



UNIVERSIDAD DE LA REPÚBLICA  
FACULTAD DE INGENIERÍA



# Proyecto AUION

**PROYECTO DE FIN DE CARRERA**  
**AUTOMATIZACIÓN DEL VUELO DE UN AVIÓN**

**Tutor: Ing. Rafael Canetti**

**Integrantes del grupo**

**Rodrigo Alonso**  
**Guzmán Hernández**  
**Pablo Iturralde**

**Montevideo - Uruguay**  
**Abril 2009**



# Índice general

---

<b>Índice general</b>	<b>2</b>
<b>1. Aeronáutica y modelo del avión</b>	<b>4</b>
1.1. Definición de la nomenclatura aeronáutica . . . . .	4
1.2. Modelo de avión utilizado: <i>DHC-2 BEAVER</i> . . . . .	9
1.2.1. Parámetros de geometría del avión . . . . .	10
1.2.2. Masa total y tensor de inercia . . . . .	11
<b>2. Modelado del sistema</b>	<b>12</b>
2.1. Modelado físico del sistema . . . . .	12
2.2. Modelado de la atmósfera y superficie terrestre . . . . .	16
2.3. Definición de los subconjuntos de variables . . . . .	18
2.4. Fuerzas y momentos involucrados . . . . .	19
2.4.1. Peso del avión . . . . .	19
2.4.2. Fuerzas y momentos aerodinámicos . . . . .	20
2.4.3. Fuerzas y momentos del motor . . . . .	21
2.4.4. Resumen de coeficientes . . . . .	22
2.4.5. Comentarios sobre el modelado de las fuerzas y momentos . . . . .	23
2.5. Resumen de efectos no considerados en el modelado . . . . .	27
2.6. Modelo en variables de estado completo . . . . .	29
<b>3. Simulación del lazo abierto</b>	<b>30</b>
3.1. Desarrollo del simulador . . . . .	31
3.1.1. Primer versión del simulador . . . . .	31
3.1.2. Versión del simulador en <i>simulink</i> . . . . .	31
3.2. Modelos implementados para el lazo abierto . . . . .	35
3.3. Pruebas del simulador . . . . .	35
3.3.1. Prueba 1: Caída libre . . . . .	36
3.3.2. Prueba 2: Planeo . . . . .	38
<b>4. Generador de trayectorias</b>	<b>39</b>
4.1. ¿Qué entendemos por generador de trayectorias? . . . . .	39
4.2. Elementos a considerar para la construcción del generador de trayectorias . . . . .	39
4.3. El generador de trayectorias implementado . . . . .	40
4.3.1. Especificación funcional del generador de trayectorias. . . . .	41
4.3.2. Análisis del problema geométrico en 2D. . . . .	42
4.3.3. Representación elegida. . . . .	42
4.3.4. Optimización de la solución. . . . .	43
4.3.5. Tratamiento de los puntos a distinta altura (generalización del programa a 3D). . . . .	47
4.3.6. Post-procesamiento de la solución optimizada. . . . .	48
4.4. Análisis de los resultados . . . . .	48

4.4.1.	Criterios de éxito. . . . .	48
4.4.2.	Indicadores cuantitativos. . . . .	49
4.4.3.	Casos de prueba y resultados . . . . .	50
4.4.4.	Conclusiones . . . . .	51
<b>5.</b>	<b>Linealización</b>	<b>52</b>
5.1.	Trayectorias estacionarias y puntos de operación . . . . .	53
5.2.	Vuelo en línea recta a velocidad constante . . . . .	54
5.3.	Giro a velocidad angular constante . . . . .	55
5.4.	Software de linealización y trimado . . . . .	57
<b>6.</b>	<b>Diseño del controlador</b>	<b>58</b>
6.1.	Gain Scheduling . . . . .	58
6.2.	Control lineal . . . . .	59
6.2.1.	Realimentación de estados . . . . .	59
6.2.2.	Cambio de variables de estado para el vuelo en círculos . . . . .	63
6.2.3.	Modelo del sistema para el análisis de los controladores . . . . .	64
6.2.4.	Agregado de dinámica para controlar errores grandes en la posición . . . . .	64
<b>7.</b>	<b>Software implementado</b>	<b>66</b>
7.1.	Software integrado . . . . .	66
7.1.1.	Especificaciones . . . . .	67
7.1.2.	Diseño de la integración . . . . .	67
7.1.3.	Interfaz gráfica . . . . .	67
<b>8.</b>	<b>Consideraciones generales para la implementación del sistema propuesto</b>	<b>70</b>
8.1.	El generador de trayectorias . . . . .	70
8.2.	Validación del modelo . . . . .	70
8.3.	Evaluación de necesidades en aviónica . . . . .	71
8.4.	Trimado y linealización . . . . .	71
8.5.	Observación de estados . . . . .	72
8.6.	Entradas y actuadores . . . . .	72
8.7.	Controlador lineal . . . . .	73
8.8.	Regeneración de trayectorias . . . . .	73
<b>A.</b>	<b>Cálculos realizados</b>	<b>75</b>
A.1.	Cálculos relativos a las linealizaciones . . . . .	75
A.1.1.	Linealización del MVE en torno a una trayectoria de operación . . . . .	76
A.1.2.	Linealización en torno a una trayectoria de giro a velocidad angular constante . . . . .	77
<b>B.</b>	<b>Definición de la simbología utilizada</b>	<b>79</b>
<b>C.</b>	<b>Acrónimos utilizados</b>	<b>85</b>
	<b>Índice de figuras</b>	<b>86</b>
	<b>Bibliografía</b>	<b>87</b>

# Aeronáutica y modelo del avión

El objetivo de este capítulo es definir el vocabulario técnico a utilizar, los distintos términos utilizados en el área de la aeronáutica y demás elementos que se utilizarán con fluidez en el resto de esta documentación. Además, queremos hacer un breve repaso del avión que disponemos, pues los esfuerzos realizados durante el transcurso de este proyecto se enmarcar en la línea de generar un piloto automático para el avión disponible por parte de Universidad de la República.

## 1.1. Definición de la nomenclatura aeronáutica

Los aviones, en su configuración estandar, constan de un fuselaje el cual representa el cuerpo del avión y de distintas superficies de vuelo, a saber, las alas propiamente dichas ubicadas a ambos lados del fuselaje y los estabilizadores los cuales se encuentran situados en la cola del avión. La cola del avión consta de dos estabilizadores horizontales y un estabilizador vertical. Por otro lado, el avión del que se dispone es propulsado mediante el giro de un hélice ubicada en la parte delantera del avión.

