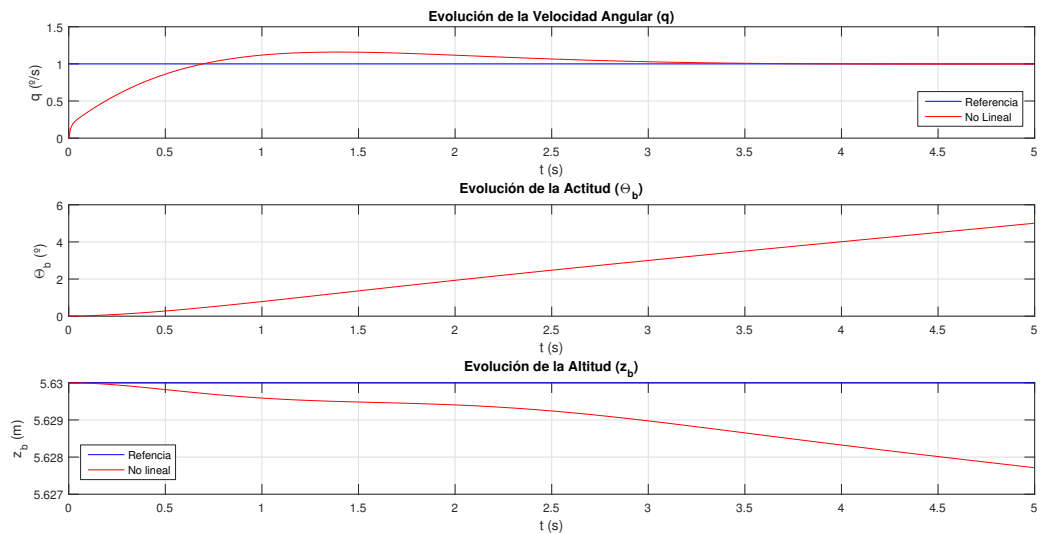
Parámetro	Valor	Observación
RATE_PIT_P	0.14	Kp
RATE_PIT_I	0.14	Ki
RATE_PIT_D	0.00925	Kd
RATE_PIT_IMAX	1000	IMAX
RATE_PIT_FILT_HZ (Hz)	20	FILT

Una vez obtenidas las distintas partes del controlador PID, se suman y se calcula la salida en pitch correspondiente.

$$salida_pitch = p_k + i_k + d_k \quad (4.39)$$

4.2.2 Análisis de la Respuesta en Pitch

De forma similar al caso de alabeo, se ha realizado una prueba en bucle cerrado consistente en introducir una entrada escalón de $1^\circ/s$ en la dirección de cabeceo, dejando nulas el resto de entradas. Los resultados son muy similares a los ya vistos para la dirección de alabeo como se deduce de la Figura 4.8. La velocidad de cabeceo alcanza la referencia en un corto periodo de tiempo, aunque oscilando siempre en torno a ella. El tiempo de subida y la sobreoscilación no son muy grandes, $0.6s$ y 16% respectivamente. Igualmente, el ángulo de cabeceo del quadrotor crece en correspondencia con la velocidad fijada, dando lugar a una caída de la altitud no deseable.

**Figura 4.8** Respuesta de q ante entrada escalón de $1^\circ/s$.

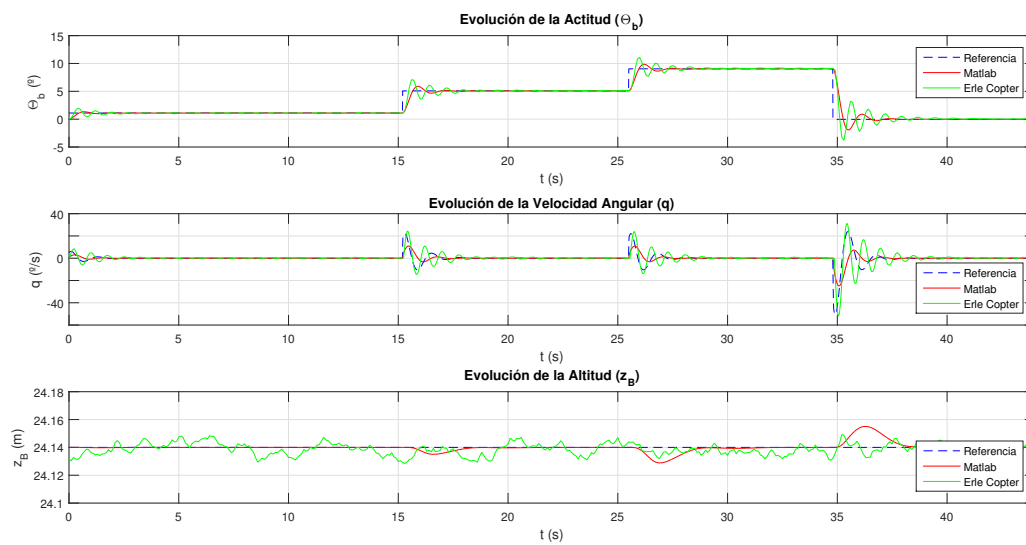
Para ver el resultado final, tras aplicar los esquemas de control desarrollados por ArduPilot al sistema modelado en MATLAB® SIMULINK®, se procede a realizar las pruebas en Gazebo y MATLAB® pertinentes. Esta vez, los PWM serán introducidos a través del canal 2, que es el encargado de estabilizar el ángulo de cabeceo del quadrotor en el valor deseado. Las diferentes señales introducidas y los ángulos generados vienen representados en la Tabla 4.7.

De igual forma, se introducen en SIMULINK® los ángulos deseados anteriores y se corre el sistema formado por la dinámica del quadrotor más el modo Stabilize, dejando el resto de variables de entrada nulas. Para este modelo en MATLAB®, el ángulo de cabeceo alcanza la referencia en un corto periodo de tiempo y

Tabla 4.7 Valores PWM en Gazebo y Ángulo θ_B Deseado.

Canal (rc)	PWM	θ_{Bin} ($^{\circ}$)	Tiempo (s)
2	1550	1.1	0
2	1600	5.08	15.2
2	1650	9.04	25.5
2	1500	0	34.8

con una oscilación moderada como se deduce del transitorio presentado en la Tabla 4.8. Sin embargo, la respuesta del modelo de Erle Copter es más rápida, pero se ve penalizada con una mayor oscilación, tardando más tiempo en conseguir estabilizarse. Este mismo comportamiento se observa en la velocidad angular, expresada en ejes cuerpos, cuya diferencia con respecto al modelo en MATLAB[®] es mayor en las fases de cambio de ángulo.

**Figura 4.9** Respuesta de θ_B ante varias entradas escalón.

La posición del vehículo también se encuentra afectada por el cambio de actitud. Así, la oscilación en altitud viene a ser del mismo orden en ambos modelos y su error es pequeño tal y como se deduce de la Figura 4.10. La distancia en “x” aumentará como consecuencia del cambio de ángulo, que implica una determinada velocidad horizontal. Aunque el ángulo de alabeo no cambie, la dirección “y” se encuentra afectada por el acoplamiento dinámico del ángulo de guiñada.

Tabla 4.8 Análisis de la respuesta temporal de θ_B .

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	0.4	2.1	21
Erle Copter	0.2	15	76

Los errores en ángulo y velocidad se recogen en la Figura 4.10 y, como se ha comentado anteriormente, las mayores diferencias se encuentran durante el transitorio. Así, el mayor error en ángulo entre ambos modelos se produce cuando se desea que el ángulo de cabeceo vuelva a ser nulo y lo mismo ocurre con la velocidad angular generada para conseguirlo. Esta gran diferencia en velocidad angular puede ser consecuencia, no sólo del mayor error en ángulo que genera más perturbación a la entrada del controlador en velocidad, sino

también por la diferente dinámica y por el ruido de dispositivos electrónicos como las IMUs presentes el modelo de Erle Copter.

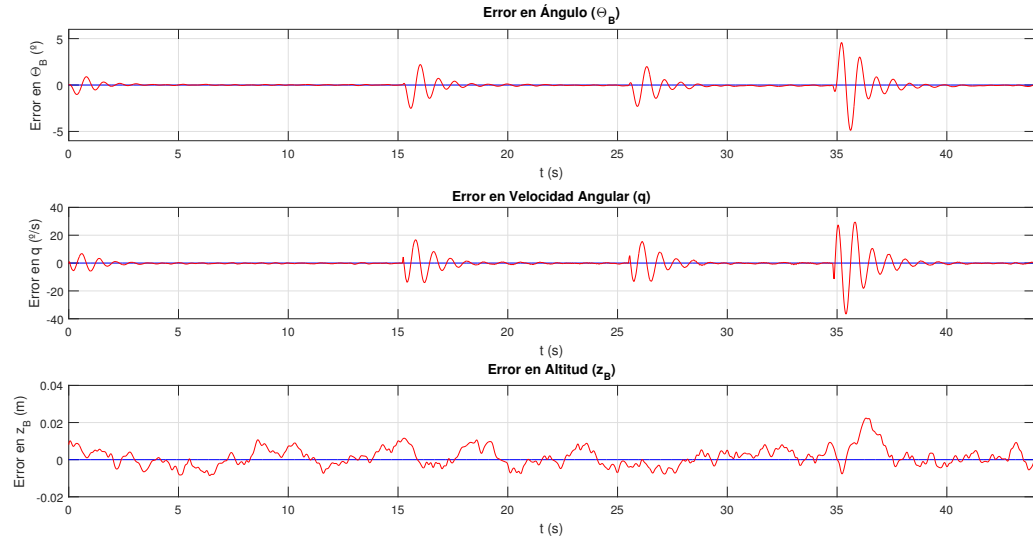


Figura 4.10 Error entre el modelo en Matlab y Gazebo para θ_B .

En este caso, los rotores “1” y “3” son los responsables del cambio de ángulo de cabeceo del vehículo, variando su empuje entre los límites razonables establecidos. Incluso si el máximo ángulo de cabeceo es requerido, los rotores no se encontrarán saturados en ningún momento. La variación del empuje en los rotores “2” y “4” es debida a la pérdida de altitud y su aumento es relativamente pequeño.

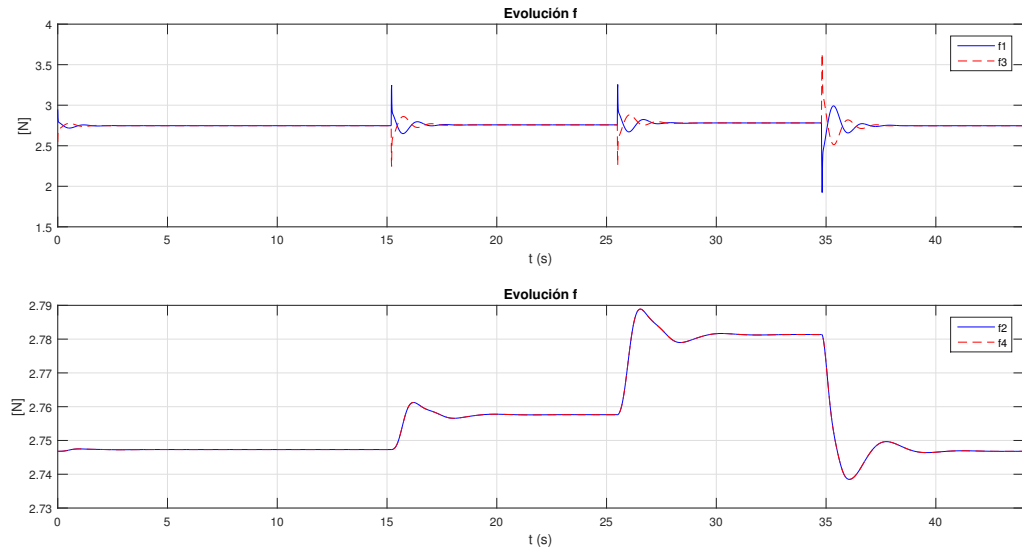


Figura 4.11 Variación de Empuje de los Rotores ante cambios en θ_B .

A la vista de los resultados, se deduce que el esquema de controladores de ArduPilot es un buen método para el control del quadrotor desarrollado en este proyecto. A pesar de la diferente respuesta en pitch, ambos modelos tienen en líneas generales la misma tendencia y estabilizan en pocos segundos al vehículo. Cabría

La velocidad límite determina el mayor incremento que puede experimentar la velocidad prealimentada en cada instante de tiempo.

$$rate_change_limit = accel_yaw_max * T \quad (4.42)$$

Este incremento en velocidad, $rate_change_k$, se obtiene como la diferencia entre la velocidad angular de referencia y la velocidad de guiñada prealimentada del instante anterior. A su vez, tal diferencia en velocidad se verá saturada por la velocidad angular límite.

$$\begin{aligned} rate_change_k &= yaw_rate_ef_k - rate_ef_desired.z_{k-1} \\ -rate_change_limit &\leq rate_change_k \leq rate_change_limit \end{aligned} \quad (4.43)$$

Una vez obtenida la variación de velocidad prealimentada, se actualiza dicha velocidad al nuevo valor sumándole el incremento ya calculado.

$$rate_ef_desired.z_k = rate_ef_desired.z_{k-1} + rate_change_k \quad (4.44)$$

A partir de la velocidad requerida en ejes tierra, se obtiene el ángulo objetivo mediante la integración de esta velocidad y su resultado será empleado para calcular el error en guiñada. Este error viene dado como la diferencia entre el ángulo de guiñada objetivo y el ángulo actual de dron. Posteriormente, se transformará a un valor comprendido entre $\pm 180^\circ$ y se hará uso del máximo error permitido para restringir su valor. Esta saturación en el error constituye un parámetro interno proporcionado por ArduCopter y su valor vendrá dado en centígrados.

$$\begin{aligned} angle_ef_error.z_k &= angle_ef_target.z_{k-1} - C * \psi_{Bk} \\ -1000 &\leq angle_ef_error.z_k \leq 1000 \end{aligned} \quad (4.45)$$

Siendo C la constante establecida para pasar de radianes a centígrados. El ángulo yaw objetivo se actualiza evitando superar la máxima sobreoscilación establecida en el error y se incrementa de acuerdo con la velocidad de guiñada prealimentada. Además, para obtener un resultado razonable se restringe su valor a un ángulo comprendido entre $\pm 360^\circ$.

$$\begin{aligned} angle_ef_target.z_k &= angle_ef_error.z_k + C * \psi_{Bk} \\ angle_ef_target.z_k &= angle_ef_target.z_k + rate_ef_desired.z_k * T \end{aligned} \quad (4.46)$$

Establecido el error en guiñada ($angle_ef_error.z_k$) y la velocidad deseada o prealimentada ($rate_ef_desired.z_k$), se procede a realizar el cambio de ejes tierra a coordenadas del propio cuerpo.

$$angle_bf_error = W(\eta_B) * angle_ef_error \quad (4.47)$$

$$rate_bf_desired = W(\eta_B) * rate_ef_desired \quad (4.48)$$

Una vez obtenido el error en guiñada ($angle_bf_error.z_k$), expresado en coordenadas del propio cuerpo, se pasa su resultado al bloque `sqr_controller_yaw`. Este controlador es idéntico al ya visto anteriormente, pero con la diferencia de que tanto su ganancia proporcional como su aceleración característica no coinciden

con las ya establecidas para los ángulos de alabeo y cabeceo, por lo que su comportamiento será distinto.

Tabla 4.9 Ganancias características del controlador en Yaw y valores máximos.

Parámetro	Valor	
ACRO_YAW_P	3	-
STB_YAW_P	6	-
ANGLE_MAX	3000	Centígrados
ATC_ACCEL_Y_MAX	18000	Centígrados/s ²

Al igual que en las secciones anteriores, se procede a calcular la velocidad angular máxima permitida en guiñada. Esta aceleración es inferior a la ya vista para las direcciones horizontales del vehículo, pues se prioriza el control sobre los ángulo de alabeo y cabeceo frente al de guiñada. Este efecto puede apreciarse también en el uso de una ganancia menor para esta dirección, dando lugar a una estabilización más lenta. Además, una velocidad de guiñada mayor provocaría un par más elevado que podría saturar los rotores.

$$\begin{aligned}
 accel_max &= (ATC_ACCEL_Y_MAX)/2 \\
 kp &= STB_YAW_P \\
 \text{Asegura que } 9000 &\leq accel_max \leq 36000 \\
 rate_bf_target.z_k &= sqrt_controller(angle_bf_error.z_k, kp, accel_max)
 \end{aligned}
 \tag{4.49}$$

El estudio de su comportamiento para diferentes errores en ángulo que van desde los -30° a los 30° se recoge en la Figura 4.13. La zona lineal tiene ahora un tamaño menor y comprende valores inferiores a $\pm 2.5^\circ$. Igualmente, las velocidades máximas alcanzables a la salida se han reducido más de la mitad si las comparamos con las velocidades a la salida de los ángulo de alabeo y cabeceo. Esto se traduciría en una respuesta más lenta del ángulo yaw ante cambios en las condiciones de referencia del mismo.

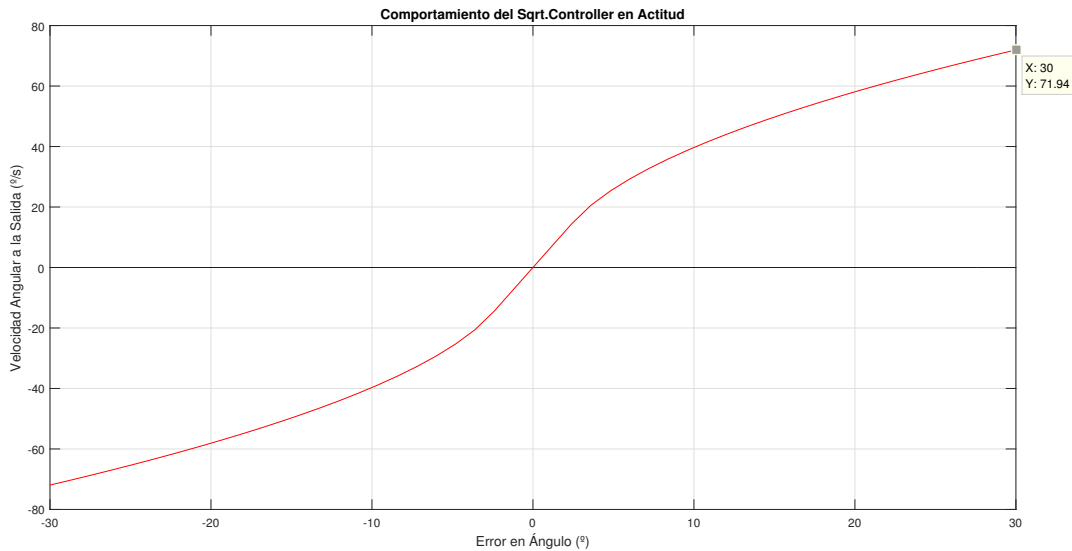


Figura 4.13 Respuesta del *Sqrt_Controller* ante entrada rampa del error en guiñada.