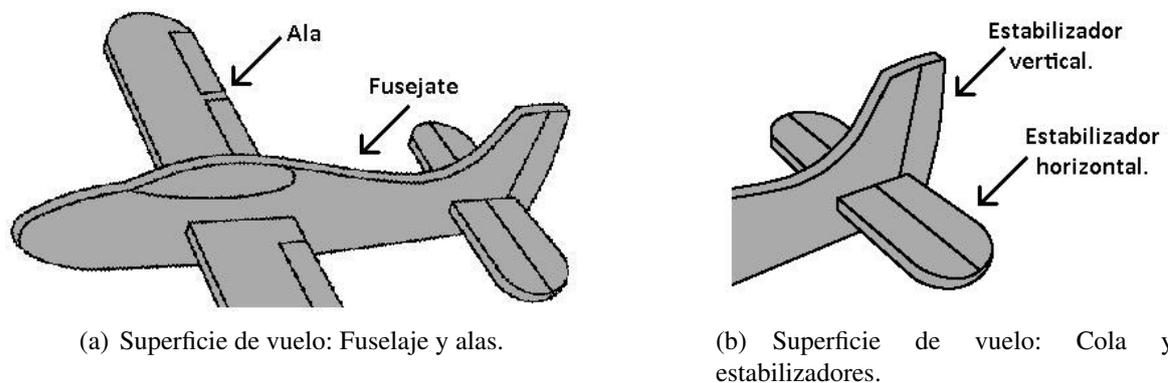


Figura 1.1: Distintos elementos que componen la configuración estandar en un avión. [9] *Imagen modificada.*

Si llamamos  $O$  al centro de gravedad del avión, podemos definir tres ejes que pasan por dicho punto. Si nos referimos a la figura 1.2 tenemos que; el eje  $Ox_B$  será el eje longitudinal, el eje  $Oy_B$  será el eje lateral y el eje  $Oz_B$  será el eje vertical. Las distintas rotaciones que el avión puede tener respecto a estos ejes son representadas en una base solidaria a él y son llamadas en el ambiente de la aeronáutica *roll*, *pitch* y *yaw*. Si colocamos una base ortonormal  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$  solidaria al avión <sup>a</sup>, tenemos que el *roll* corresponde a una rotación según el vector  $\hat{i}_B$ , el *pitch* corresponde a una rotación según el vector  $\hat{j}_B$  y finalmente el *yaw* se corresponde con una rotación respecto al vector  $\hat{k}_B$ , como se puede observar en la imagen 1.2.

La forma de maniobrar el avión es mediante la deflexión de sus superficies de vuelo y la variación de la velocidad a la que gira la hélice. La deflexión de las superficies de vuelo se logra mediante partes móviles situadas en dichas superficies de vuelo, mientras que la variación del giro de la hélice se logra acelerando el motor proporcionándole más combustible. Las partes móviles en las superficies de vuelo, también llamadas superficies de control, son:

<sup>a</sup>En el siguiente capítulo se hará más extensiva la explicación respecto a esta base.

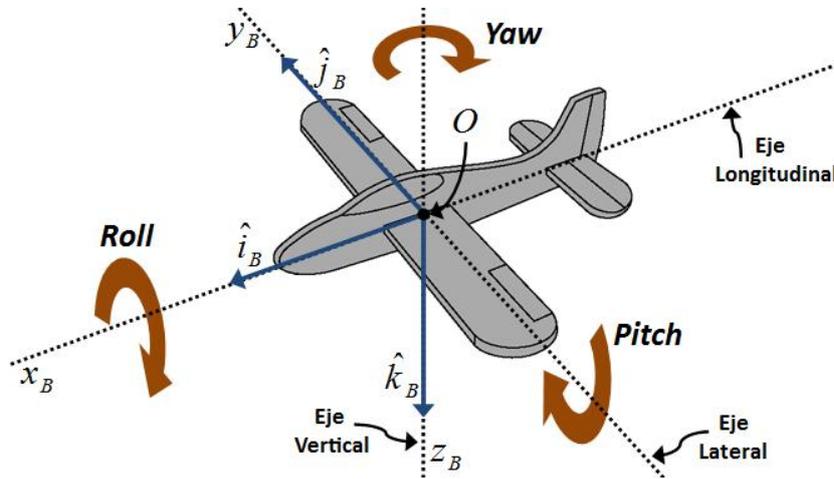


Figura 1.2: Descriptivo de los ejes longitudinal, lateral y vertical. Se puede notar el sentido definido positivo para las rotaciones de *roll*, *pitch* y *yaw*. [9] *Imagen modificada*.

- ▷ **Alerones** - Se encuentran situados en las alas, en la parte más alejada del fuselaje.
- ▷ **Elevadores** - Se encuentran situados en los estabilizadores horizontales.
- ▷ **Timón de cola** - Se encuentra situado en el estabilizador vertical.
- ▷ **Flaps** - Se encuentran situados en las alas, en la parte más cercana al fuselaje.

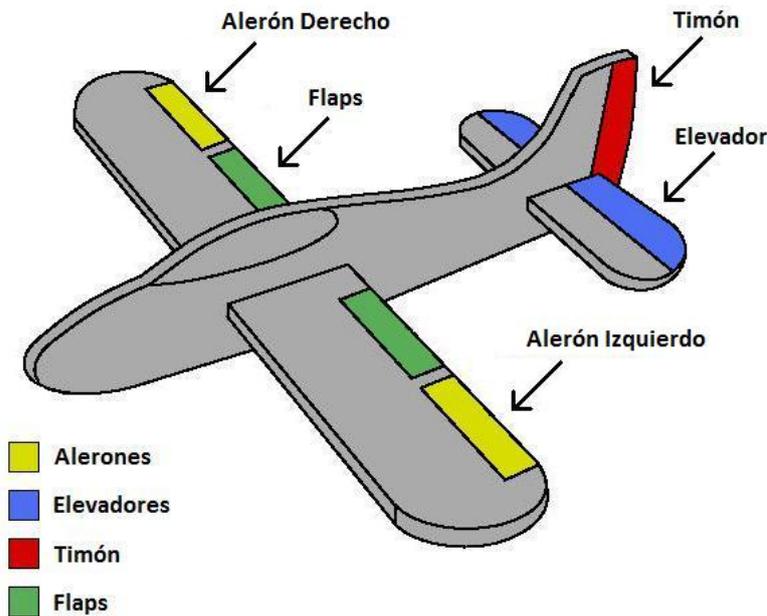
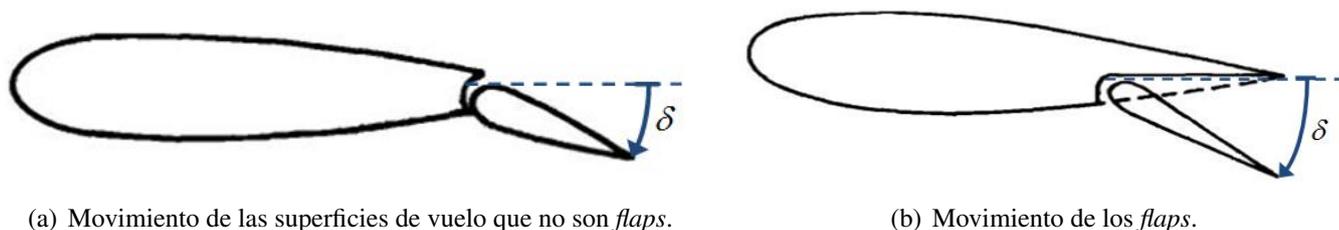


Figura 1.3: Descriptivo de las distintas superficies de vuelo. [9] *Imagen modificada*.

La ubicación de los alerones, elevadores, timón de cola y *flaps* se puede observar en la figura 1.3. Se puede apreciar también la convención que utilizaremos para definir cual será el alerón derecho y el alerón izquierdo, la izquierda y la derecha son tales que se corresponden con un observador parado en la parte trasera del avión mirando hacia la parte delantera del mismo.

El movimiento de éstas superficies será descrito por un ángulo, del cual interesa en particular definir desde donde está medido y cual es el sentido positivo. La forma en que se mueven los *flaps* es distinta a las demás superficies de vuelo, si bien el ángulo se medirá de la misma manera. En la figura 1.4 se puede observar la diferencia entre las dos formas de movimientos de las superficies de control. Cabe observar además de

dicha figura lo que respecta a la posición de la superficie móvil para  $\delta = 0$ , es decir, el origen de medición que utilizaremos para cada superficie de control. Los ángulos que definiremos a continuación se medirán de manera similar a la que se expone en dicha figura.



(a) Movimiento de las superficies de vuelo que no son *flaps*.

(b) Movimiento de los *flaps*.

Figura 1.4: Forma de medir los ángulos para la deflexión de las superficies de vuelo. [2] *Imagen modificada*.

La nomenclatura utilizada para medir estos desplazamientos, y que se utilizará en el resto de este documento, es la de llamar  $\delta'_a$  al ángulo correspondiente al alerón,  $\delta_e$  al ángulo correspondiente al elevador,  $\delta_r$  al ángulo que corresponde al movimiento del timón de cola<sup>b</sup> y  $\delta_f$  para el ángulo correspondiente al movimiento de los *flaps*. El movimiento de los alerones es conjugado, a saber, cuando el alerón derecho baja una cierta cantidad de radianes, el alerón izquierdo sube la misma cantidad de radianes. Por esto último es que a efectos del modelado, no nos interesará expresar las fuerzas en términos de  $\delta'_a$ , sino que lo haremos en términos de  $\delta_a$  siendo  $\delta_a = 2\delta'_a$ . Esto es consistente intuitivamente con el hecho de que cuando se mueve una cierta cantidad  $\delta'_a$  los alerones, al estar el movimiento conjugado, en un primer análisis las fuerzas aerodinámicas que aparecen en las superficies de vuelo apuntan hacia arriba en un ala y hacia abajo en la otra ala, generando un momento respecto al eje longitudinal del avión en el mismo sentido. En cuanto a lo que respecta a los elevadores, ambos se mueven en la misma dirección en todo momento, un valor  $\delta_e$  positivo generará un movimiento hacia abajo de los elevadores. Para el caso del timón de cola, la convención de signos utilizada para el ángulo  $\delta_r$ , es que cuando se tiene un  $\delta_r$  positivo el timón se moverá hacia la izquierda (respetando la definición de izquierda y derecha realizada párrafos anteriores). Finalmente, para el movimiento de los *flaps* no es necesaria ninguna aclaración dado que el movimiento sólo se puede efectuar en un sentido, ese único sentido en el que se puede mover será el definido positivo.

En la figura 1.5 aparecen representados los distintos ángulos utilizados, identificando su origen de medición y su sentido positivo. Ésta figura es para  $\delta'_a > 0$ ,  $\delta_e > 0$  y  $\delta_r > 0$ . Es de notar que  $\delta'_a > 0$  implica necesariamente que la variable  $\delta_a$  que utilizaremos es tal que  $\delta_a > 0$ .

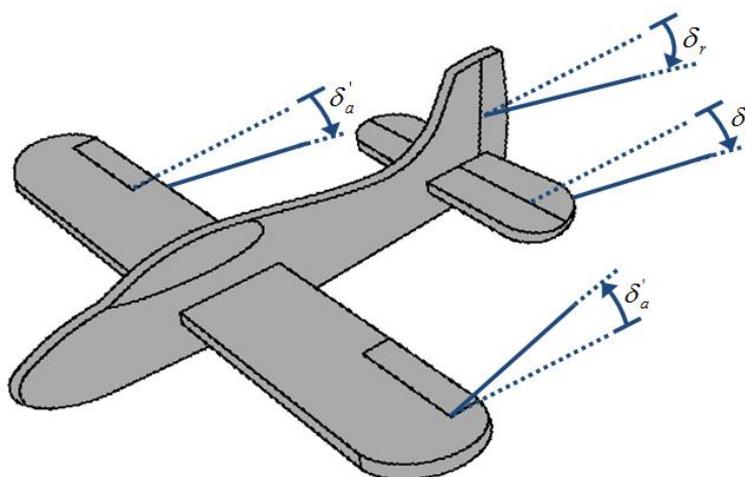


Figura 1.5: Descriptivo de los ángulos utilizados para medir los desplazamientos de los alerones, elevadores y timón. Se muestra en caso para valores de los parámetros tales que  $\delta'_a, \delta_e, \delta_r > 0$ . [9] *Imagen modificada*.

<sup>b</sup>El subíndice  $r$  que se utiliza para el movimiento  $\delta_r$  del timón de cola proviene del ingles *rudder*.