A continuación, se le suma a la velocidad objetivo calculada mediante el controlador en guiñada ($rate_bf_target.z_k$), la velocidad prealimentada obtenida previamente para dicha dirección.

$$rate_bf_target.z = rate_bf_target.z + rate_bf_desired.z \quad (4.50)$$

Finalmente, a partir de la velocidad objetivo en guiñada, se calcula el error en velocidad angular como la diferencia entre dicha velocidad objetivo y la velocidad actual del quadrotor. Este error es luego filtrado antes de correr el controlador PID en guiñada, incluido en $rate_bf_to_motor_yaw$. Se obtiene así una entrada filtrada que será introducida tanto en la parte proporcional como integral del controlador (inp_{fk}), y una derivada filtrada que será utilizada en la parte derivativa del mismo (der_k).

$$e_k = rate_error_yaw = rate_bf_target.z_k - r_k \quad (4.51)$$

$$\begin{aligned} filt &= \frac{T}{T + (2 * \pi * FILT)} \\ inp_{fk} &= inp_{fk-1} + filt * (e_k - inp_{fk-1}) \\ der_k &= \frac{filt * (e_k - inp_{fk-1})}{T} \end{aligned} \quad (4.52)$$

Filtrado el error y obtenida las entradas al controlador del lazo interno, la distintas partes del PID se obtienen de forma idéntica a la ya descritas en la Subsección 4.1.1. Para ello, se multiplica el error filtrado por sus respectivas ganancias y se integra en el caso de la parte integral para después trunca su resultado mediante el término *IMAX*. Las ganancias para este controlador se detallan en la Tabla 4.10, donde se puede apreciar una mayor diferencia entre estos valores y los proporcionados por Ardupilot para los ángulos roll y pitch. Esta mayor variación se justifica por la diferencias existentes entre la dinámica en esta dirección y la establecida para las otras direcciones. Finalmente, la salida total del controlador PID en guiñada es saturada, tanto en sentido positivo como negativo, mediante un valor interno establecido por ArduPilot (± 4500).

Tabla 4.10 Ganancias características del controlador RATE_YAW.

Parámetro	Valor	Observación
RATE_YAW_P	0.40	Kp
RATE_YAW_I	0.04	Ki
RATE_YAW_D	0.004	Kd
RATE_YAW_IMAX	1000	IMAX
RATE_YAW_FILT_HZ (Hz)	5	FILT

La respuesta final en dirección yaw es la suma de las distintas partes del controlador PID.

$$salida_yaw = p_k + i_k + d_k \quad (4.53)$$

4.3.2 Análisis de la Respuesta en Yaw

En primer lugar, se prueba el sistema formado por el controlador PID más quadrotor en la dirección de guiñada. Al igual que en las secciones anteriores, se introduce una velocidad escalón de $1^\circ/s$ en la dirección de guiñada, dando como resultado la respuesta presentada en la Figura 4.14. De la gráfica, se deduce el poco tiempo que tarda la velocidad angular del quadrotor en alcanzar la velocidad deseada, aproximadamente $0.6s$, y la pequeña oscilación con la que lo hace, menos del 1 %. Igualmente, el ángulo de guiñada del vehículo

responde bien al cambio en velocidad establecido y la altitud, aunque decae debido fundamentalmente a la variación del ángulo, se consigue recuperar con el tiempo.

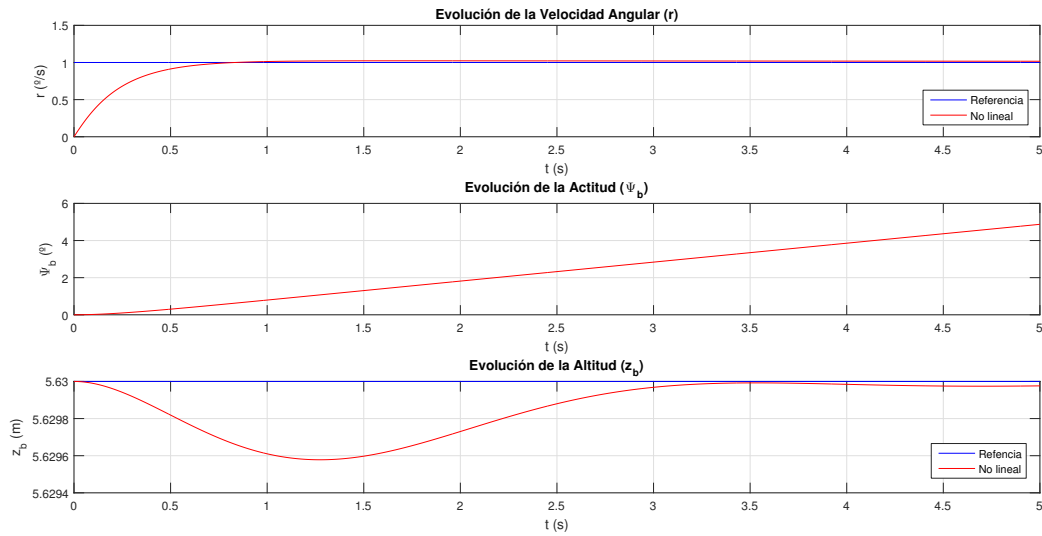


Figura 4.14 Respuesta de r ante entrada escalón de $1^\circ/\text{s}$.

Aunque no se hayan representado tanto la actitud en alabeo como en cabeceo del vehículo, estas no se verán afectadas por los cambios ocurridos en la velocidad de guiñada, y por consiguiente, la posición horizontal del quadrotor permanece sin variaciones.

A continuación, se aplica el esquema de control completo a la dirección de guiñada en el modelo matemático y se compara su respuesta con el modelo de Erle Copter. El experimento realizado en Gazebo consiste en introducir varias señales escalón a través del canal 4, presentando la problemática de responder lentamente a los PWM y con las desventaja de tener una gran oscilación tanto en cabeceo como alabeo que desestabiliza al vehículo. Además, la emisora que controla la velocidad de guiñada en Gazebo no responde de forma instantánea a los cambios introducidos en velocidad, teniendo un sistema reactivo a cambiar de dirección. El inconveniente principal para modificar la guiñada es la elevada potencia necesaria para esta labor en comparación con cambios equivalentes en alabeo o cabeceo, pudiendo saturar los rotores cuando un determinado ángulo de inclinación es requerido al mismo tiempo.

Tabla 4.11 Valores PWM en Gazebo y Velocidad ψ_B Deseada.

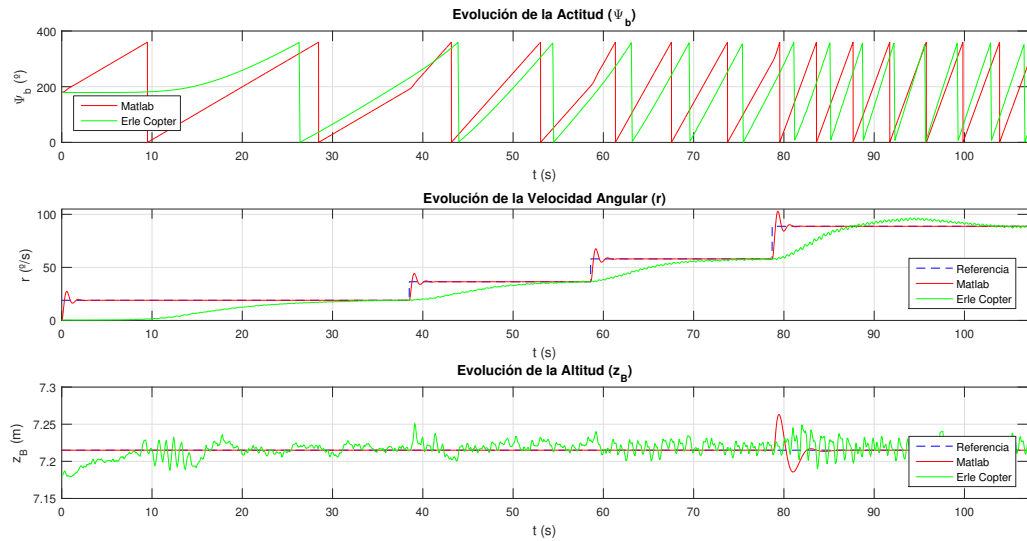
Canal (rc)	PWM	ψ_{Bin} ($^\circ/\text{s}$)	Tiempo (s)
4	1600	6.3	0
4	1700	12.15	38.5
4	1800	19.33	58.6
4	1900	29.58	78.7

A partir de los datos en velocidad de guiñada proporcionados por la Tabla 4.11, se efectúa la correspondiente prueba en SIMULINK[®]. Esta arroja resultados totalmente distintos al experimento realizado en Gazebo, pues la aplicación del esquema de control proporcionado por ArduPilot al modelo en MATLAB[®], supone estabilizar la velocidad de guiñada en mucho menos tiempo. Prueba de ello son los datos de tiempo y oscilación proporcionados en la Tabla 4.12, donde se observa la rapidez del sistema en MATLAB[®] para alcanzar la velocidad deseada frente a la lentitud del mismo proceso en Gazebo.

Tabla 4.12 Análisis de la respuesta temporal de ψ_B .

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	0.3	2.1	44
Erle Copter	31	35	1

De la misma forma, el ángulo de guiñada tiene similar evolución en ambos casos, llegando un momento en que el ángulo en MATLAB[®] adelanta al de Gazebo, pues el primero alcanza la velocidad deseada en menos tiempo. Por otra parte, la oscilación en altitud tiene el mismo orden de magnitud para ambos modelos como se deduce de la Figura 4.15, aunque existe una mayor oscilación en Gazebo debida quizás al ruido de los sensores.

**Figura 4.15** Respuesta de ψ_B ante entradas escalón en guiñada.

En la Figura 4.16, se representan las grandes oscilaciones en los ángulos de alabeo y cabeceo que han sido comentadas previamente. Valores de hasta 20° son alcanzados en Gazebo mientras que dichos ángulos no se encuentran afectados en el modelo de SIMULINK[®]. Esto viene a reafirmar las diferencias expuestas entre ambos modelos para el control de la guiñada y en la repuesta inestable del vehículo ante variaciones de la misma.

Para cambiar la velocidad de guiñada, es necesario actuar simultáneamente sobre los cuatro rotores que forman parte del vehículo tal y como se observa en la Figura 4.17. El hecho de cambiar el ángulo de guiñada supone un mayor empuje requerido por parte de los rotores. Estos se encontrarán en el límite de los valores razonables establecidos y generarán una pérdida de control efectivo sobre el vehículo.

Con el fin reducir las diferencias más que evidentes en el ángulo de guiñada, se podría dar un paso más lejos e implementar el bloque Stability Patch, recogido en [25]. Este bloque intenta gestionar la potencia total del quadrotor cuando maniobras de alabeo, cabeceo y guiñada están ocurriendo simultáneamente, intentando además priorizar el mantenimiento de la altitud. En definitiva, da mayor relevancia al control de la inclinación o altitud del quadrotor, reservando la potencia mínima necesaria para mantener la dirección o rumbo establecido. La no inclusión de este bloque junto con las diferencias en el modelado dinámico del vehículo pueden ser las causantes de la diferente respuesta en guiñada.

Aunque el modelo de control diseñado para el ángulo de guiñada es bastante lento e introduce una importante oscilación en el sistema en el caso de Gazebo, no tendrá un efecto negativo sobre el objetivo final de controlar la posición de la carga. La razón de ello, es que el control de la posición depende fundamentalmente de los

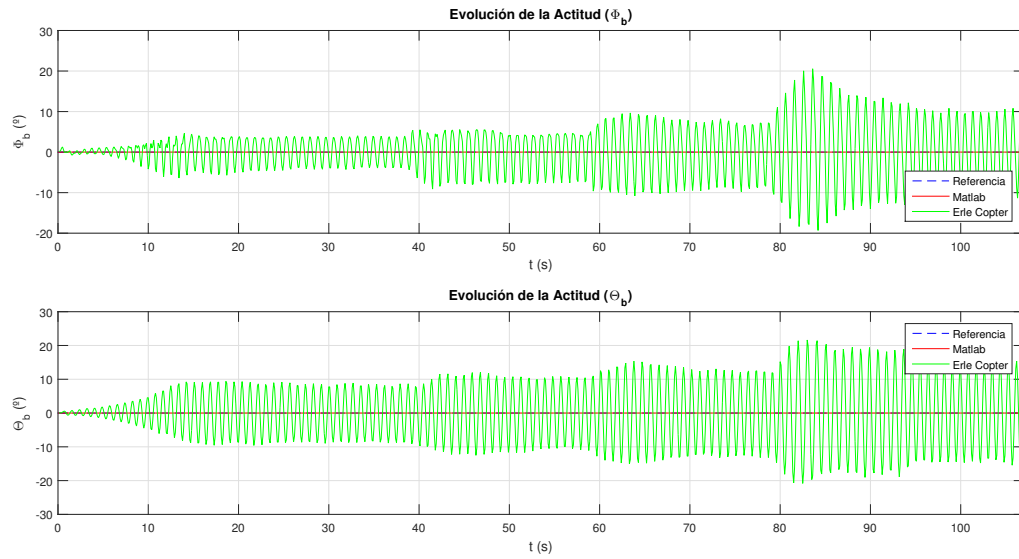


Figura 4.16 Respuesta de ϕ_B y θ_B ante entradas escalón en guiñada.

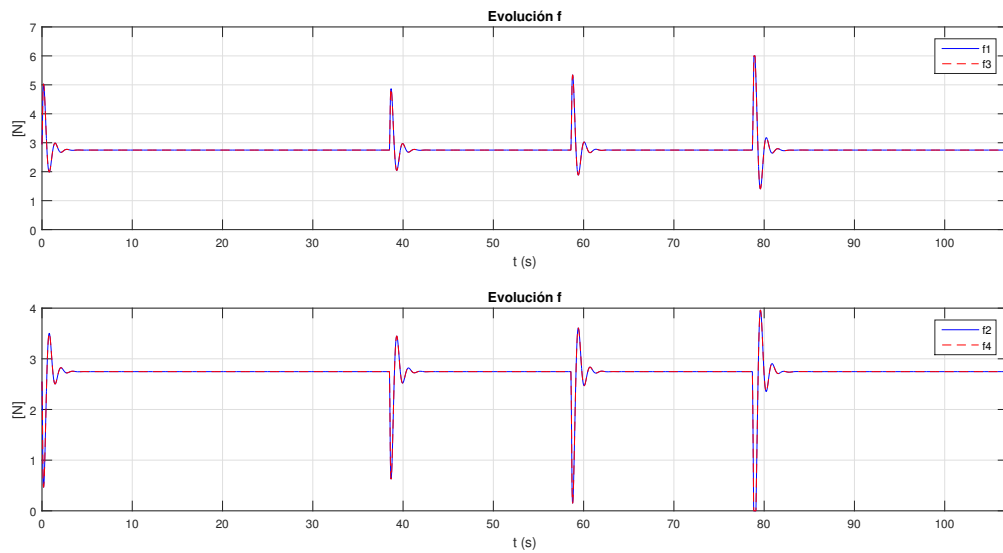


Figura 4.17 Variación de Empuje de los Rotores ante cambios en ψ_B .

ángulos de cabeceo y alabeo. Por tanto, será el diseño de estos controladores el que más influya en el control direccional de la carga.

5 Modo Alt-Hold

En el modo Alt-Hold o modo mantiene la altitud, se intenta estabilizar la altura del vehículo en la posición fijada cuando los mandos se encuentran en la posición intermedia o las entradas al sistema son nulas. La actitud del quadrotor es controlada usando los esquemas de control propuestos en el Capítulo 4 y los mandos o señales de referencia tendrán un control directo sobre los ángulos de alabeo y cabeceo y la velocidad de guiñada del mismo. Para mas información sobre este modo el lector es referido a [2].

Para la altitud, se ha optado por controlar directamente la velocidad de ascenso y descenso del quadrotor siguiendo las pautas establecidas por ArduPilot en su versión *Copter 3.3.1 26-Oct-2015*. De esta forma, cuando dicha velocidad es nula, el quadrotor mantiene la altitud a la que se encuentra independientemente de su actitud. Para poder llevar a cabo esta tarea, se han desarrollado un conjunto de tres controladores PID anidados en aceleración, velocidad y posición. El controlador ACCEL_Z, genera a partir del error en aceleración, la señal de actuación necesaria para cambiar el empuje total del quadrotor. El controlador superior VEL_Z, calcula a partir de la diferencia entre la velocidad deseada y real, la aceleración objetivo que alimenta al lazo inferior. Por último, el controlador POS_Z, fija la velocidad de referencia a partir del error en altitud. Además, complementando a estos tres controladores, se encuentra un integrador que convierte la velocidad vertical deseada en una altitud de referencia que alimentará a todo el sistema. Su modelo en SIMULINK® se ha representado en la Figura 5.1.

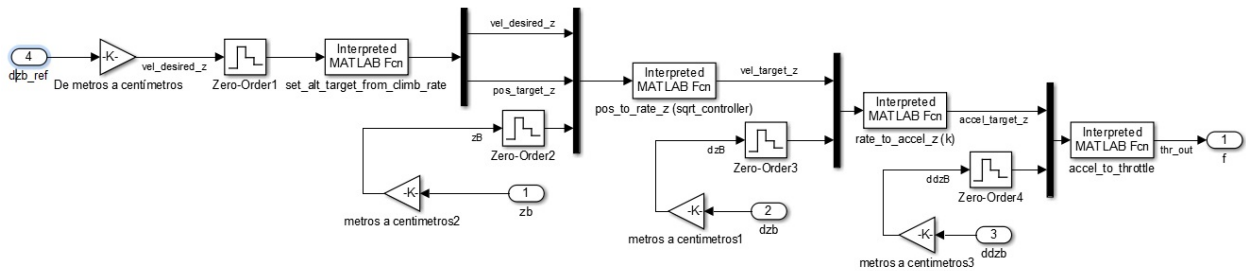


Figura 5.1 Esquema para el control de la altura en SIMULINK®.

Este Capítulo se dividirá en dos apartados. En el primero, se detallan los pasos seguidos para el control de la posición vertical del quadrotor, desarrollando para ello los controladores PID en cascada y las limitaciones impuestas a sus valores. En el segundo, se presentan los resultados obtenidos cuando se desea actuar sobre aceleración, velocidad y posición del quadrotor y, finalmente, la respuesta completa del sistema en velocidad vertical. Estos resultados finales se compararán con los obtenidos para el modelo en Gazebo de ErleCopter.

5.1 Esquema de Control para Alt-Hold

En primer lugar, se establecen las velocidades de ascenso y descenso máximas junto con la aceleración, expresadas en cm/s y cm/s^2 respectivamente. Estas variables serán usadas a lo largo de todo el proceso de control de la altitud, por lo que se establecerán al principio y no tendrán que ser llamadas de forma reiterativa. La variable lógica, *flags_speed_updown_cms*, será la encargada de activar el cálculo de las mismas.

```
if flags_speed_updown_cms == 1
    [speed_down_cms, speed_up_cms] = set_speed_z(-PILOT_VELZ_MAX, PILOT_VELZ_MAX)
    accel_z_cms = set_accel_z(PILOT_ACCEL_Z)
    flags_speed_updown_cms = 0
end
```

(5.1)

Tanto la velocidad como la aceleración tendrán asignados unos valores por defecto que pueden ser iguales o distintos de los introducidos por el usuario mediante el APMPlanner. Los parámetros *PILOT_VELZ_MAX* y *PILOT_ACCEL_Z* pueden consultarse en la Tabla 5.1 y son establecidos como modificables en el APM-Planner. El comportamiento de la función *set_speed_z* puede consultarse en el Código 5.1 y, en resumen, reasigna los valores proporcionados por el usuario a las variables que representan la velocidad límite.

Código 5.1 Código para establecer las velocidades de ascenso y descenso máximas.

```
function [speed_down_cms, speed_up_cms] = set_speed_z(speed_down, speed_up)

% Asigna valores por defecto a las velocidad de ascenso y descenso.
speed_down_cms = -150;
speed_up_cms = 250;

% Asegura que speed_down sea siempre negativa
speed_down = -abs(speed_down);
if abs(speed_down_cms-speed_down) > 1 | abs(speed_up_cms-speed_up) > 1
    % Establece el nuevo valor para la velocidad de descenso máxima
    speed_down_cms = speed_down;
    % Establece el nuevo valor para la velocidad de ascenso máxima
    speed_up_cms = speed_up;
end
```

La función *set_accel_z* es similar a la anterior y su objetivo es sustituir la aceleración máxima establecida por defecto y asignarle la proporcionada por el usuario. Su código se puede consultar en 5.2.

Código 5.2 Código para establecer la aceleración máxima.

```
function accel_z_cms = set_accel_z(accel_cmss)

% Define la aceleración por defecto
accel_z_cms = 250;

if abs(accel_z_cms-accel_cmss) > 1
    % Establece el nuevo valor para la aceleración vertical
end
```

```

    accel_z_cms = accel_cmss;

end

```

Una vez asignadas la velocidad y aceleración máximas, se ejecuta el bloque *set_alt_target_from_climb_rate* y se ajusta el objetivo de subida o bajada usando la velocidad de entrada, *climb_rate.z_k*. Esta viene expresada en centímetros por segundo y se encuentra discretizada en el tiempo. Además, la altitud actual no deberá ser movida más rápido que la velocidad de ascenso y descenso establecidas, y la distancia restringida en dirección vertical no deberá superarse en ningún caso.

Para conseguir la altitud objetivo, se define el diferencial de aceleración *jerk_z* como aquel que permite alcanzar la aceleración requerida en el tiempo establecido por la variable interna *Poscontrol_Jerk_Ratio*. Esta constante viene dada internamente en el código y su valor es igual a 1, es decir, busca conseguir dicha aceleración en aproximadamente un segundo.

$$jerk_z = accel_z_cms * Poscontrol_Jerk_Ratio \quad (5.2)$$

A partir de *jerk_z*, se calcula la aceleración máxima en dirección vertical y se busca que no sobrepase el valor de la aceleración calculada en 5.1. Esta vendrá dada como la raíz cuadrada de una diferencia en velocidad por un diferencial de aceleración. La aceleración transitoria obtenida se empleará solo en el cálculo de la altitud objetivo, utilizándose *accel_z_cms* para los restantes casos.

$$\begin{aligned}
 accel_z_max_k &= \sqrt{2 * |vel_desired.z_{k-1} - climb_rate.z_k| * jerk_z} \\
 accel_z_max_k &= \min(accel_z_cms, accel_z_max_k)
 \end{aligned} \quad (5.3)$$

A su vez, se va actualizando la aceleración última para cada intervalo de tiempo transcurrido, teniendo en cuenta el *jerk_z* previamente establecido. Se establece que esta nueva aceleración no supere nunca la aceleración máxima calculada en 5.3, es decir, irá creciendo de acuerdo con *jerk_z* hasta alcanzar el límite último establecido.

$$\begin{aligned}
 accel_last.z_k &= accel_last.z_{k-1} + jerk_z * T \\
 accel_last.z_k &= \min(accel_z_max_k, accel_last.z_k)
 \end{aligned} \quad (5.4)$$

Siendo $T = 0.01$ el intervalo de tiempo transcurrido entre una llamada y otra de la función. Obtenida la aceleración requerida, se define la velocidad de cambio que será empleada para restringir la velocidad vertical de entrada al sistema. Se obtiene así la velocidad prealimentada que será sumada a la salida del controlador *sqrtr_controller* en altitud.

$$\begin{aligned}
 vel_change_limit_k &= accel_last.z_k * T \\
 vel_desired.z_k &= climb_rate.z_k \\
 vel_desired.z_{k-1} - vel_change_limit_k &\leq vel_desired.z_k \leq vel_desired.z_{k-1} + vel_change_limit_k
 \end{aligned} \quad (5.5)$$

A partir de la velocidad prealimentada, la posición objetivo se obtiene mediante la integración de la misma en el tiempo. El valor resultante será empleado para calcular el error en altitud y así poder controlar la

dirección vertical del vehículo.

$$pos_target.z_k = pos_target.z_{k-1} + vel_desired.z_k * T \quad (5.6)$$

Como paso previo a calcular la diferencia de altitud, se procede a obtener las restricciones en distancia establecidas para el error. Estas dependerán no solo de la velocidad y aceleración máximas fijadas para el dron en dirección vertical, sino también de la ganancia proporcional.

```
if flags_recalc_leash.z == 1
    leash_up_z = calc_leash_length(speed_up_cms, accel_z_cms, POS_Z_P)
    leash_down_z = calc_leash_length(-speed_down_cms, accel_z_cms, POS_Z_P)
    flags_recalc_leash.z = 0
end
```

(5.7)

Estas restricciones en longitud pueden interpretarse como la distancia límite por encima de la cual se superaría la máxima velocidad vertical permitida para el quadrotor ($speed_cms = POS_Z_P * leash_length$). Siguiendo con las unidades establecidas, su cálculo se realizará en centímetros en lugar de metros.

Código 5.3 Código para calcular las restricciones horizontales y verticales.

```
function leash_length=calc_leash_length(speed_cms, accel_cms, kP)

% Chequea la validez de la aceleración.
% El valor 50 cm/s^2 es un parámetro interno proporcionado por ArduCopter
if accel_cms <= 0
    accel_cms = 50;
end

% Evita dividir por cero y ganancia negativas.
if kP <= 0
    leash_length = 100;           % Fija la restricción a 1 m
end

% Calcula el tamaño de la restricción
if speed_cms <= accel_cms / kP
    % El tamaño de la restricción estará basado en la velocidad cuando se
    % está cerca
    leash_length = speed_cms / kP;
else
    % La distancia restringida crece como la raíz cuadrada de la
    % velocidad cuando se requiere cubrir distancias mayores
    leash_length = (accel_cms / (2*kP*kP)) + (speed_cms*speed_cms / (2*
        accel_cms));
end

% Asegura que el tamaño sea al menos de 1 metro de largo
if leash_length < 100
    leash_length = 100;
end
```

A partir de la altitud objetivo generada por el bloque superior, $pos_target.z_k$, se calcula el error en altitud como la diferencia entre la altitud objetivo y la actual del dron. Su valor vendrá dado en centímetros y sus

expresiones se encuentran en $pos_to_rate_z$.

$$pos_error.z_k = pos_target.z_k - z_{Bk} \quad (5.8)$$

Con el fin de evitar que la diferencia de altitud se aleje mucho de la actual, se restringe su valor mediante la distancia superior ($leash_up_z$) e inferior ($leash_down_z$) establecidas.

$$\begin{aligned} & \text{if } pos_error.z_k > leash_up_z \\ & \quad pos_target.z_k = z_{Bk} + leash_up_z \\ & \quad pos_error.z_k = leash_up_z \\ & \text{end} \end{aligned} \quad (5.9)$$

$$\begin{aligned} & \text{if } pos_error.z_k < -leash_down_z \\ & \quad pos_target.z_k = z_{Bk} - leash_down_z \\ & \quad pos_error.z_k = -leash_down_z \\ & \text{end} \end{aligned} \quad (5.10)$$

El control de la altitud se efectúa mediante el *sqrt_controller*, el cual ha sido ya estudiado en la Subsección 4.1.1 para el ángulo roll. Este controlador tendrá en líneas generales la misma tendencia tanto para la posición como para la actitud y, a partir del error en altitud, calcula la velocidad objetivo que alimentará al controlador en velocidad del lazo interno. La expresión de la aceleración empleada se puede consultar en fórmula 5.1 y el valor de la ganancia proporcional en la Tabla 5.1.

Tabla 5.1 Ganancias características del control en altitud y valores máximos.

Parámetro	Valor	Observación
POS_Z_P	1	-
VEL_Z_P	5	-
PILOT_VELZ_MAX	250	cm/s
PILOT_ACCEL_Z	250	cm/s ²

$$vel_target.z_k = sqrt_controller(pos_error.z_k, POS_Z_P, accel_z_cms) \quad (5.11)$$

Con el fin de visualizar el comportamiento que tiene este controlador cuando es usado en altitud, se realizó una prueba en bucle abierto para diferentes valores del error. Estos crecían en el rango de $\pm 2.5m$, que se corresponde con el intervalo entre las distancias límite superior e inferior. A la vista de los resultados de la Figura 5.2, está claro que el comportamiento del *sqrt_controller* es el de un controlador P. Este comportamiento es debido no solo a las restricciones en distancia, sino también a los valores de aceleración máxima y ganancia que se le han asignado.

Obtenida la velocidad objetivo, se le suma ahora la velocidad deseada o prealimentada generada en la expresión 5.5. Su resultado será finalmente la velocidad objetivo buscada.

$$vel_target.z_k = vel_target.z_k + vel_desired.z_k \quad (5.12)$$

El cálculo del error en velocidad y la aceleración generada, se llevarán a cabo en el bloque conocido

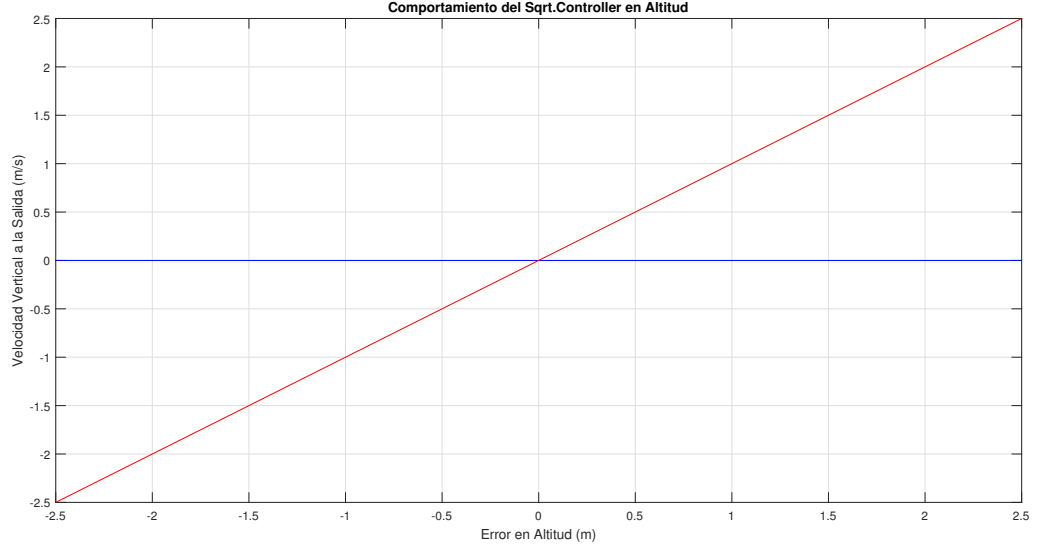


Figura 5.2 Respuesta del *Sqrt_Controller* ante entrada rampa del error en altitud.

como *rate_to_accel_z*. Se comenzará verificando la validez de la velocidad vertical objetivo, *vel_target.z_k*, de acuerdo con los límites establecidos por ArduCopter para la misma. La máxima velocidad de ascenso permitida vendrá dada por la variable *speed_up_cms* y su mínimo valor de descenso por la constante *speed_down_cms*, calculadas ambas en la expresión 5.1.

$$\begin{aligned}
 & \text{if } vel_target.z_k < speed_down_cms \\
 & \quad vel_target.z_k = speed_down_cms \\
 & \text{end}
 \end{aligned}
 \tag{5.13}$$

$$\begin{aligned}
 & \text{if } vel_target.z_k > speed_up_cms \\
 & \quad vel_target.z_k = speed_up_cms \\
 & \text{end}
 \end{aligned}
 \tag{5.14}$$

La aceleración prealimentada se calcula como la diferencia entre la velocidad objetivo actual y la velocidad objetivo del instante anterior. Esta se sumará posteriormente a la aceleración objetivo, obtenida a partir del error en velocidad, y alimentarán conjuntamente al controlador PID en aceleración.

$$\begin{aligned}
 & \text{if } T > 0 \\
 & \quad accel_feedforward.z_k = (vel_target.z_k - vel_last.z) / T \\
 & \text{else} \\
 & \quad accel_feedforward.z_k = 0 \\
 & \text{end}
 \end{aligned}
 \tag{5.15}$$

La variable $vel_last.z$ almacena la velocidad objetivo actual para poder ser usada en la próxima iteración. Su actualización es necesaria para poder calcular la aceleración prealimentada.

$$vel_last.z = vel_target.z_k \quad (5.16)$$

El error en velocidad se obtiene ahora como la diferencia entre la velocidad objetivo que se viene utilizando y la velocidad vertical del drone actual. Además, este error se resetea nada más llamar al controlador, evitando así acumular posibles errores provenientes de otros procesos.

$$\begin{aligned} & \text{if } flags_reset_rate_to_accel.z == 1 \\ & \quad vel_error.z_k = 0 \\ & \quad flags_reset_rate_to_accel.z = 0 \\ & \text{else} \\ & \quad vel_error.z_k = vel_target.z_k - \dot{z}_{Bk} \\ & \text{end} \end{aligned} \quad (5.17)$$

Donde $\dot{z}_{Bk}$ se ha multiplicado previamente por 100 para pasar de metros a centímetros y, además, se ha discretizado en el tiempo para poder operar con ella. A partir del error en velocidad, la aceleración objetivo en dirección vertical se calcula como el producto de dicho error por la ganancia proporcional correspondiente (VEL_Z_P), cuyo valor puede consultarse en la Tabla 5.1.

$$accel_target.z_k = VEL_Z_P * vel_error.z_k \quad (5.18)$$

Para finalizar con el control de la velocidad vertical, se suma a la aceleración objetivo, la aceleración prealimentada calculada previamente en la expresión 5.15.

$$accel_target.z_k = accel_target.z_k + accel_feedforward.z_k \quad (5.19)$$

Una vez establecida la aceleración vertical objetivo ($accel_target.z_k$) y tras discretizar dicha señal en el tiempo mediante un mantenedor de orden cero, se efectúa el cálculo del error en aceleración. Este vendrá dado en cm/s^2 en correspondencia con las unidades con las que trabaja el controlador ACCEL_Z, contenido en el bloque $accel_to_throttle$.

$$accel_error.z_k = accel_target.z_k - \ddot{z}_{Bk} \quad (5.20)$$

Donde $\ddot{z}_{Bk}$ es la aceleración real del vehículo en el instante k .

A partir de la diferencia entre la aceleración objetivo y la actual, se obtiene las entradas al controlador PI. Sólo la entrada derivativa será filtrada siguiendo el procedimiento descrito en la ecuación 4.21, aunque su efecto sobre el empuje total será nulo debido a que su coeficiente derivativo también lo es.

La parte proporcional se obtiene multiplicando el error en aceleración por una ganancia proporcional cuyo valor se recoge en la Tabla 5.2.

$$p.z_k = Kp * accel_error.z_k \quad (5.21)$$

La parte integral se calcula integrando el error de forma discreta y multiplicando posteriormente su valor por una ganancia integral (Ki). Su resultado es limitado por la constante (IMAX), evitando así un empuje

Tabla 5.2 Ganancias características del controlador ACCEL_Z.