Como se verá más adelante, las fuerzas aerodinámicas dependen fuertemente -además de las deflexiones de las superficies de vuelo que se explicó en párrafos anteriores- de dos ángulos que se definen a partir del eje longitudinal, el eje lateral, el eje vertical, el vector velocidad lineal del avión que en este caso conviene expresarlo en términos de la base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$  como  $\vec{v}_B = v_{Bx}\hat{i}_B + v_{By}\hat{j}_B + v_{Bz}\hat{k}_B$ , y el vector velocidad del viento que también expresaremos en dicha base como  $\vec{v}_v = v_{vBx}\hat{i}_B + v_{vBy}\hat{j}_B + v_{vBz}\hat{k}_B$ . Éstos dos ángulos son; el ángulo de ataque al cual nos referiremos utilizando la letra  $\alpha$ , y el ángulo de *sideslip* para el cual se utiliza la letra  $\beta$ . Tanto  $\alpha$  como  $\beta$  se definen más específicamente en base a la velocidad relativa al viento del avión que llamaremos  $\vec{v}_{rB}$ , y se define como la diferencia entre la velocidad lineal del avión y la velocidad del viento. La forma en que queda expresada esta velocidad se muestra en la ecuación 1.1.

$$\begin{aligned} \vec{v}_{rB} = \vec{v}_B - \vec{v}_v &= (v_{Bx} - v_{vBx})\hat{i}_B + (v_{By} - v_{vBy})\hat{j}_B + (v_{Bz} - v_{vBz})\hat{k}_B \\ &= v_{rBx}\hat{i}_B + v_{rBy}\hat{j}_B + v_{rBz}\hat{k}_B \end{aligned} \quad (1.1)$$

Finalmente,  $\alpha$  es el ángulo comprendido entre la proyección de  $\vec{v}_{rB}$  sobre el plano  $Ox_Bz_B$  y el eje longitudinal  $Ox_B$ . Por otro lado el ángulo  $\beta$  es el ángulo que forman el vector velocidad relativa al viento  $\vec{v}_{rB}$  y el vector resultante de la proyección de dicha velocidad sobre el plano  $Ox_Bz_B$ . En la figura 1.6 se puede apreciar más claramente a qué ángulos nos referimos.

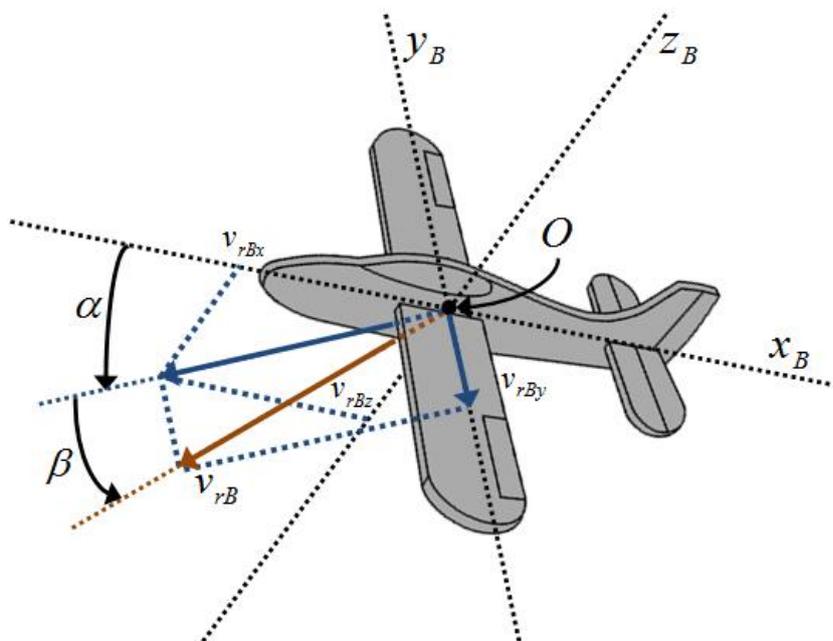


Figura 1.6: Ángulo de ataque  $\alpha$  y ángulo *sideslip*  $\beta$ . [9] Imagen modificada.

A partir de la figura 1.6 podemos observar de forma más clara como escribir los ángulos  $\alpha$  y  $\beta$  en función de las componentes  $v_{rBx}$ ,  $v_{rBy}$  y  $v_{rBz}$  de la velocidad lineal del avión. Los triángulos involucrados en la determinación de estos ángulos se pueden extraer de la figura 1.6 y son los que se muestra en la figura 1.7, de los cuales se pueden obtener las expresiones que se muestran en la ecuaciones 1.2 y 1.3. En el próximo capítulo se verá que el vector  $\vec{v}_{rB}$  es extremadamente importante en el modelado de la fuerzas y momentos aerodinámicos involucrados en el vuelo de un avión. En este punto podemos observar que dicho vector es necesario para poder definir los ángulos  $\alpha$  y  $\beta$ . En algunos casos se puede despreciar el efecto de la velocidad del viento  $\vec{v}_v$  respecto a la velocidad lineal del avión  $\vec{v}_B$ ; en nuestro caso queremos hacer un modelado lo más representativo posible. Por otro lado, la velocidad del viento es un parámetro externo a nuestro sistema - que no podemos controlar de ninguna manera - con lo cual no podríamos asegurar que dicha hipótesis se cumpla para todos los casos de situación de vuelo.

Cabe destacar en este punto, que si no se considera la velocidad del viento en el modelado, es decir, si se considera dicha velocidad como como perturbación al sistema, el efecto esperado es que el avión se aparte de la trayectoria deseada, situación que deberá ser solucionada por el sistema de control. Por lo tanto,

no considerarlo no representa un problema mayor en los casos usuales. En este trabajo, en algunos puntos se deberá realizar esta hipótesis y se considerará el viento con velocidad nula, tomando su efecto en la realidad como una de las tantas perturbaciones al sistema.

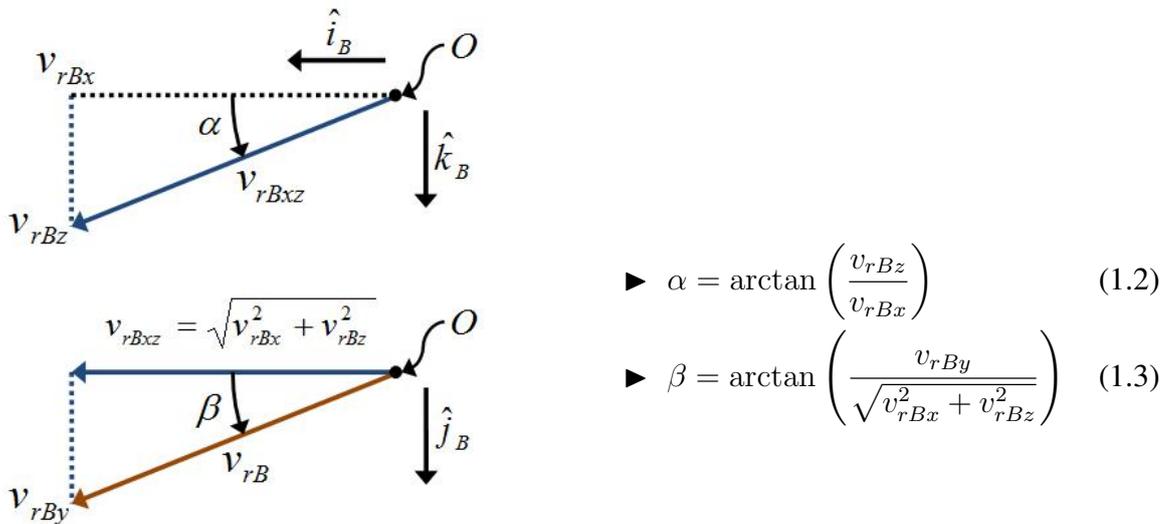


Figura 1.7: Triángulos involucrados en la determinación de  $\alpha$  y  $\beta$ .

Finalmente, nos quedan por definir diversos aspectos y parámetros que se obtienen a partir de la geometría del avión. Los más relevantes y que serán los que abordaremos en estos últimos párrafos de esta sección son: la superficie alar, la cuerda aerodinámica media, el ángulo de diedro, el ángulo de flecha (*sweepback*) y la envergadura del avión (*wing span*).

► **Superficie alar.**

A esta magnitud la denotaremos con la letra *S*. Es la superficie total de un ala del avión. Está íntimamente ligada a las fuerzas aerodinámicas, pues, éstas últimas son directamente proporcional a dicha superficie. Por lo tanto es un parámetro impotántísimo a definir, en un predimensionamiento para el diseño de un avión es uno de los primeros parámetros que se definen.

► **Cuerda aerodinámica media.**

MAC por su siglas en ingles (*Mean Aerodynamic Chord*), es otro de los parámetros importantes a efectos del diseño y/o el modelado del sistema. Para definir esta magnitud primero debemos definir qué se entiende por cuerda. Dado un perfil de ala infinito según el eje perpendicular a la hoja y de sección homogénea, la cuerda aerodinámica a la cual denominaremos *c* es la distancia que aparece descrita en la figura 1.8.

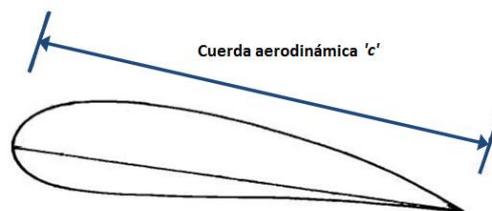


Figura 1.8: Cuerda aerodinámica para un perfil infinito y homogéneo. [2] *Imagen modificada.*

En el caso de estar frente a un ala finita la cual carezca de un perfil homogéneo, entonces se define la MAC como el promedio de las cuerdas aerodinámicas a lo largo del ala, al que denotaremos según el parámetro  $\bar{c}$ . Este promedio es claro que se trata de un promedio continuo. En el caso de que el perfil varíe de forma continua (lo cual siempre es válido en la realidad), esta MAC  $\bar{c}$  coincide con una cuerda aerodinámica  $c$  para una sección de ala a una determinada distancia del fuselaje.

▷ **Ángulo de diedro.**

El ángulo de diedro es el ángulo que forman la posición de las alas con el eje lateral del avión si se observa el avión sobre el plano  $Oy_Bz_B$ . Este ángulo por lo general es pequeño y muchas veces es nulo.

▷ **Sweepback. Ángulo de flecha.**

El ángulo de *sweepback* o ángulo de flecha, es el ángulo que forman la posición de las alas con el eje longitudinal del avión si se observa el avión sobre el plano  $Ox_By_B$ . Este ángulo se puede observar en la figura 1.9.

▷ **Wing Span. Envergadura.**

La envergadura de un avión se define como la distancia desde una punta de un ala a la punta de la otra ala. En la figura 1.9 se puede observar además cuál es esta distancia. Esta magnitud es también muy importante para la definición de las fuerzas aerodinámicas, y muy en particular en los momentos que ejercen dichas fuerzas.

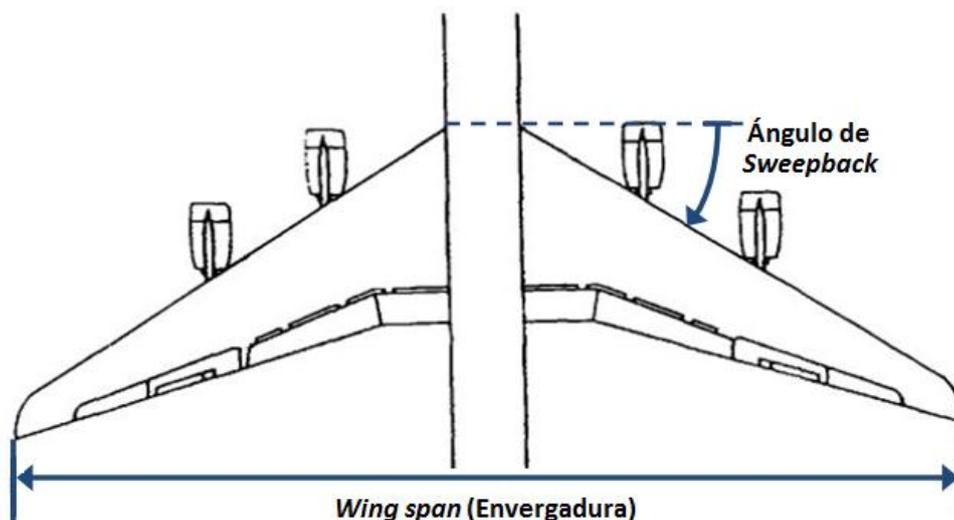


Figura 1.9: Wing Span y ángulo de sweepback. [2] Imagen modificada.

## 1.2. Modelo de avión utilizado: *DHC-2 BEAVER*

Dada la imposibilidad de realizar las medidas necesarias sobre el avión disponible dentro de los márgenes de duración del proyecto, se decidió utilizar un modelo de avión disponible en la bibliografía. El avión a escoger debía ser - dentro de lo posible - lo más parecido al avión que disponíamos, en particular nos interesaba fundamentalmente similitud en el mecanismo de propulsión, a saber, propulsión mediante una única hélice situada en la parte delantera del avión. Otros aspectos a tener en cuenta eran la característica de la

alas y la estructura general del avión.