Parámetro	Valor	Observación
ACCEL_Z_P	0.5	Kp
ACCEL_Z_I	1	Ki
ACCEL_Z_D	0	Kd
ACCEL_Z_IMAX	800	IMAX
ACCEL_Z_FILT_HZ (Hz)	20	FILT

muy elevado que pueda provocar que los rotores se encuentren saturados.

$$i.z_k = i.z_{k-1} + Ki * accel_error.z_k * T$$

$$Asegura \quad que \quad -IMAX \leq i.z_k \leq IMAX \quad (5.22)$$

Siendo $T = 0.01$ el tiempo de muestreo del controlador discreto. Finalmente, al resultado calculado por el controlador PI, se le suma el throttle de equilibrio obtenido como el producto de la masa total del cuerpo (m_B) por la gravedad (g).

$$throttle_hover = g * m_B * 100$$

$$thr_out_k = p.z_k + i.z_k + throttle_hover \quad (5.23)$$

Esta salida es enviada directamente a las ecuaciones dinámicas del vehículo y se multiplica por una ganancia proporcional para poder transformar dicha salida en una fuerza compatible con el sistema de ecuaciones. Su valor es $F_Z = 0.01$, y es equivalente a pasar de centímetros a metros.

5.2 Análisis de la Respuesta en Alt-Hold

Una vez obtenidas las expresiones que permiten el control de la altitud del vehículo, se realiza el análisis de los distintos bloques que componen el esquema desarrollado previamente. Para estudiar el comportamiento de la aceleración vertical, se introduce una entrada escalón de $1m/s^2$ y se dejan nulas las entradas en alabeo, cabeceo y guiñada del Stabilize. Obviamente, los bloques afectados por la entrada escalón son el *accel_to_throttle* y el subsistema que contiene las ecuaciones dinámicas del drone.

La respuesta de la aceleración se ha dibujado en la Figura 5.3, y se han representado también la velocidad y posición vertical del vehículo. La aceleración deseada es alcanzada en aproximadamente 4s y sin oscilación. Este controlador PI, genera una respuesta muy rápida debido a la parte proporcional, luego esta componente disminuye al decaer el error y comienza a actuar la parte integral algo más lenta. La actitud del drone no se ha representado debido a que no se ve alterada por cambios en la altitud y lo mismo ocurre con la posición horizontal del mismo.

Para el seguimiento de la velocidad vertical, se ha empleado básicamente un controlador P cuyo funcionamiento se recoge en el bloque *rate_to_accel_z*. Igualmente, se ha introducido una velocidad vertical de $1m/s$, manteniendo nulas el resto de entradas al sistema. Se estudia así el comportamiento conjunto de los controladores en velocidad y aceleración vertical.

Los resultados se han presentado en la Figura 5.4, y se observa una rápida respuesta de la velocidad que alcanza la referencia en menos de 1s y con una oscilación no muy grande (15%). Las velocidades máximas de $2.5m/s$ son alcanzadas sin que los rotores lleguen a saturarse en ningún momento. La aceleración aumenta hasta alcanzar la velocidad deseada y luego disminuye hasta hacerse nula como cabría de esperar. Señalar también, que tanto la posición horizontal del quadrotor como la actitud del mismo no se ven afectadas por

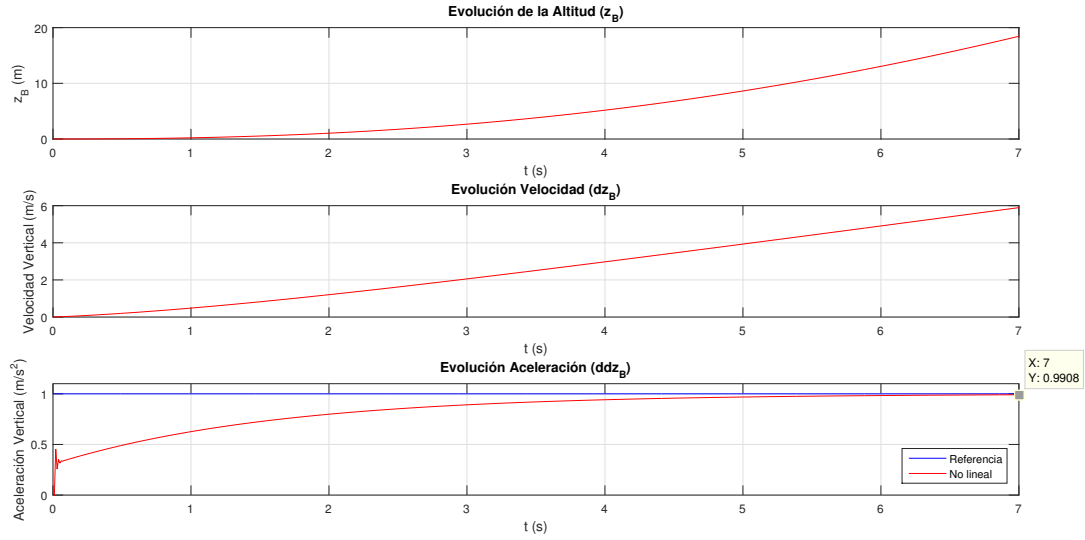


Figura 5.3 Respuesta de $\ddot{z}_B$ ante entrada escalón de $1m/s^2$.

los cambios en la velocidad de ascenso o descenso que pudieran tener lugar.

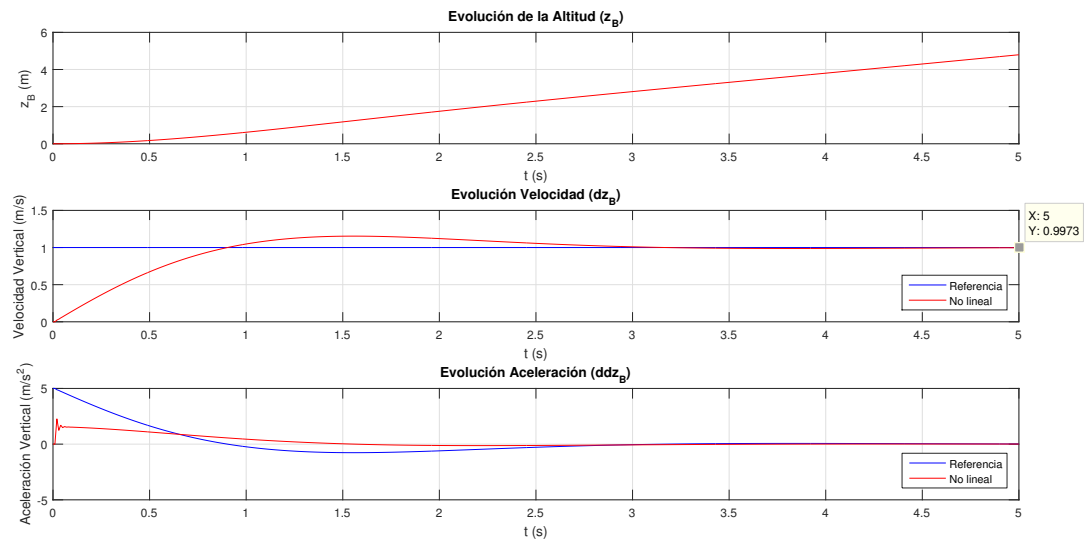


Figura 5.4 Respuesta de $\dot{z}_B$ ante entrada escalón de $1m/s$.

A la vista de los resultados, queda claro que se podría controlar la velocidad vertical del vehículo usando un controlador en velocidad junto con uno en aceleración. Sin embargo, para poder mantener la altitud y que esta se encuentre en una posición fija cuando deje de actuar la señal de control, es necesario colocar un controlador de posición superior para tal misión. Este se encuentra en el bloque *pos_to_rate_z* y para ver su integración con el resto de bloques, se introduce una entrada escalón de $1m$ en el sistema.

La respuesta en altitud está representada en la Figura 5.5 y genera muy buenos resultados dado que se consigue alcanzar la posición requerida en poco tiempo ($2s$) y sin oscilación. Por otra parte, tanto la velocidad como la aceleración vertical aumentan en concordancia con el desplazamiento de ascenso para luego disminuir una vez que se ha alcanzado la posición vertical deseada. Además, el valor máximo alcanzado

por la velocidad se encuentra dentro del límite razonable establecido para la misma y aunque la diferencia de altitud sea muy grande, no se podrá ascender a más de 2.5m/s .

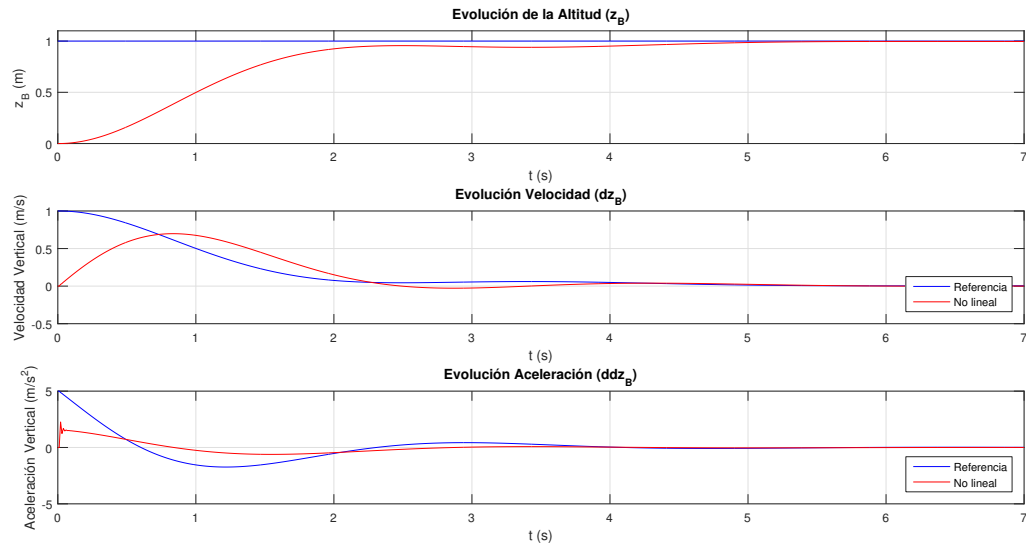


Figura 5.5 Respuesta de z_B ante entrada escalón de 1m .

Para finalizar esta sección, se realiza una comparación gráfica entre el modelo no lineal desarrollado en el Capítulo 2 y el modelo en Gazebo de ErleCopter para varias entradas escalón. Debido a la simplificaciones tomadas en la obtención del modelo dinámico del drone, será difícil que su respuesta coincida plenamente en el tiempo con la del modelo de ErleCopter. Así pues, se busca que sean lo más parecidas posibles dado que se ha desarrollado el mismo procedimiento empleado por ArduPilot para el control de su altitud. La prueba en Gazebo se ha llevado a cabo para varios valores de la emisora o PWM que generan a su vez distintos valores de velocidad de ascenso. En la Tabla 5.3 se puede consultar la correspondencia entre los anchos de pulso y la señal de referencia generada.

Tabla 5.3 Valores PWM en Gazebo y Velocidad $\dot{z}_B$ Generada.

Canal (rc)	PWM	$\text{climb_rate.z}_k \text{ (m/s)}$	Tiempo (s)
3	1600	0.43	0
3	1650	0.81	11.4
3	1700	1.12	20.5
3	1500	0	31.9

Para el modelo en SIMULINK® se introdujeron las velocidades de ascenso presentadas en la Tabla 5.3, de forma que para cada intervalo de tiempo transcurrido se activa la señal escalón correspondiente. Al igual que en los casos anteriores, en la Figura 5.6 se representan la posición, la velocidad y la aceleración vertical resultantes de las pruebas realizadas en Simulink y Gazebo. Ni la posición horizontal ni la actitud del quadrotor se han representado, ya que los cambios en la velocidad de ascenso no influirán en dichas magnitudes.

Como se deduce de la figura anterior, los dos modelos de quadrotor alcanzan la velocidad de ascenso deseada en muy poco tiempo, aunque presentan diferentes características temporales. Así, el modelo en Gazebo de ErleCopter adquiere la velocidad objetivo en un tiempo menor que el modelo desarrollado en SIMULINK® una vez que se ha estabilizado el drone. También se observa una inevitable oscilación de ambas

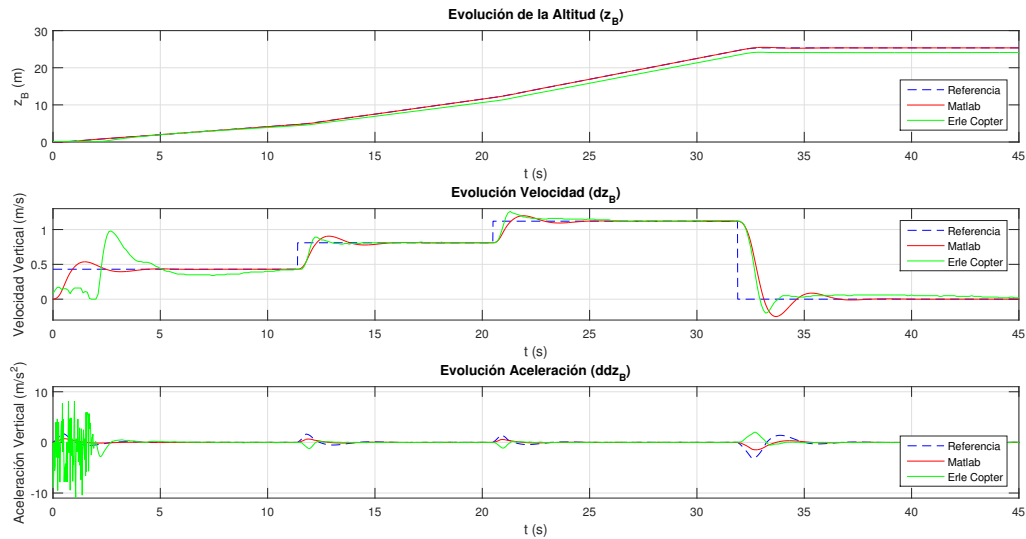


Figura 5.6 Respuesta de $\dot{z}_B$ ante varias entradas escalón.

respuestas en torno a la referencia y un tiempo adicional para armar motores en el caso de ErleCopter. Este arranque de motores tiene un efecto perjudicial sobre la aceleración que oscila de forma muy rápida por lo que se tarda más en despegar y ponerse en vuelo. La Tabla 5.4, recoge los tiempos de subida, establecimiento y la sobreoscilación de la respuesta, constatando también las diferencias existentes entre los modelos en SIMULINK® y ErleCopter.

Tabla 5.4 Análisis de la respuesta temporal de $\dot{z}_B$.

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	0.8	4	12
Erle Copter	0.5	7	10

Por otra parte, el incremento de altitud del modelo en SIMULINK® comienza siendo muy similar al del modelo en Gazebo, pero a medida que crece el tiempo y la velocidad de ascenso aumenta, la diferencia en posición se hace cada vez mayor debido en parte a la acumulación de errores. Prueba de ello es la Figura 5.7, donde puede verse un error inicial nulo que va creciendo hasta alcanzar prácticamente los $2m$, justo cuando la velocidad deja de actuar. En cambio, las mayores diferencias en velocidad y aceleración vertical tienen lugar durante el transitorio, es decir, cuando se produce un cambio en las velocidades de ascenso y descenso. Estas desaparecerán al estabilizarse la velocidad en un determinado valor, haciéndose nula la aceleración.

En la Figura 5.8, se representa la variación de empuje que experimentan los cuatro rotores debido a cambios en la velocidad de ascenso y descenso del vehículo. Obviamente, todos los rotores actúan al unísono aumentando su velocidad de rotación hasta alcanzar la velocidad vertical deseada, relajándose posteriormente una vez que el régimen permanente ha sido establecido. Como se deduce de la imagen anterior, sus valores de sustentación se encuentran dentro de los límites razonables e incluso para la velocidad máxima de ascenso, sigue dándose el mismo resultado. Es decir, en ningún caso se producirá la saturación de los mismos con la consiguiente pérdida de control sobre el quadrotor.

Por último, señalar que la respuesta del sistema quadrotor más modo Alt-Hold es simétrica, es decir, se obtendrían los mismos resultados considerando velocidades de descenso o negativas. Por tanto, daría lo mismo subir o bajar y cambiaría únicamente el signo de las distintas variables presentes.

Los controladores desarrollados en este capítulo son extensivos al vehículo en modo vuelo, y no consideran

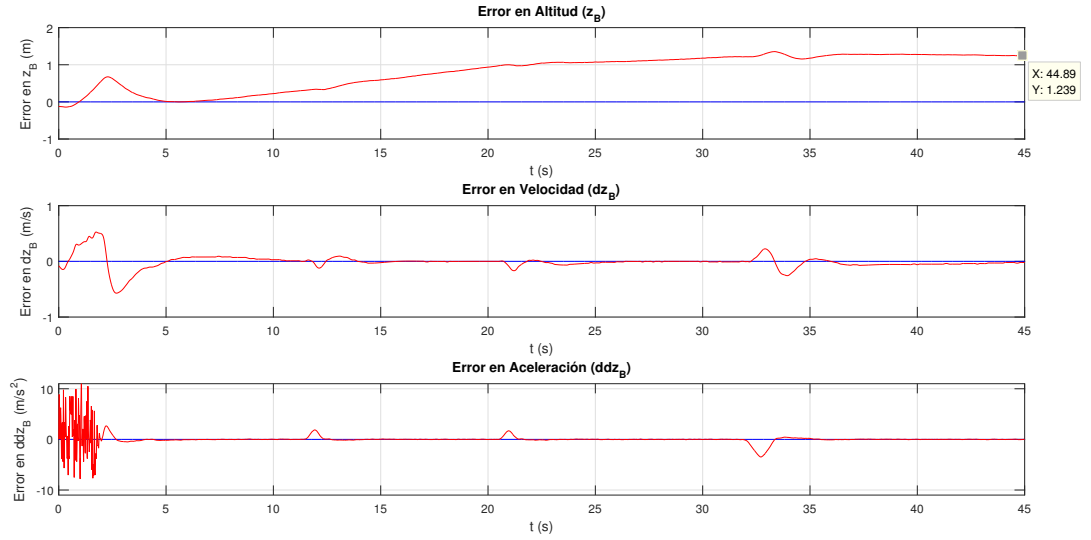


Figura 5.7 Error en altitud, velocidad y aceleración.