El modelo de avión que finalmente utilizamos para el desarrollo de este proyecto fue el *BEAVER*. Es claro que existen notorias diferencias con el avión que dispone Universidad de la República, pero es importante que comparten varias características lo cual hace de este modelo una aproximación cercana al avión que se dispone en la realidad. Este avión fue confeccionado por la empresa canadiense *The De Havilland Aircraft of Canada Ltd.* (LHC de ahora en más) durante la década del 50, y el nombre completo del modelo es *DHC-2 BEAVER*. Disponemos según las referencias [5] y [10] todos los valores numéricos para los parámetros que se han venido definiendo durante el transcurso de este capítulo, pero lo más importante es que disponemos de la relación funcional entre las distintas fuerzas y momentos aerodinámicos y del motor con las distintas superficies y variables de control.

El modelo es un modelo aceptable para una vasta cantidad de situaciones de vuelo posibles según la referencia [5], si bien dicho modelo fue obtenido mediante simulaciones en un rango de velocidades entre  $35m/s$  y  $55m/s$ . En la medida que se trabaje en este rango de velocidades se asegura que el modelo es excelente. Es claro que el presente modelo deja de ser válido en situaciones de despegue y aterrizaje, en cuyo caso la referencia [5] brinda algunas correcciones a realizar de modo de seguir utilizando el modelo sin grandes cambios de estructura <sup>c</sup>. Esto de por sí no es del todo problemático pues en este proyecto nos concentraremos en el *tracking* de trayectorias una vez el avión se encuentre en situación de vuelo, y nos independizaremos del problema del aterrizaje y del despegue.

En la figura 1.10 se puede observar un dibujo esquemático del *BEAVER*, en donde se puede apreciar su geometría básica.

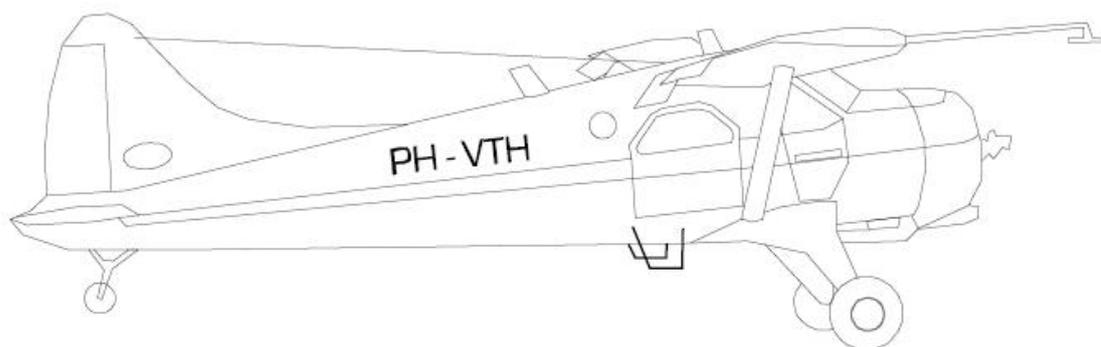


Figura 1.10: Vista lateral del *BEAVER*. [5].

### 1.2.1. Parámetros de geometría del avión

Es esta subsección expondremos los parámetros que dependen de la geometría del avión, de los cuales se han realizado diversas explicaciones durante el desarrollo de este capítulo. En la tabla 1.1 que se expone a continuación se pueden observar los valores obtenidos para las magnitudes de interés para esta subsección.

Parámetros de la geometría del avión <i>BEAVER</i>	
Superficie alar - $S$	$23,23m^2$
MAC - $\bar{c}$	$1,5875m$
Wing span - $b$	$14,63m$
Ángulo de diedro	$1^\circ$
Ángulo de <i>sweepback</i>	$0^\circ$
Longitud el fuselaje	$9,22m$
Altura	$2,74m$

<sup>c</sup>Por ejemplo, cambian algunos parámetros del modelado del motor

Perfil del ala	NACA 64 A 416
Diametro de la hélice	2,59m
Motor	Motor de pistones con 450HP@2300rpm
Tanque de combustible	521l

Cuadro 1.1: Tabla de parámetros referentes a la geometría del *BEAVER*. [5][10].

## 1.2.2. Masa total y tensor de inercia

Dos parámetros que serán de particular interés para el capítulo siguiente - donde desarrollaremos el modelado físico del sistema - son: la masa total del avión y el tensor de inercia del mismo.

El valor utilizado para la masa total del avión  $m$ , será entonces:

### ▷ Masa total del avión:

$$m = 2288,231kg \quad (1.4)$$

En lo referente al tensor de inercia del avión, utilizaremos la nomenclatura usual para los momentos de inercia de la forma  $I_{xx}$ ,  $I_{yy}$  y  $I_{zz}$ , y para los productos de inercia  $I_{xy}$ ,  $I_{xz}$ ,  $I_{yx}$ ,  $I_{yz}$ ,  $I_{zx}$  y  $I_{zy}$ . Es claro que el tensor de inercia que nos interesa es aquél que está medido en los ejes principales del avión, es decir, el eje longitudinal, el eje lateral y el eje vertical. Este tensor estará medido entonces desde el punto  $O$  y expresado en términos de la base del SB. El tensor que corresponde al *BEAVER* se puede observar en la ecuación 1.5.

### ▷ Tensor de inercia del avión:

$$I_{OB} = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \Rightarrow \begin{cases} I_{xx} = 5368,39kgm^2 \\ I_{yy} = 6928,93kgm^2 \\ I_{zz} = 11158,75kgm^2 \\ I_{xz} = I_{zx} = -117,64kgm^2 \\ I_{xy} = I_{yx} = I_{zy} = I_{yz} = 0 \end{cases} \quad (1.5)$$

Más adelante será de utilidad tanto la matriz  $I_{OB}$  como su inversa. La inversión de esta matriz,  $I_{OB}^{-1}$ , se puede obtener de manera sencilla aplicando la regla de *Kramer* por ejemplo, y la misma se muestra a continuación en la ecuación 1.6.

$$I_{OB} = \begin{pmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{pmatrix} \Rightarrow I_{OB}^{-1} = \frac{1}{\Upsilon} \begin{pmatrix} I_{zz} & 0 & -I_{xz} \\ 0 & \Upsilon/I_{yy} & 0 \\ -I_{xz} & 0 & I_{xx} \end{pmatrix} \quad (1.6)$$

Con:  $\Upsilon = |I_{OB}|/I_{yy} = I_{xx}I_{zz} - I_{xz}^2$ .