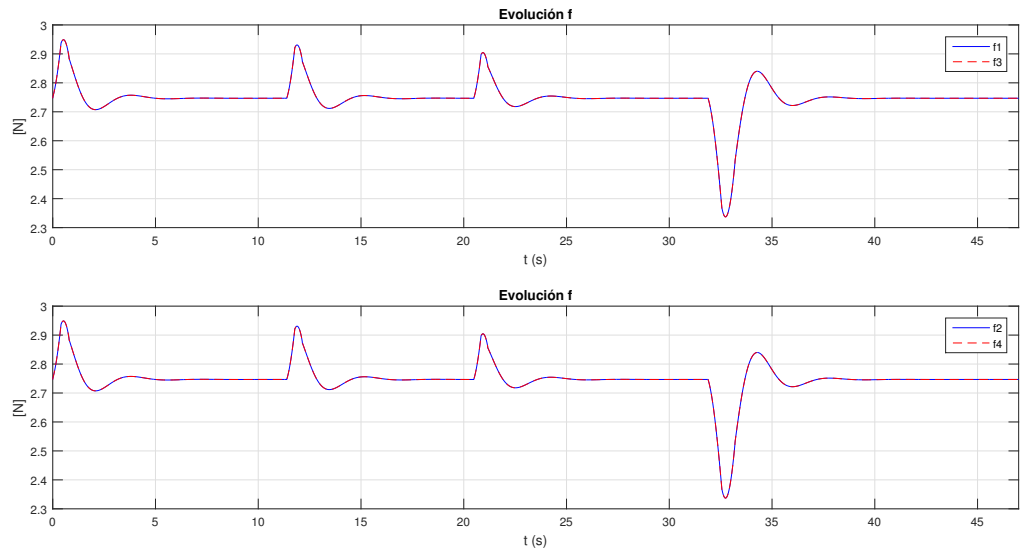


Figura 5.8 Variación de Empuje de los Rotores ante cambios en $\dot{z}_B$.

la puesta en funcionamiento del modo Alt-Hold o el caso de aterrizaje. Estas situaciones son más complejas de modelar y se dejan para futuros proyectos. De hecho, en la Figura 5.6 se muestra lo costoso del arranque y las múltiples oscilaciones experimentadas por el vehículo en Gazebo hasta alcanzar la velocidad deseada en pleno vuelo.

6 Modo Loiter

El modo Loiter o modo revoloteo pretende establecer la posición horizontal del quadrotor controlando directamente su velocidad en dirección horizontal. Para poder llevar a cabo esta tarea hace uso de los controladores desarrollados para el modo Stabilize en el Capítulo 4, considerando velocidad de prealimentación nula, y de los controladores presentados para el modo Alt-Hold en el Capítulo 5. De esta forma, se busca conseguir la posición deseada mediante una serie de controladores PID anidados que incluyen los del modo Stabilize.

En este modo, el usuario controla la dirección “x” del vehículo a través del mando encargado de controlar el ángulo de cabeceo y la dirección “y” del mismo mediante el mando de alabeo. El control de la guiñada y de la altitud se realizan de forma idéntica al modo Alt-Hold, es decir, usando los respectivos mandos de guiñada y de throttle. Para obtener una información más completa sobre este modo de vuelo, el lector es referido al enlace [15].

El empleo del modo Loiter requiere la adaptación de los algoritmos de control desarrollados por ArduPilot al modelo en SIMULINK[®] que se viene empleando. Estos se basan en la versión *Copter 3.3.1 26-Oct-2015* para Gazebo. Una de las características de este modo es que las señales de actuación generadas a la salida del modo Loiter, alimentan a su vez las entradas del Stabilize. El bloque encargado de transformar las aceleraciones lineales en ángulos de referencia es el conocido como *accel_to_lean_angles*.

El esquema de control que rige la posición horizontal consta de una serie de controladores anidados que realizan las funciones propias de un controlador PID, aunque con algunas variaciones. El controlador *Rate_Loiter* calcula a partir del error en velocidad horizontal, la aceleración necesaria que será transformada en ángulo de alabeo o cabeceo. Por su parte, el controlador *Loiter* superior transforma la diferencia de posición en velocidad hacia el objetivo que alimenta a su vez al controlador *Rate_Loiter*. Finalmente, un integrador transforma la velocidad horizontal deseada en posición objetivo, manteniendo el quadrotor su posición actual si la velocidad de entrada es nula. El esquema de control aplicado en SIMULINK[®] puede verse en la Figura 6.1.

6.1 Esquema de Control para Loiter

Dado que este esquema de control será idéntico tanto para la dirección “x” como para la dirección “y”, se formula el problema de control en ambas direcciones a la vez. El primer bloque, *desired_vel_to_pos*, emplea la velocidad deseada para mover el quadrotor hacia la posición objetivo. Por tanto, las entradas al sistema vendrán determinadas por las velocidades deseadas en dirección “x” e “y”, $vel_desired.x_k$ y $vel_desired.y_k$.

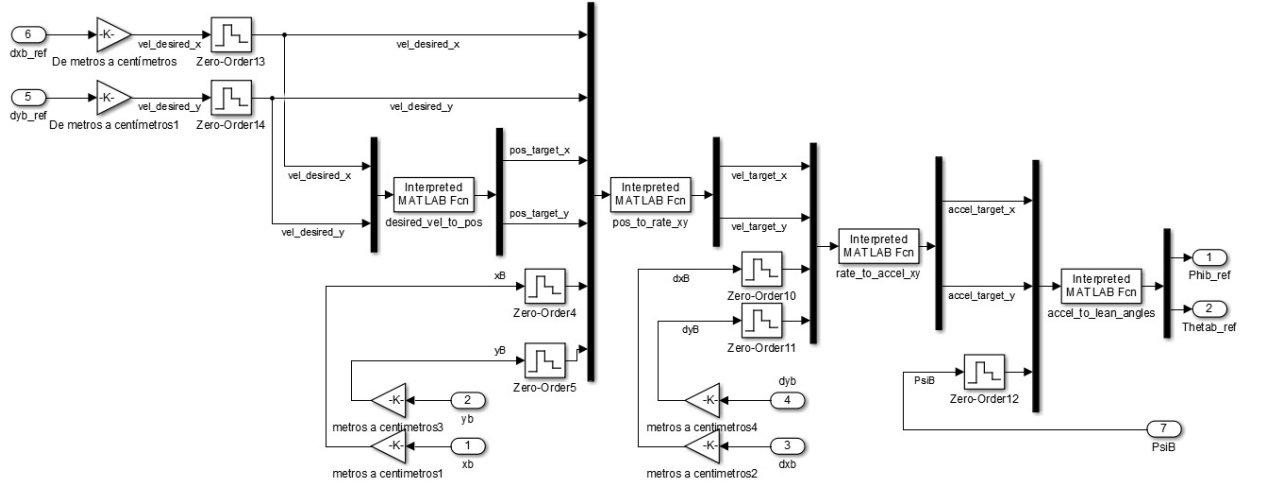


Figura 6.1 Esquema para el control de la posición en SIMULINK®.

Antes de comenzar con el cálculo de la posición objetivo, se obtienen las restricciones en distancia horizontal o error que serán empleadas en los sucesivos bloques.

$$\begin{aligned} speed_cms &= WPNAV_LOIT_SPEED \\ accel_cms &= WPNAV_LOIT_MAXA \end{aligned} \quad (6.1)$$

Los valores de la velocidad horizontal máxima ($speed_cms$), de la aceleración límite ($accel_cms$) y de la ganancia proporcional se recogen en la Tabla 6.1 y generan la restricción en distancia impuesta para la dirección horizontal. La función $calc_leash_length$ es idéntica a la ya presentada para el error en altitud y su código puede consultarse en 5.3.

$$\begin{aligned} & \text{if } flags_recalc_leash.xy == 1 \\ & \quad leash = calc_leash_length(speed_cms, accel_cms, POS_XY_P) \\ & \quad flags_recalc_leash.xy = 0 \\ & \text{end} \end{aligned} \quad (6.2)$$

Obtenida la distancia límite, se actualiza la posición objetivo a partir de su posición inicial y se integra en el tiempo. La velocidad deseada será la que establezca el cambio de posición y se discretiza con periodo $T = 0.01$ para poder generar una señal compatible con el sistema. El parámetro $flags_reset_desired_vel_to_pos$ es el encargado de activar el cálculo del ángulo objetivo al comienzo de la simulación.

$$\begin{aligned} & \text{if } flags_reset_desired_vel_to_pos == 1 \\ & \quad pos_target.x_k = x_{B0} * 100 \\ & \quad pos_target.y_k = y_{B0} * 100 \\ & \quad flags_reset_desired_vel_to_pos = 0 \\ & \text{else} \\ & \quad pos_target.x_k = pos_target.x_{k-1} + vel_desired.x_k * T \\ & \quad pos_target.y_k = pos_target.y_{k-1} + vel_desired.y_k * T \\ & \text{end} \end{aligned} \quad (6.3)$$

El control de la posición horizontal del vehículo se realiza en el siguiente bloque, llamado *pos_to_rate_xy*. En él, se transforma el error en posición horizontal a una velocidad objetivo que deberá ser seguida por el quadrotor. En primer lugar, se calcula el error en distancia tanto para la dirección “x” como para la dirección “y”.

$$\begin{aligned} pos_error.x_k &= pos_target.x_k - x_{Bk} \\ pos_error.y_k &= pos_target.y_k - y_{Bk} \end{aligned} \quad (6.4)$$

Donde las posiciones x_{Bk} e y_{Bk} , hacen referencia a la posición actual del quadrotor y serán expresadas tanto en centímetros como en tiempo discreto. A continuación, se calcula la distancia total hacia el objetivo y se compara con la restricción horizontal obtenida en la expresión 6.2.

$$distance_to_target_k = \sqrt{(pos_error.x_k)^2 + (pos_error.y_k)^2} \quad (6.5)$$

Si esta distancia es mayor que la restricción horizontal impuesta, entonces se limita la posición objetivo para tener una distancia razonable a la posición actual del vehículo y, en consecuencia, se recalcula el error en distancia para lo nuevos valores.

$$\begin{aligned} & \text{if } distance_to_target_k > leash \ \& \ distance_to_target_k > 0 \\ & \quad pos_target.x_k = x_{Bk} + leash * \frac{pos_error.x_k}{distance_to_target_k} \\ & \quad pos_target.y_k = y_{Bk} + leash * \frac{pos_error.y_k}{distance_to_target_k} \\ & \quad pos_error.x_k = pos_target.x_k - x_{Bk} \\ & \quad pos_error.y_k = pos_target.y_k - y_{Bk} \\ & \quad distance_to_target_k = leash \\ & \text{end} \end{aligned} \quad (6.6)$$

Tabla 6.1 Ganancias características del control en posición y valores máximos.

Parámetro	Valor	Observación
POS_XY_P	1	-
WPNAV_LOIT_SPEED	1000	cm/s
WPNAV_LOIT_MAXA	250	cm/s ²

Una vez actualizado el error, se busca la distancia lineal (*LD*) que establece la condición para el cálculo de la respuesta lineal o cuadrática según corresponda. Para ello, se sigue un procedimiento similar al desarrollado para la obtención de la respuesta en el controlador *sqrt_controller*.

$$LD = \frac{accel_cms}{2 * (POS_XY_P)^2} \quad (6.7)$$

Si el error total es mayor que la distancia lineal, se procede entonces al cálculo de la respuesta cuadrática

como el producto de la aceleración por el exceso de distancia respecto al punto de transición. Si por el contrario se está cerca del objetivo, se obtiene la velocidad objetivo como el producto lineal de una ganancia por el error en posición.

$$\begin{aligned}
 & \text{if } distance_to_target_k > 2 * LD \\
 & \quad vel_sqrt_k = \sqrt{2 * accel_cms * (distance_to_target_k - LD)} \\
 & \quad vel_target.x_k = vel_sqrt_k * \frac{pos_error.x_k}{distance_to_target_k} \\
 & \quad vel_target.y_k = vel_sqrt_k * \frac{pos_error.y_k}{distance_to_target_k} \\
 & \text{else} \\
 & \quad vel_target.x_k = POS_XY_P * pos_error.x_k \\
 & \quad vel_target.y_k = POS_XY_P * pos_error.y_k \\
 & \text{end}
 \end{aligned} \tag{6.8}$$

Para finalizar con el cálculo de la velocidad objetivo se escala la velocidad objetivo dentro de los valores máximos admisibles. Otra opción posible, sería actualizar la velocidad objetivo añadiéndole la velocidad prealimentada utilizada en la búsqueda de la posición objetivo, $vel_desired.x_k$ y $vel_desired.y_k$ respectivamente. La adición de esta variable daba lugar a una velocidad horizontal más distante de la generada por el modelo de quadrotor en Gazebo, por lo que se desechó considerar la velocidad prealimentada en este caso.

La velocidad objetivo total vuelve a ser la raíz de la suma de los cuadrados de la velocidad en cada dirección.

$$vel_total_k = \sqrt{(vel_target.x_k)^2 + (vel_target.y_k)^2} \tag{6.9}$$

Y si la velocidad resultante es mayor que la máxima velocidad permitida, entonces la velocidad en cada dirección se calcula restringiendo su valor al intervalo establecido por la máxima velocidad.

$$\begin{aligned}
 & \text{if } vel_total_k > speed_cms \\
 & \quad vel_target.x_k = speed_cms * \frac{vel_target.x_k}{vel_total_k} \\
 & \quad vel_target.y_k = speed_cms * \frac{vel_target.y_k}{vel_total_k} \\
 & \text{end}
 \end{aligned} \tag{6.10}$$

Conocidas las velocidades objetivo en dirección horizontal, la aceleración correspondiente se obtiene tras ejecutar el lazo interno conocido como $rate_to_accel_xy$. Este bloque realiza la misma función que un controlador PI, a cuya salida se le suma una aceleración prealimentada que ha sido calculada previamente. Su resultado se enviará a un bloque inferior que adapta las aceleraciones producidas en ambos ejes, a ángulos de inclinación compatibles con los controladores del modo Stabilize.

Antes de comenzar con el cálculo propio de la aceleración objetivo, se actualizan las variables internas que serán empleadas en la obtención de esta aceleración. Estas variables serán la velocidades últimas, donde se almacenan las velocidades objetivo correspondientes al instante de tiempo que se esté ejecutando y que serán usadas en la próxima llamada a la función. El parámetro $flags_reset_rate_to_accel.xy$ es el encargado de

Tabla 6.2 Ganancias características del controlador VEL_XY.

Parámetro	Valor	Observación
VEL_XY_P	1	Kp
VEL_XY_I	0.5	Ki
VEL_XY_IMAX	1000	IMAX
VEL_XY_FILT_HZ (Hz)	5	FILT

iniciar dicha asignación inicial, por lo que valdrá 1 nada más comenzar a ejecutar el programa.

$$\begin{aligned}
 & \text{if } flags_reset_rate_to_accel.xy == 1 \\
 & \quad vel_last.x_k = vel_target.x_k \\
 & \quad vel_last.y_k = vel_target.y_k \\
 & \quad flags_reset_rate_to_accel.xy = 0 \\
 & \text{end}
 \end{aligned} \tag{6.11}$$

Seguidamente se actualiza la aceleración prealimentada a partir de la diferencia de velocidad existente entre la velocidad objetivo actual y la del instante anterior. Esta aceleración se sumará a la salida del controlador PI y valdrá cero nada más comenzar el proceso.

$$\begin{aligned}
 accel_feedforward.x_k &= \frac{(vel_target.x_k - vel_last.x_k)}{T} \\
 accel_feedforward.y_k &= \frac{(vel_target.y_k - vel_last.y_k)}{T}
 \end{aligned} \tag{6.12}$$

Calculada la aceleración deseada, se almacena en la velocidad última, la velocidad actual para la próxima iteración.

$$\begin{aligned}
 vel_last.x_{k+1} &= vel_target.x_k \\
 vel_last.y_{k+1} &= vel_target.y_k
 \end{aligned} \tag{6.13}$$

Y se procede a calcular el error en velocidad horizontal para obtener así la entrada al controlador PI. Esta entrada se filtra nada más introducirse en el sistema para minimizar el efecto de los posibles errores en la señal de entrada.

$$\begin{aligned}
 vel_error.x_k &= vel_target.x_k - \dot{x}_{Bk} \\
 vel_error.y_k &= vel_target.y_k - \dot{y}_{Bk}
 \end{aligned} \tag{6.14}$$

Los valores del filtro a la entrada y los de las ganancias proporcionales del controlador pueden ser consultados en Tabla 6.2. Además, serán los mismos para cada dirección horizontal dada la simetría del quadrotor, lo que se traduce en una dinámica muy similar para ambos ejes perpendiculares.

$$filt_alpha = \frac{(2 * \pi * FILT) * T}{(2 * \pi * FILT) * T + 1} \tag{6.15}$$

$$\begin{aligned} input_filt_change.x_k &= (vel_error.x_k - input.x_{k-1}) * filt_alpha \\ input.x_k &= input.x_{k-1} + input_filt_change.x_k \end{aligned} \quad (6.16)$$

$$\begin{aligned} input_filt_change.y_k &= (vel_error.y_k - input.y_{k-1}) * filt_alpha \\ input.y_k &= input.y_{k-1} + input_filt_change.y_k \end{aligned} \quad (6.17)$$

La parte proporcional es el resultado de multiplicar la entrada al controlador por la ganancia proporcional correspondiente ($Kp = VEL_XY_P$).

$$\begin{aligned} vel_p.x_k &= input.x_k * Kp \\ vel_p.y_k &= input.y_k * Kp \end{aligned} \quad (6.18)$$

La parte integral requiere la integración del error en velocidad y su multiplicación posterior por la ganancia integral ($Ki = VEL_XY_I$). Su resultado será posteriormente truncado por los valores de $\pm IMAX$, evitando así aceleraciones muy elevadas a la salida del controlador.

$$\begin{aligned} vel_i.x_k &= vel_i.x_{k-1} + (input.x_k * Ki) * T \\ vel_i.y_k &= vel_i.y_{k-1} + (input.y_k * Ki) * T \end{aligned} \quad (6.19)$$

Para terminar con este bloque y antes de transformar las aceleraciones generadas, se combinan la aceleración prealimentada calculada en la expresión 6.12 con la aceleración obtenida a partir del error en velocidad horizontal.

$$\begin{aligned} accel_target.x_k &= accel_feedforward.x_k + vel_p.x_k + vel_i.x_k \\ accel_target.y_k &= accel_feedforward.y_k + vel_p.y_k + vel_i.y_k \end{aligned} \quad (6.20)$$

El bloque *accel_to_lean_angles*, es el encargado de transformar las aceleraciones horizontales en ángulos de inclinación en los ejes de alabeo y cabeceo. La máxima aceleración horizontal por defecto a la salida del controlador de velocidad, viene dada por el parámetro interno $accel_max = POSCONTROL_ACCEL_XY_MAX = 980cm/s^2$. Su valor podría ser limitado si la altitud del vehículo se encontrara comprometida, dando prioridad al establecimiento de la misma sobre la posición.

$$accel_total_k = \sqrt{(accel_target.x_k)^2 + (accel_target.y_k)^2} \quad (6.21)$$

Por tanto, se busca que la aceleración total a la salida no sea mayor que la aceleración fijada como máxima. Si se supera esta valor, entonces, se escala la aceleración resultante dentro de los valores admisibles de

aceleración.

$$\begin{aligned}
 & \text{if } accel_total_k > accel_max \\
 & \quad accel_target.x_k = accel_max * \frac{accel_target.x_k}{accel_total_k} \\
 & \quad accel_target.y_k = accel_max * \frac{accel_target.y_k}{accel_total_k} \\
 & \text{end}
 \end{aligned} \tag{6.22}$$

A la aceleración objetivo calculada se le puede aplicar un límite de sacudida, que tendrá un efecto importante en los instantes iniciales de cambio de velocidad, cuando se alcanzan los valores más elevados.

A continuación, se rotan las aceleraciones obtenidas para situarlas en los ejes correspondientes del vehículo, evitando así problemas con el ángulo de guiñada cuando su valor sea distinto de cero. El sentido tomado como positivo para el ángulo de inclinación difiere en ambos modelos, por lo que habrá que cambiar el signo de la aceleración lineal al aplicar el control al modelo de quadrotor establecido en MATLAB®.

$$\begin{aligned}
 accel_forward_k &= accel_target.x_k * \cos(\psi_B) + accel_target.y_k * \sin(\psi_B) \\
 accel_right_k &= -accel_target.x_k * \sin(\psi_B) + accel_target.y_k * \cos(\psi_B)
 \end{aligned} \tag{6.23}$$

Finalmente, se actualizan los ángulos deseados que serán enviados a los controladores del modo Stabilize y que se emplean como entradas al sistema. Estos ángulos de inclinación no deberán de ser superiores a los máximos permitidos en el modo Stabilize y serán restringidos en caso de superar dicho valor. Además, sus valores serán pasados a centígrados en correspondencia con las unidades con las que trabaja el controlador del lazo inferior.

$$pitch_target_k = \arctan\left(\frac{accel_forward_k}{g * 100}\right) * \frac{18000}{\pi} \tag{6.24}$$

$$-ANGLE_MAX \leq pitch_target_k \leq ANGLE_MAX$$

$$roll_target_k = \arctan\left(\frac{-accel_right_k * \cos\left(pitch_target_k * \frac{\pi}{18000}\right)}{g * 100}\right) * \frac{18000}{\pi} \tag{6.25}$$

$$-ANGLE_MAX \leq roll_target_k \leq ANGLE_MAX$$

Una vez descritas las ecuaciones que rigen el problema de control en dirección horizontal se procede a estudiar el comportamiento del sistema. Para ello, se han realizado distintas pruebas en los diferentes escalones de control del modo Loiter, tanto en velocidad como en el bloque superior que integra además de la velocidad el control de la posición horizontal. Con el fin de poder comparar ambas direcciones horizontales, se analizan por separado y se mantienen a cero las entradas de la dirección contraria a la de estudio.

6.2 Análisis de la Respuesta en Loiter

6.2.1 Control de “ x_B ”

Como ya se ha visto, la aceleración horizontal a la salida del controlador PI en velocidad, es transformada en un ángulo de cabeceo que alimenta a los sucesivos controladores del modo Stabilize. Para ver el comportamiento tanto del bloque que controla la velocidad en “x” del vehículo como del que realiza el cambio a ángulo de cabeceo, se corta al sistema en este punto y se introduce una velocidad escalón de 1 m/s . La respuesta del sistema ante entrada escalón en velocidad puede verse en la figura Figura 6.2, donde se ha representado no solo la velocidad horizontal del vehículo sino también la evolución de la posición en “x” y la actitud afectada por el respectivo cambio en velocidad.

De esta imagen, se observa como la velocidad del vehículo trata de alcanzar rápidamente la referencia aunque siempre oscilando en torno a ella. El tiempo de subida es apenas de 1.2 s aunque la sobreoscilación se encuentra alrededor del 29% . De igual forma, el ángulo de cabeceo oscila cambiando su valor hasta que la velocidad deseada es alcanzada. Esto ocurre dado que no se ha considerado resistencia alguna al movimiento del vehículo y una vez alcanzada la velocidad, esta puede mantenerse en el tiempo sin necesidad de fijar el ángulo de cabeceo constante.

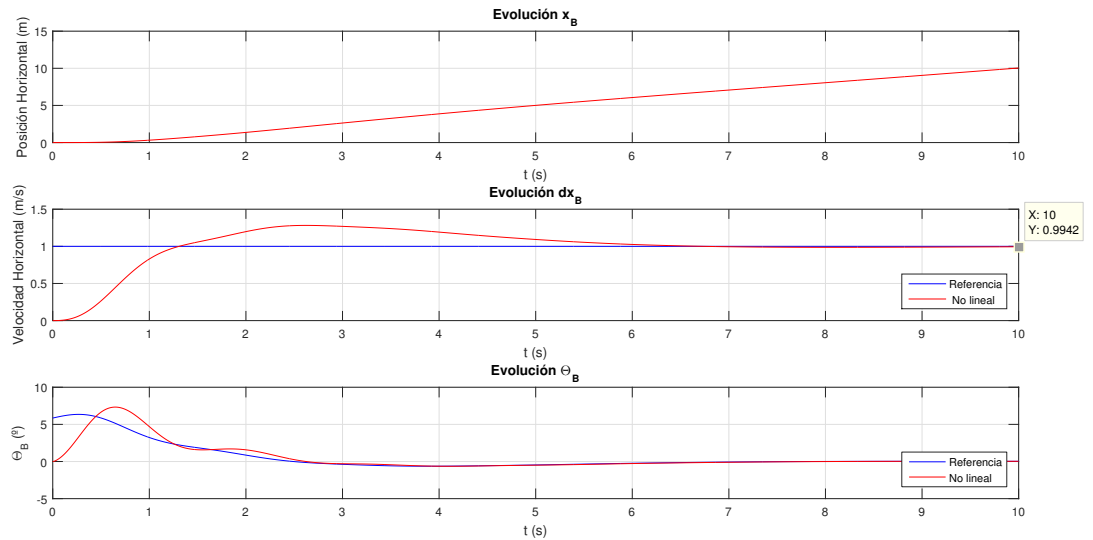


Figura 6.2 Respuesta de $\dot{x}_B$ ante entrada escalón de 1 m/s .

Tanto la posición en “y” como la actitud respectiva no se ven afectadas por cambios en $\dot{x}_B$ y, por tanto, no se han representado. Este desacoplamiento entre ambas direcciones es solo válido cuando el ángulo de guiñada es nulo. Si por el contrario, el ángulo de guiñada estuviera fijado a otro valor, existiría una pequeña perturbación en dirección “y” que rápidamente sería compensada por el controlador en dicha dirección. También se produce una ligera oscilación de la altitud que será estabilizada por los controladores del modo Alt-Hold y que será mayor cuanto más grande sea la velocidad de referencia requerida.

Para estudiar el comportamiento del controlador en posición horizontal junto con el resto de controladores que completa el control en dirección “x”, se introduce una entrada escalón en posición horizontal de 1 m . Al igual que para la velocidad, se han representado la posición, la velocidad y la actitud afectada en la Figura 6.3. La posición deseada es alcanzada en un corto intervalo de tiempo ($\sim 2.5\text{ s}$) y la oscilación es menor que en el caso de la velocidad (14%). De igual forma, la velocidad aumenta hasta conseguir la posición objetivo para después reducirse hasta hacerse nula y lo mismo ocurre con el ángulo objetivo. La oscilación que sufre este

último es ahora mayor, pues al efecto de la velocidad, habría que sumarle la perturbación en posición.

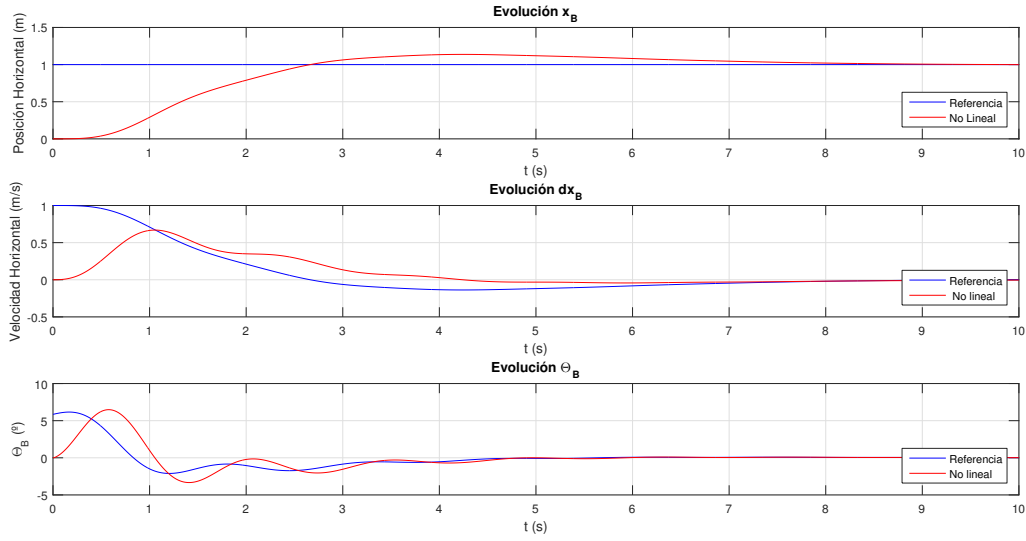


Figura 6.3 Respuesta de x_B ante entrada escalón de 1m.

De idéntica forma, el ángulo de guiñada introduce un efecto de acoplamiento entre ambas direcciones que tiende a desaparecer al actuar el sistema de control. La altitud también cae debido al cambio de posición horizontal, pero su valor es muy pequeño y será rápidamente compensado por el modo Alt-Hold.

Finalmente, se compara el modelo de quadrotor desarrollado en SIMULINK® con el modelo en Gazebo de ErleCopter. Para ello, se realiza una primera prueba en Gazebo consistente en introducir distintos valores a través de la emisora que controla el pitch, la cual es empleada en modo Loiter para controlar la posición horizontal del vehículo en dirección “x”. Los valores introducidos y sus correspondientes respuestas en forma de velocidad deseada se pueden visualizar en la Tabla 6.3.

Tabla 6.3 Valores PWM en Gazebo y Velocidad $\dot{x}_B$ Generada.

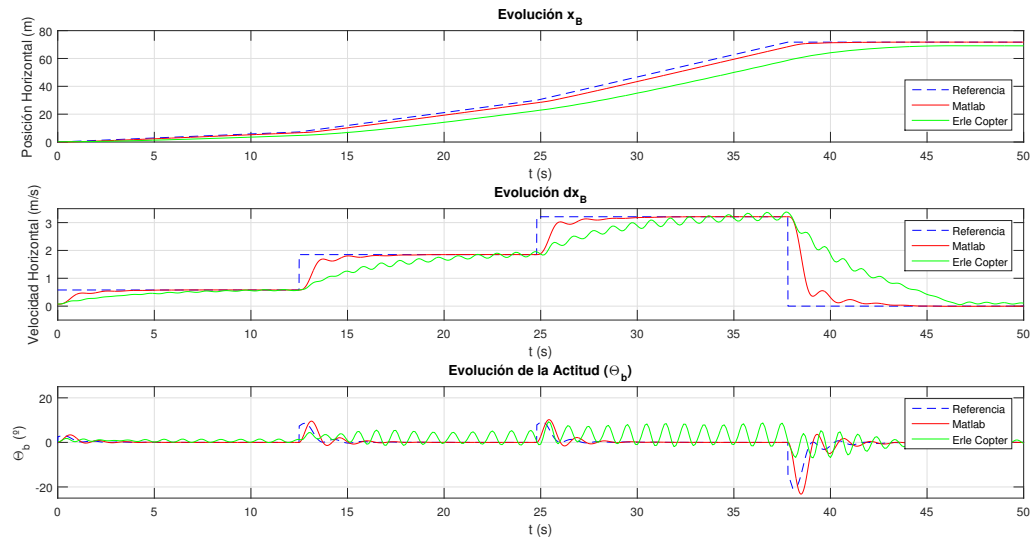
Canal (rc)	PWM	$vel_desired.x$ (m/s)	Tiempo (s)
2	1550	0.58	0
2	1600	1.85	12.5
2	1650	3.21	24.8
2	1500	0	37.8

Obtenidos los datos del experimento realizado en Gazebo, se ejecuta el esquema de control del modo Loiter junto con las ecuaciones dinámicas desarrolladas en el Capítulo 2. La prueba realizada en SIMULINK® para la velocidad horizontal en “x” genera las entradas escalón recogidas en la tabla anterior y mantiene el resto de entradas al sistema nulas. Es decir, la velocidad horizontal en “y” junto con la altitud y la guiñada deberían estar ancladas al valor inicial y recuperar rápidamente su posición inicial si una perturbación tiene lugar.

Tabla 6.4 Análisis de la respuesta temporal de $\dot{x}_B$.

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	3.6	5	0
Erle Copter	8.5	12	4

En la Figura 6.4, se representa el comportamiento de la velocidad horizontal del drone para los dos modelos de estudio. En ella, se observa una clara diferencia entre ambos modelos, siendo el modelo desarrollado en MATLAB® mucho más rápido que el modelo de ErleCopter. La oscilación está también presente en ambos modelos, pero la del modelo en MATLAB® se termina amortiguando con el paso del tiempo. En la Tabla 6.4, se recogen algunos de estos datos característicos como son el tiempo de establecimiento, subida o la oscilación de la respuesta.

**Figura 6.4** Respuesta de $\dot{x}_B$ ante varias entradas escalón.

Paralelamente al aumento de velocidad, tiene lugar un incremento de la posición en dirección “x”, siendo la distancia recorrida mayor en el caso de MATLAB®, pues se alcanzan las velocidades de referencia en menos tiempo. Lo mismo sucede con el ángulo de inclinación que oscila hasta conseguir la velocidad deseada. Además, tanto la altitud como la dirección “y” del vehículo se verán afectadas por cambios en la posición “x” del mismo. Con el fin de evitar una pérdida brusca de altitud o una variación no deseada en la posición, los controladores del modo Stabilize, Alt-Hold y Loiter actuarán rápidamente para compensar dichos cambios.

Como ya se ha comentado, los errores en velocidad y posición horizontal son grandes entre un modelo y otro. Ahora, a los errores que se tenían en el control del ángulo, se le suman los errores correspondientes al control de la velocidad y posición. Estos alcanzan sus valores máximos durante la fase del transitorio, llegando a tener durante la parada un error en velocidad de aproximadamente $2m/s$.

Los rotores más afectados por el cambio de velocidad en dirección “x” son los mismos que se tenían para el ángulo de cabeceo. Serán los rotores “1” y “3” los que verán alterados su empuje para alcanzar la velocidad horizontal deseada. La oscilación es también mayor debido al acoplamiento dinámico de los controladores de posición y ángulo, aunque afortunadamente todos sus valores se encontrarán dentro de los límites razonables establecidos para el empuje. Incluso si la velocidad máxima es demandada, los rotores no se encontrarán saturados en ningún momento. Los rotores “2” y “4” también se ven afectados pero su perturbación es debida al efecto generado por la caída de altitud o cambio en la posición “y”.

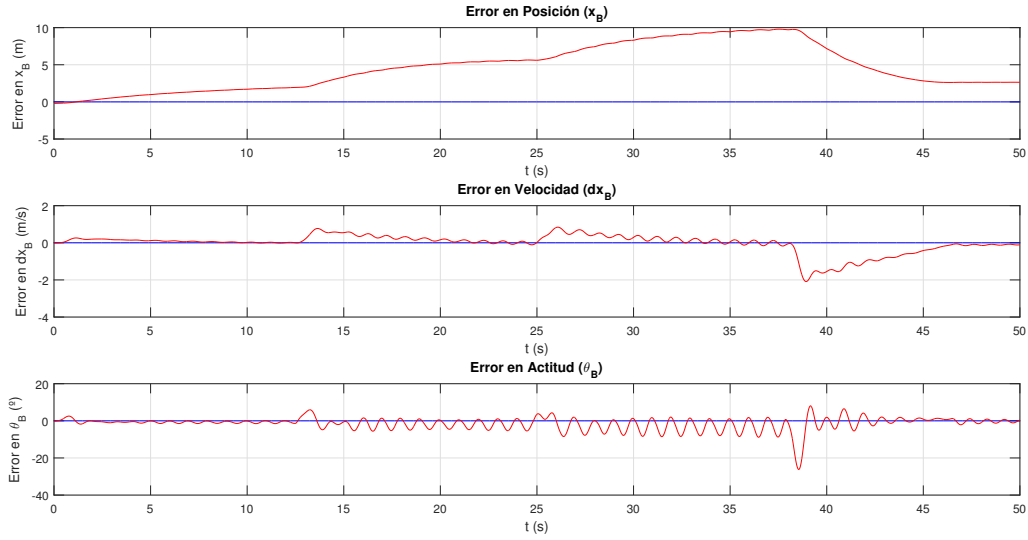


Figura 6.5 Error en posición, velocidad y actitud en dirección “x”.