---

## Capítulo 2

# Modelado del sistema

---

A efectos de lograr cualquier tipo de control sobre un determinado sistema, se hace completamente indispensable conocer este último en profundidad. Es decir, es imprescindible conocer de forma fehaciente las propiedades del sistema, las ecuaciones rigen su dinámica, así como los distintos comportamientos frente a las diferentes situaciones que se pueden llegar a plantear. No es sólo importante obtener las ecuaciones que gobiernan la dinámica, sino también lograr una comprensión del sistema tal que los resultados que se desprenden de esas ecuaciones resulten naturales e intuitivos, o por lo menos comprensibles y coherentes con la situación de se desea modelar. Es por todo esto que el primer paso a dar en camino a obtener un sistema de vuelo autónomo, es conseguir un Modelo en Variables de Estado (MVE) del sistema, es decir, un modelo de la forma  $\dot{\vec{x}} = \vec{f}(\vec{x}, \vec{u}, t)$ ; donde  $\vec{x}$  es el vector de estados del sistema,  $\vec{u}$  el vector de entradas al sistemas y  $t$  el tiempo (que podría aparecer en principio explícitamente en las ecuaciones).

Como se puede desprender de lo anterior, es completamente necesario definir cual es el sistema a modelar, o dicho de otra manera, cual es el sistema a controlar. En nuestro caso el sistema a modelar es un avión en situación de vuelo, o más abstractamente, una cierta masa distribuida según una cierta geometría que, en situación de vuelo, está sometida a distintas fuerzas aerodinámicas sobre sus superficies. Se encuentra disponible en este informe en el apéndice A una pequeña reseña sobre mecánica de los fluidos, materia que fue muy necesaria de aprender y entender para lograr los objetivos del proyecto. En base a esto pasaremos a explicar la obtención del MVE de nuestro sistema.

## 2.1. Modelado físico del sistema

Antes de empezar a deducir las distintas ecuaciones que gobiernan la dinámica de nuestro sistema, es conveniente gastar unas cuantas líneas a fin de definir de buena manera las diferentes coordenadas y magnitudes físicas que formarán parte del vector de estados de nuestro MVE. Por otro lado, siempre que se desea realizar un correcto modelado de una situación física se debe especificar claramente que sistemas de referencia y de coordenadas se están utilizando, no sólo para la comprensión posterior sino también para poder chequear la consistencia de todos los pasos realizados.

Como primer sistema de referencia se escogió el sistema  $(\hat{i}_I, \hat{j}_I, \hat{k}_I)$ , un sistema de referencia inercial fijo al que llamaremos Sistema Inercial (SI) y las magnitudes medidas en la base de ese referencial se notarán con el subíndice I. El otro sistema de referencia que se utilizará es el sistema  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$  al que llamaremos Sistema Body (SB), en este caso las magnitudes medidas en ese referencial se notarán con el subíndice B. Dado que, por ejemplo, el momento de inercia del avión es constante en el SB <sup>a</sup> o que el SI es poco conveniente para expresar las fuerzas aerodinámicas que actúan sobre las superficies de vuelo, es de particular interés expresar las magnitudes físicas y las distintas ecuaciones en la base del SB.

En la figura 2.1 se muestra de forma clara estos sistemas de medida. Por otro lado es de notar que la posición absoluta del avión se puede expresar desde el SI como  $\vec{r} = x\hat{i}_I + y\hat{j}_I - z\hat{k}_I$ , y por lo tanto la velocidad absoluta expresada en esa base será  $\vec{v}_I = \dot{x}\hat{i}_I + \dot{y}\hat{j}_I - \dot{z}\hat{k}_I$ .

---

<sup>a</sup>En realidad esta afirmación no sería completamente cierta, pues sin duda tenemos una variación de la masa y del tensor de inercia del avión debido al consumo de nafta por parte del motor. Efecto que consideramos despreciable y modelable dentro de las perturbaciones del sistema.

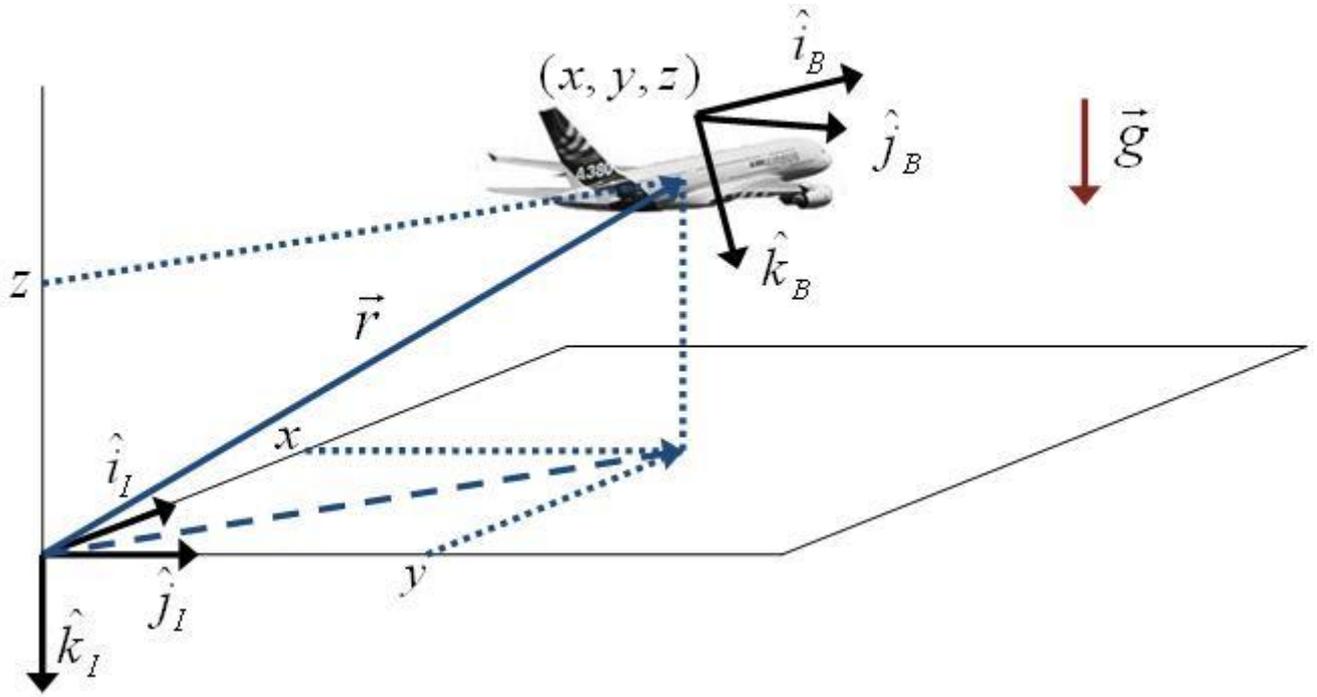
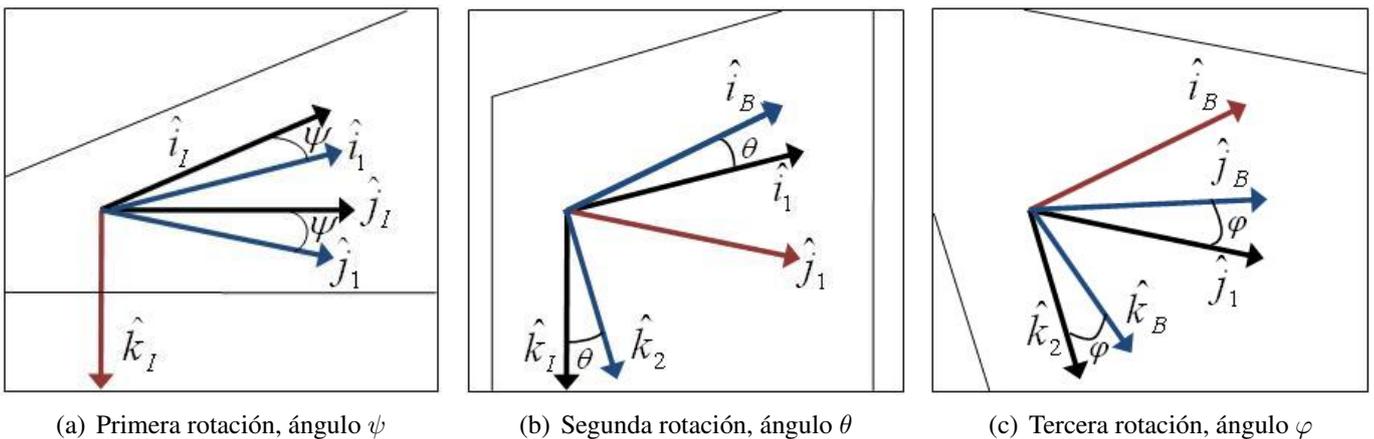