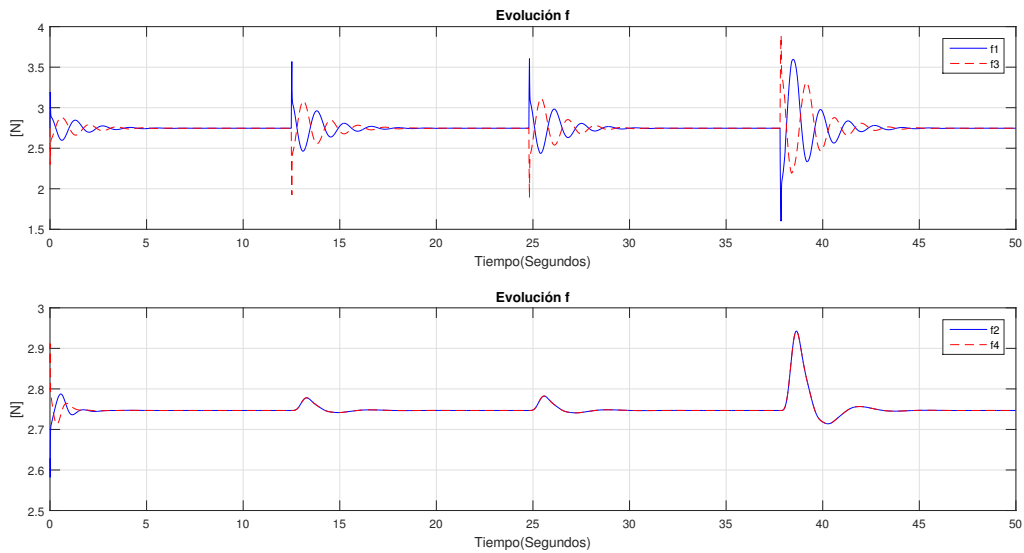


Figura 6.6 Variación de Empuje de los Rotores ante cambios en $\dot{x}_B$.

La gran diferencia existente entre el modelo en MATLAB® y el modelo de ErleCopter se debe a que este último constituye una aproximación más cercana a la realidad, pues diversas hipótesis simplificadoras habían sido tomadas en la derivación de las ecuaciones del movimiento del Capítulo 2. Otras fuentes de error, pueden ser la no inclusión de modelos matemáticos para la IMUs, el Optical Flow o diversos sensores que sí se han tenido en cuenta en el modelo en Gazebo. De hecho, la velocidad y posición horizontales del drone son calculadas a través del Optical Flow e introducen un cierto ruido que no ha sido modelado en MATLAB® y que puede ser un factor diferencia entre ambos modelos.

6.2.2 Control de “ y_B ”

Dado que las ganancias empleadas para el control de la dirección “y” son las mismas que las ya usadas para la dirección “x”, es de esperar que el comportamiento en esta dirección sea similar al estudiado en el apartado anterior. Ahora, la aceleración objetivo a la salida del controlador PI alimenta a la entrada de alabeo

del Stabilize y se tiene así simetría en ambas direcciones al igual que sucedía con los ángulos de cabeceo y alabeo.

De igual forma, para ver el comportamiento del controlador PI en velocidad se introduce una entrada escalón de 1m/s en dirección “y”, manteniendo a cero el resto de entradas. Los resultados se dibujan en la Figura 6.7 y se observa el mismo comportamiento para la velocidad y posición horizontal con tiempos de subida y oscilación muy semejantes. El ángulo de alabeo, sin embargo, tendrá un comportamiento inverso al de cabeceo, siendo negativo cuando la velocidad es positiva. Esto es consecuencia de las relaciones cinemáticas establecidas para el vehículo y del criterio de signos adoptado para su obtención, no teniendo mayor repercusión sobre el control.

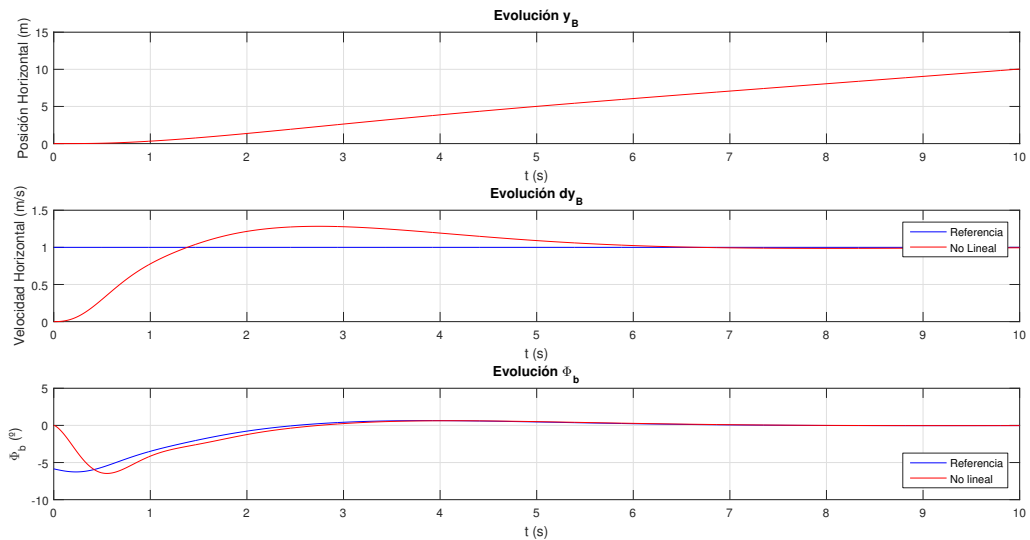


Figura 6.7 Respuesta de $\dot{y}_B$ ante entrada escalón de 1m/s .

El efecto de introducir una entrada de 1m en el sistema en dirección “y”, se representa en la Figura 6.8. El resultado es idéntico al de la dirección opuesta con la diferencia de que el ángulo de alabeo tiene ahora sentido contrario a la velocidad. Igualmente, se produce una ligera caída de la altitud y ambas direcciones se ven afectadas cuando el ángulo de guiñada es distinto de cero.

A la vista de los resultados, queda claro que es posible controlar la posición del vehículo utilizando solamente un controlador en velocidad y otro superior en posición. Sin embargo, se quiere ir más allá y tener un control directo en velocidad. Esta tarea se lleva a cabo en el bloque *desired_vel_to_pos*, y cierra el control en posición.

Analizados los controladores en velocidad y posición por separado, se compara finalmente el modelo completo de quadrotor en MATLAB® con el modelo de ErleCopter para la velocidad horizontal en dirección “y”. El estudio en Gazebo se ha realizado introduciendo distintas señales a través de la emisora que controla el ángulo roll, manteniendo constantes las del resto. Los PWM empleados se recogen en la Tabla 6.5 y generan velocidades negativas a diferencia de la dirección “x”. Esto tendrá que ver con el criterio de signos tomado por ArduPilot, significando que para valores de la emisora menores que 1500, las velocidades han de ser necesariamente positivas.

A partir de las velocidades deseadas recogidas en la tabla anterior, se efectúa la prueba en MATLAB® consistente en introducir varias velocidades escalón en dirección “y”, dejando nulas el resto de entradas al sistema (altitud, guiñada y velocidad en “x”).

Los resultados obtenidos para ambas pruebas se pueden visualizar en la Figura 6.9 y su comportamiento es muy similar al ya estudiados para la velocidad $\dot{x}_B$. El modelo de ErleCopter tarda más en alcanzar la

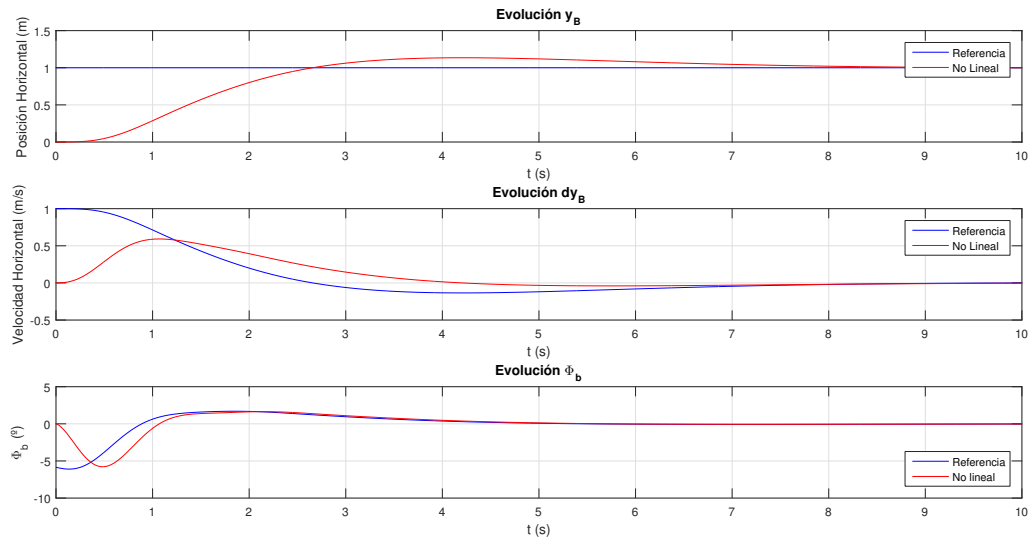


Figura 6.8 Respuesta de y_B ante entrada escalón de 1m.

Tabla 6.5 Valores PWM en Gazebo y Velocidad $\dot{y}_B$ Generada.

Canal (rc)	PWM	$vel_desired.y$ (m/s)	Tiempo (s)
1	1550	-0.5	0
1	1600	-1.77	9.8
1	1650	-2.71	21.3
1	1500	0	29.9

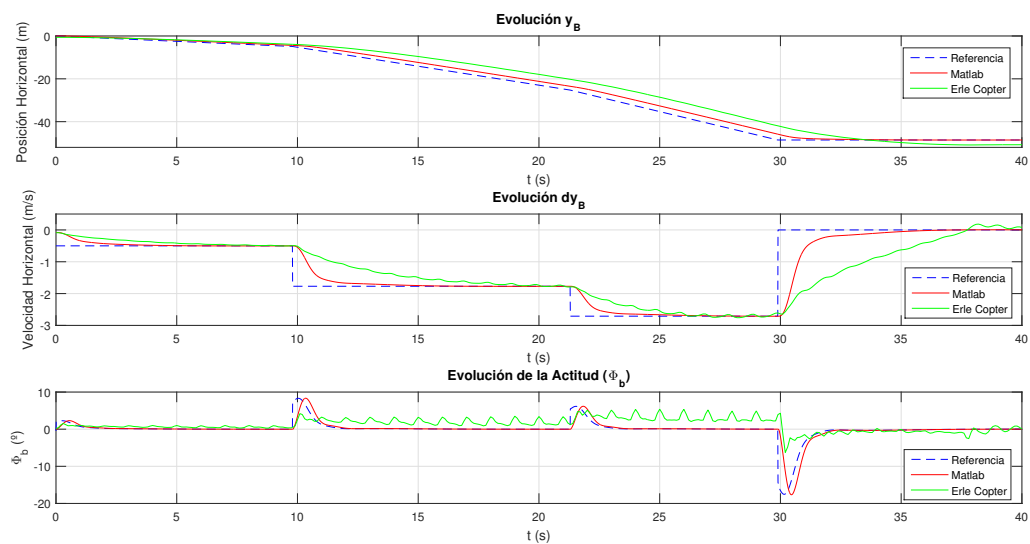


Figura 6.9 Respuesta de $\dot{y}_B$ ante varias entradas escalón.

velocidad objetivo y siempre oscilando en torno a ella. Lo mismo ocurre con la distancia horizontal, que al ser el modelo en MATLAB® más rápido en el tiempo, su curva quedará por debajo y alcanzará la posición final deseada mucho antes. El ángulo de alabeo también difiere en ambos modelos y es consecuencia de no considerar resistencia alguna que se oponga al movimiento del vehículo en MATLAB®. Por tanto, una vez alcanzada la velocidad deseada, el ángulo se vuelve nulo al no tener que compensar ninguna otra fuerza

horizontal que se oponga a su movimiento en dicha dirección.

Tabla 6.6 Análisis de la respuesta temporal de $\dot{y}_B$.

Modelo	RiseTime (s)	SettlingTime (s)	Overshoot (%)
Matlab	4	5	0
Erle Copter	7	8	1

Los errores en posición, velocidad y ángulo se han representado también en la Figura 6.10, dando lugar a errores similares en ambas direcciones. La mayor diferencia en distancia para ambos modelos es del orden de $5m$ y es suma de los sucesivos errores en posición tras cada variación en velocidad. En cambio, las mayores diferencias en velocidad y ángulo se dan durante el transitorio, coincidiendo cuando la respuesta del vehículo difiere más en uno y otro modelo.

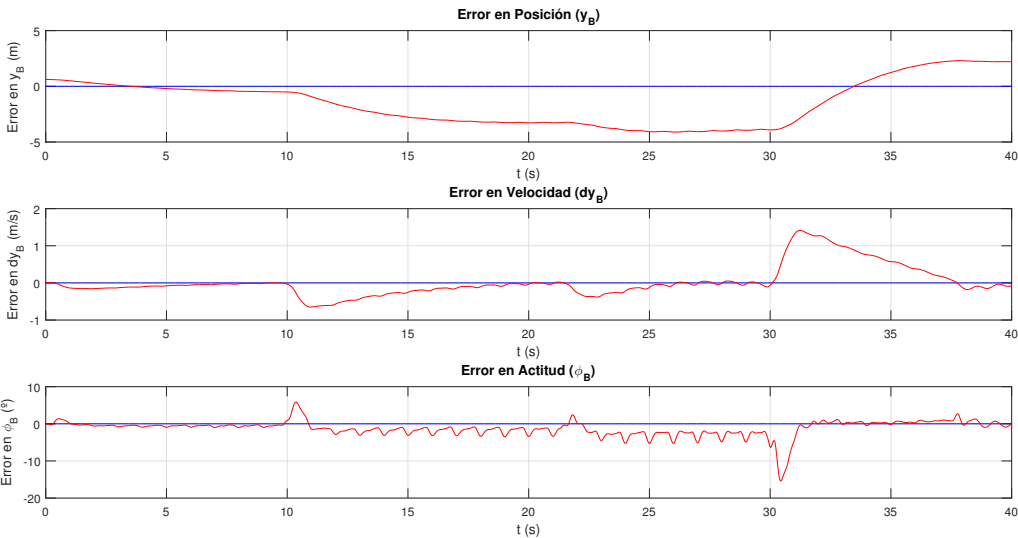


Figura 6.10 Error en posición, velocidad y actitud en dirección “y”.

Finalmente, se presentan también las fuerzas experimentadas por los rotores. En este caso, los rotores más afectados son los correspondientes al ángulo de alabeo, “2” y “4”. Su comportamiento es similar a los estudiados para la dirección “x”, con bastante oscilación debido al acoplamiento dinámico pero con sus valores situados dentro de los límites razonables. En ningún caso, se encontrarán saturados para el rango de velocidades permitidas, lo cual presenta la ventaja de no perder control efectivo sobre el vehículo. Sus resultados se presentan en la Figura 6.11 y confirman lo expuesto hasta ahora.

Como ya se ha comentado, las diferencias existentes entre ambos modelos se encuentran no solo en la formulación dinámica de las ecuaciones del quadrotor, sino también en el modelado de sensores y otros instrumentos de medida, así como de los motores presentes en el modelo de ErleCopter en Gazebo y no aplicados al modelo desarrollado en MATLAB®.

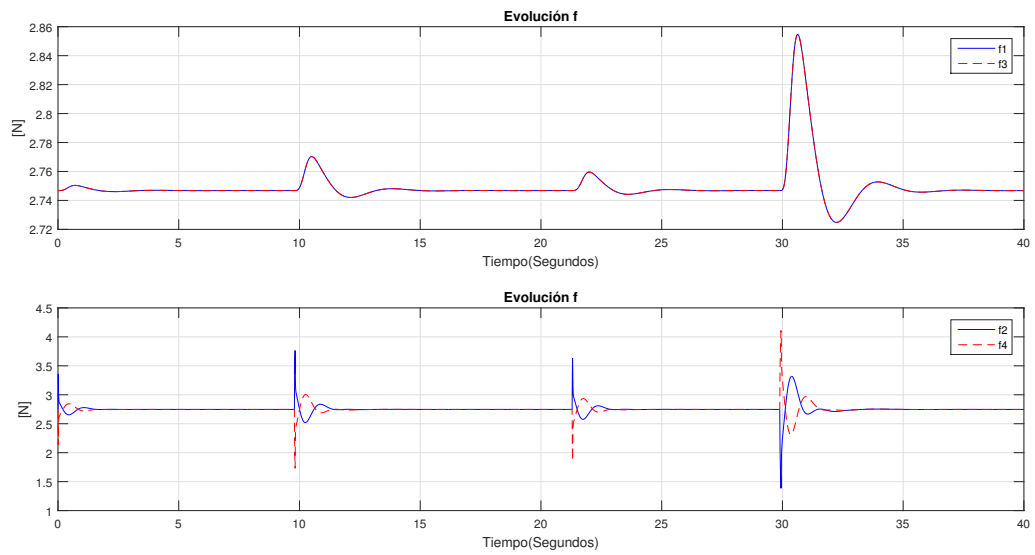


Figura 6.11 Variación de Empuje de los Rotores ante cambios en $\dot{\gamma}_B$.

7 Conclusión Final

La derivación del modelo no lineal formado por el vehículo más la carga mediante dos formulaciones distintas, reafirma la validez de las ecuaciones obtenidas. Supone disponer de un conjunto de ecuaciones simplificadas y realistas a las que poder aplicar diferentes técnicas de control en futuros proyectos. Su extensión a las coordenadas de la propia carga, permite tener un mejor control de la misma para misiones cercanas a la tierra, donde la altitud de la misma puede verse comprometida. De esta forma, se genera un modelo cercano a la realidad sobre el que poder aplicar nuevos algoritmos de control antes de ser aplicados al modelo real. Se evitan así posibles fallos que pudieran estrellar el vehículo o hacerlo caer.

Teniendo en mente la posible aplicación de las ecuaciones desarrolladas a futuros trabajos, se linealizaron sus expresiones tanto en coordenadas carga como en cuerpo. La ventaja de disponer de un modelo en velocidad nula y otro en velocidad constante se manifiesta en la posibilidad de tener un modelo alternativo para verificar la validez de los resultados obtenidos en el primero.

La adaptación de los esquemas de control desarrollados por ArduPilot al modelo de aeronave en MATLAB® dan lugar a soluciones óptimas para el control de la actitud. Los ángulos del quadrotor alcanzan el valor deseado en un corto periodo de tiempo y sin apenas oscilación. Esto ocasiona una respuesta inmediata del vehículo ante cambios en sus condiciones de operación, estando el empuje dentro de los valores admisibles y sin comprometer en ningún momento la actuación del vehículo. Por tanto, se puede decir que el empleo de controladores anidados en velocidad y en ángulo, genera un sistema bastante estable en actitud que podría ser aplicado al sistema formado por el quadrotor más la carga.

De igual forma, el comportamiento del vehículo ante variaciones en su altitud es bastante bueno. Evita por un lado una caída brusca de la misma cuando cambios en la actitud o posición son demandados y permite adquirir la velocidad de ascenso deseada de forma rápida y eficiente. Es decir, el conjunto de controladores PID en posición, velocidad y aceleración generan un control en altitud bastante robusto y eficaz, estabilizando al vehículo de forma que pueda operar en todo tipo de situaciones.

Y lo mismo ocurre para la posición horizontal, donde a pesar del acoplamiento dinámico entre los controladores de posición y actitud, la aeronave adquiere la velocidad horizontal deseada en muy poco tiempo aunque con algo más de oscilación. Solo en este caso, se encuentra una mejoría en el modelo en MATLAB® con respecto al de Gazebo. Así pues, el control primero de la velocidad seguido del control de la posición es un método muy robusto y permite tener un sistema estable en posición que recupere rápidamente el equilibrio en caso de perturbaciones externas.

Los experimentos realizados en los Capítulo 4, Capítulo 5 y Capítulo 6 para los modos Stabilize, Alt-Hold y Loiter ponen de manifiesto lo comentado hasta ahora. Se tiene un sistema que responde correctamente ante cambios en sus condiciones de vuelo, aunque las diferencias existentes entre el modelo matemático simulado en SIMULINK® y el modelo en Gazebo de ErleCopter son más que evidentes. Esta falta de similitud es mayor durante el transitorio y es debida fundamentalmente a las simplificaciones tomadas para derivar el modelo de Quadrotor desarrollado en el Capítulo 2. Este constituye una aproximación física de la realidad lo suficientemente buena para que la aplicación de un control robusto genere unos resultados más que aceptables.

Con el objetivo de tener un modelo en MATLAB[®] más similar al vehículo real, se podrían modelar otros efectos aerodinámicos que no se han tenido en cuenta como son la resistencia de pala, el momento de arrastre o el momento de balanceo de la hélice. Se podría haber tomado también un modelo de atmósfera distinto que tuviera en cuenta el viento o la existencia del mismo.

Otra de las diferencias existentes, con respecto al modelo en Gazebo, es la no inclusión de elementos externos como instrumentos o sensores de medida. Podría realizarse un modelado matemático de dispositivos como las IMU's o los barómetros e incluso un bloque que incluyera la dinámica de los motores. Sin embargo, la determinación de los ángulos y posición se realiza mediante la integración de las ecuaciones dinámicas.

Respecto al control, se han adaptado los esquemas correspondientes al vehículo en pleno vuelo, pudiéndose adaptar en un futuro los relacionados con el despegue o aterrizaje. Esta diferencia se reflejaba en el tiempo que tardaba en coger el quadrotor la velocidad de ascenso cuando se inicializa el modo Alt-Hold.

Obtenido el control del quadrotor, el objetivo más próximo sería la inclusión de la carga colgante cuya dinámica ha sido modelada en los primeros capítulos. Esta carga se podría incluir fácilmente en el modelo en SIMULINK[®], pues se disponen de las ecuaciones dinámicas del sistema formado por el quadrotor más la carga. La problemática que podría surgir, sería un empeoramiento en el comportamiento del sistema teniendo que ajustar las ganancias de los controladores para que se adaptaran mejor a las nuevas circunstancias. Otro procedimiento sería el desarrollo de controladores superiores para el control directo de la posición de la carga. Estos generarían, a partir del error en posición, la señal de actuación necesaria que modificaría la velocidad horizontal o vertical del quadrotor para seguir la trayectoria fijada. Se tendría así un controlador de altitud que alimentaría a su vez a los controladores del modo Alt-Hold y otros dos controladores de posición horizontal que estarían conectados con los del modo Loiter.

En resumen, el objetivo de controlar la posición y actitud del modelo dinámico de quadrotor desarrollado en este trabajo se ha cumplido satisfactoriamente y se plantan las bases para la adición y control posterior de una carga colgante. Añadida al modelo en MATLAB[®], se buscará incluirla también en el modelo real de quadrotor presente en el laboratorio.

Índice de Figuras

1.1	Drone con cámara incorporada para vigilancia tomada de [28]	1
1.2	Drone Amazon para transporte de carga tomado de [26]	2
2.1	Quadrotor con Carga Colgante tomado de [30]	6
4.1	Esquema para el control del Roll en SIMULINK®	50
4.2	Respuesta del <i>Sqrt_Controller</i> ante entrada rampa del error en alabeo	54
4.3	Respuesta de p ante entrada escalón de $1^\circ/s$	56
4.4	Respuesta de ϕ_B ante varias entradas escalón en ángulo	57
4.5	Error entre el modelo en Matlab y Gazebo para ϕ_B	58
4.6	Variación de Empuje de los Rotores ante cambios en ϕ_B	58
4.7	Esquema para el control del Pitch en SIMULINK®	59
4.8	Respuesta de q ante entrada escalón de $1^\circ/s$	62
4.9	Respuesta de θ_B ante varias entradas escalón	63
4.10	Error entre el modelo en Matlab y Gazebo para θ_B	64
4.11	Variación de Empuje de los Rotores ante cambios en θ_B	64
4.12	Esquema para el control del Yaw en SIMULINK®	65
4.13	Respuesta del <i>Sqrt_Controller</i> ante entrada rampa del error en guiñada	67
4.14	Respuesta de r ante entrada escalón de $1^\circ/s$	69
4.15	Respuesta de ψ_B ante entradas escalón en guiñada	70
4.16	Respuesta de ϕ_B y θ_B ante entradas escalón en guiñada	71
4.17	Variación de Empuje de los Rotores ante cambios en ψ_B	71
5.1	Esquema para el control de la altura en SIMULINK®	73
5.2	Respuesta del <i>Sqrt_Controller</i> ante entrada rampa del error en altitud	78
5.3	Respuesta de $\ddot{z}_B$ ante entrada escalón de $1m/s^2$	81
5.4	Respuesta de $\dot{z}_B$ ante entrada escalón de $1m/s$	81
5.5	Respuesta de z_B ante entrada escalón de $1m$	82
5.6	Respuesta de $\dot{z}_B$ ante varias entradas escalón	83
5.7	Error en altitud, velocidad y aceleración	84
5.8	Variación de Empuje de los Rotores ante cambios en $\dot{z}_B$	84
6.1	Esquema para el control de la posición en SIMULINK®	86
6.2	Respuesta de $\dot{x}_B$ ante entrada escalón de $1m/s$	92
6.3	Respuesta de x_B ante entrada escalón de $1m$	93
6.4	Respuesta de $\dot{x}_B$ ante varias entradas escalón	94
6.5	Error en posición, velocidad y actitud en dirección "x"	95
6.6	Variación de Empuje de los Rotores ante cambios en $\dot{x}_B$	95
6.7	Respuesta de $\dot{y}_B$ ante entrada escalón de $1m/s$	96
6.8	Respuesta de y_B ante entrada escalón de $1m$	97
6.9	Respuesta de $\dot{y}_B$ ante varias entradas escalón	97

6.10	Error en posición, velocidad y actitud en dirección “y”	98
6.11	Variación de Empuje de los Rotores ante cambios en $\dot{y}_B$	99

Índice de Tablas

2.1	Variables características del sistema quadrotor más carga	7
4.1	Ganancias características del controlador en Roll y valores máximos	53
4.2	Ganancias características del controlador RATE_RLL	55
4.3	Valor PWM en Gazebo y Ángulo ϕ_B Deseado	56
4.4	Análisis de la respuesta temporal de ϕ_B	57
4.5	Ganancias características del controlador en Pitch y valores máximos	61
4.6	Ganancias características del controlador RATE_PIT	62
4.7	Valores PWM en Gazebo y Ángulo θ_B Deseado	63
4.8	Análisis de la respuesta temporal de θ_B	63
4.9	Ganancias características del controlador en Yaw y valores máximos	67
4.10	Ganancias características del controlador RATE_YAW	68
4.11	Valores PWM en Gazebo y Velocidad ψ_B Deseada	69
4.12	Análisis de la respuesta temporal de ψ_B	70
5.1	Ganancias características del control en altitud y valores máximos	77
5.2	Ganancias características del controlador ACCEL_Z	80
5.3	Valores PWM en Gazebo y Velocidad $\dot{z}_B$ Generada	82
5.4	Análisis de la respuesta temporal de $\dot{z}_B$	83
6.1	Ganancias características del control en posición y valores máximos	87
6.2	Ganancias características del controlador VEL_XY	89
6.3	Valores PWM en Gazebo y Velocidad $\dot{x}_B$ Generada	93
6.4	Análisis de la respuesta temporal de $\dot{x}_B$	94
6.5	Valores PWM en Gazebo y Velocidad $\dot{y}_B$ Generada	97
6.6	Análisis de la respuesta temporal de $\dot{y}_B$	98

Índice de Códigos

4.1	Código del controlador <code>sqrt_controller</code> para roll y pitch	53
5.1	Código para establecer las velocidades de ascenso y descenso máximas	74
5.2	Código para establecer la aceleración máxima	74
5.3	Código para calcular las restricciones horizontales y verticales	76

Bibliografía

- [1] Oualid Araar and Nabil Aouf, *Full linear control of a quadrotor UAV, LQ vs H^∞* , Control (CONTROL), 2014 UKACC International Conference on.
- [2] ArduPilot Development Site, *Altitude Hold Mode*, [consulta: 28 de Enero de 2017]. Disponible en: <http://ardupilot.org/copter/docs/altholdmode.html>.
- [3] ArduPilot Development Site, *Complete Parameter List*, [consulta: 12 de Mayo de 2017]. Disponible en: <http://ardupilot.org/copter/docs/parameters.html>.
- [4] ArduPilot Development Site, *Stabilize Mode*, [consulta: 28 de Enero de 2017]. Disponible en: <http://ardupilot.org/copter/docs/stabilize-mode.html>.
- [5] Franz Bahner, *Modeling, simulation and control of a quadcopter carrying a slung load*, Master's thesis, Dep. Ingeniería de Sistemas y Automática. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla, 2015.
- [6] M. Bernard and K. Kondak, *Generic slung load transportation system using small size helicopters*, Robotics and Automation, 2009. ICRA '09. IEEE International Conference on.
- [7] Morten Bisgaard, Anders la Cour-Harbo, and Jan Dimon Bendtsen, *Adaptive control system for autonomous helicopter slung load operations*, Control Engineering Practice (2010).
- [8] Samir Bouabdallah, *Design and control of quadrotors with application to autonomous flying*, Ph.D. thesis, École Polytechnique federale de Lausanne, 2007.
- [9] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart, *Design and Control of an Indoor Micro Quadrotor*, Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on.
- [10] Samir Bouabdallah, André Noth, and Roland Siegwart, *PID vs LQ control techniques applied to an indoor micro quadrotor*, Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.
- [11] Samir Bouabdallah and Roland Siegwart, *Backstepping and sliding-mode techniques applied to an indoor micro quadrotor*, Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on.
- [12] Franz Bähner, Manuel Vargas, Manuel G. Ortega, and Francisco Rubio, *Control of a Quadrotor with Suspended Load. Experiences in Load-Based Flight Mode*, 2016.
- [13] Vilas K. Chitrakaran, Darren M. Dawson, Jian Chen, and Matthew Feemster, *Vision Assisted Autonomous Landing of an Unmanned Aerial Vehicle*, Decision and Control, and 2005 European Control Conference.

- CDC-ECC '05. 44th IEEE Conference on.
- [14] Jesús Egea Cárdenas, *Detección, seguimiento y estimación del centro de masas de un objeto suspendido de un quadrotor*, Master's thesis, Dep. Ingeniería de Sistemas y Automática. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla, 2015.
 - [15] ArduPilot Development Site, *Loiter Mode*, [consulta: 28 de Enero de 2017]. Disponible en: <http://ardupilot.org/copter/docs/loiter-mode.html>.
 - [16] Travis Dierks and Sarangapani Jagannathan, *Output Feedback Control of a Quadrotor UAV Using Neural Networks*, IEEE Transactions on Neural Networks (Volume: 21, Issue: 1, 2010).
 - [17] Erle Robotics, *APM Plane, APM Copter, APM Rover*, [consulta: 17 de Junio de 2017]. Disponible en: <https://github.com/erlerobot/ardupilot/tree/gazebo>.
 - [18] Ying Feng, Camille Alain Rabbath, and Chun-Yi Su, *Modeling of a Micro UAV with Slung Payload*, Handbook of Unmanned Aerial Vehicles, 2014.
 - [19] Fadri Furrer, Michael Burri, Markus Wilhelm Achtelik, and Roland Siegwart, *RotorS – A Modular Gazebo MAV Simulator Framework*, Robot Operating System (ROS): The Complete Reference (Volume 1).
 - [20] Leonard Hall, *Arducopter 2.9 PID Loops for STABILIZE ACRO and ALT_HOLD*, [consulta: 28 de Febrero de 2017]. Disponible en: <http://diydrones.com/profiles/blogs/arducopter-2-9-pid-loops-for-stabilize-acro-and-alt-hold>.
 - [21] Markus Hehn and Raffaello D' Andrea, *A Flying Inverted Pendulum*, Robotics and Automation (ICRA), 2011 IEEE International Conference on.
 - [22] Minh-Duc Hua, Tarek Hamel, Pascal Morin, and Claude Samson, *Introduction to Feedback Control of Underactuated VTOL Vehicles*, IEEE Control Systems, (Volume: 33, Issue: 1, 2013).
 - [23] Kristian Klausen, Thor I. Fossen, and Tor Arne Johansen, *Nonlinear Control of a Multirotor UAV with Suspended Load*, Unmanned Aircraft Systems (ICUAS), 2015 International Conference on.
 - [24] DongBin Lee, Timothy Burg, Bin Xian, and Darren Dawson, *Output Feedback Tracking Control of an Underactuated Quad-Rotor UAV*, 2016.
 - [25] Leonard Hall, *Stability Patch for Motor Max and Min and Yaw Curve*, [consulta: 18 de Febrero de 2017]. Disponible en: <http://diydrones.com/profiles/blogs/stability-patch-for-motor-max-and-min-and-yaw-curve>.
 - [26] Cultura Lusa, *Amazon présente un prototype de drone livreur de colis*, [consulta: 1 de Julio de 2017]. Disponible en: <http://www.culturalusa.com/news/technologie/amazon-presente-un-prototype-de-drone-livreur-de-colis/>.
 - [27] Akshay Nanjangud, *Simultaneous low-order control of a nonlinear quadrotor model at four equilibria*, Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on.
 - [28] David Onieva, *Me voy a comprar un drone, ¿con cámara integrada o instalo una después?*, [consulta: 1 de Julio de 2017]. Disponible en: <https://www.adslzone.net/2015/12/26/me-voy-a-comprar-un-drone-con-camara-integrada-o-instalo-una-despues/>.
 - [29] Ivana Palunko and Rafael Fierro, *Adaptive control of a quadrotor with dynamic changes in the center of gravity*, Proceedings 18th IFAC World Congress, 2011.
 - [30] Ivana Palunko, Rafael Fierro, and Patricio Cruz, *Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach*, Robotics and Automation (ICRA), 2012 IEEE International Conference on.

-
- [31] David Pérez-Piñar, *Ajuste del autopiloto en ARGOS–Fundamentos*, [consulta: 20 de Febrero de 2017]. Disponible en: <http://www.jovenescientificos.es/ajuste-del-autopiloto-en-argos-fundamentos/>.
- [32] Cesáreo Raimúndez and José Luis Camaño, *Transporting Hanging Loads Using a Scale Quad-Rotor*, CONTROLO '2014 - Proceedings of the 11th Portuguese Conference on Automatic Control.
- [33] Erle Robotics, *ardupilot_sitl_gazebo_plugin/ardupilot_sitl_gazebo_plugin/urdf/erlecopter_base.xacro*, [consulta: 7 de Febrero de 2017]. Disponible en: https://github.com/erlerobot/ardupilot_sitl_gazebo_plugin/blob/master/ardupilot_sitl_gazebo_plugin/urdf/erlecopter_base.xacro.
- [34] Erle Robotics, *ardupilot_sitl_gazebo_plugin/ardupilot_sitl_gazebo_plugin/urdf/erlecopter.xacro*, [consulta: 2 de Febrero de 2017]. Disponible en: https://github.com/erlerobot/ardupilot_sitl_gazebo_plugin/blob/master/ardupilot_sitl_gazebo_plugin/urdf/erlecopter.xacro.
- [35] S. Sadr, S. Ali A. Moosavian, and P. Zarafshan, *Dynamics modeling and control of a quadrotor with swing load*, Journal of Robotics (2014).
- [36] Koushil Sreenath, Taeyoung Lee, and Vijay Kumar, *Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load*, Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on.
- [37] Koushil Sreenath, Nathan Michael, and Vijay Kumar, *Trajectory generation and control of a quadrotor with a cable-suspended load - A differentially-flat hybrid system*, Robotics and Automation (ICRA), 2013 IEEE International Conference on.