Figura 2.1: Ilustrativa de los distintos sistemas de referencia a utilizar y las coordenadas  $(x, y, z)$

Si siguiendo con el análisis, una relación importante a obtener es la relación que existe entre la velocidad absoluta del avión medida desde SI  $\vec{v}_I$  y la velocidad absoluta del avión medida en la base del SB  $\vec{v}_B$ . Esta relación viene dada simplemente por el cambio de base existente entre ambas bases ortonormales  $(\hat{i}_I, \hat{j}_I, \hat{k}_I)$  y  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$ . Para esto se toma como convención realizar tres rotaciones en un determinado orden desde el SI hasta el SB. Estas rotaciones son las correspondientes a la elección de los ángulos de Euler  $(\varphi, \theta, \psi)$ , donde la primera rotación es la correspondiente a un ángulo  $\psi$  en torno al eje  $\hat{k}_I$ . Esto define otra base ortonormal  $(\hat{i}_1, \hat{j}_1, \hat{k}_1)$ . Luego se realiza la segunda rotación correspondiente a una rotación de ángulo  $\theta$  alrededor del eje  $\hat{j}_1$ , definiendo así la segunda base ortonormal  $(\hat{i}_2, \hat{j}_2, \hat{k}_2)$ . Finalmente la última rotación necesaria es la rotación de ángulo  $\varphi$  en torno al eje  $\hat{i}_2$ , obteniéndose finalmente la base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$  correspondiente al SB. Este proceso se ilustra en la figura 2.2.



(a) Primera rotación, ángulo  $\psi$

(b) Segunda rotación, ángulo  $\theta$

(c) Tercera rotación, ángulo  $\varphi$

Figura 2.2: Esquema de las distintas rotaciones en base a los ángulos de Euler. Por convención desde SI a SB.

Utilizando las ideas que se explicaron en el párrafo anterior es que se obtiene la relación entre las velocidades absolutas del avión expresadas en las bases de SI y SB.  $\vec{v}_B = H_2^B H_1^2 H_I^1 \vec{v}_I = H_I^B \vec{v}_I$ . Las matrices  $H_2^B$ ,  $H_1^2$  y  $H_I^1$  son las matrices de cambio de base entre las bases que correspondan, son relativamente sencillas de hallar y corresponden a cada una de las rotaciones en el orden correcto.

$$H_I^1 = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad H_1^2 = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad H_2^B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{pmatrix} \quad (2.1)$$

Dado que las matrices que se definen en la ecuación 2.1 son matrices de rotaciones, es fácil ver que sus inversas se obtienen efectuando rotaciones opuestas, es decir en vez de rotar  $(\varphi, \theta, \psi)$  se rota  $(-\varphi, -\theta, -\psi)$ . Esto último habilita a obtener las inversas a estas matrices de rotación sustituyendo los senos por -senos. Utilizando dichas matrices es que se obtiene finalmente la matriz de cambio de base del SI al SB como  $H_I^B = H_2^B H_1^2 H_I^1$ .

$$H_I^B = \begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi & \sin \theta \sin \psi \sin \varphi + \cos \varphi \cos \psi & \cos \theta \sin \varphi \\ \sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi & \sin \theta \sin \psi \cos \varphi - \sin \varphi \cos \psi & \cos \theta \cos \varphi \end{pmatrix} \quad (2.2)$$

Procediendo de la forma que se explicó en el párrafo anterior, con respecto a la obtención de la matriz inversa, se puede obtener  $(H_I^B)^{-1} = H_B^I = H_1^I H_2^1 H_B^2$  simplemente haciendo la sustitución de los senos por -senos y multiplicando las matrices en el orden inverso al que fueron compuestas originalmente. La matriz de rotación queda entonces como se muestra en la ecuación 2.3.

$$H_B^I = \begin{pmatrix} \cos \theta \cos \psi & \sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi & \sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi \\ \cos \theta \sin \psi & \sin \theta \sin \psi \sin \varphi + \cos \varphi \cos \psi & \sin \theta \sin \psi \cos \varphi - \sin \varphi \cos \psi \\ -\sin \theta & \cos \theta \sin \varphi & \cos \theta \cos \varphi \end{pmatrix} \quad (2.3)$$

Una vez obtenido el cambio de base  $H_B^I$ , podemos expresar la relación entre las velocidades absolutas expresadas en las bases de ambos sistemas de referencia como se muestra en la ecuación 2.3.

$$\boxed{\begin{pmatrix} \dot{x} \\ \dot{y} \\ -\dot{z} \end{pmatrix} = H_B^I \begin{pmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{pmatrix}} \quad (2.4)$$

Siguiendo con un enfoque cinemático en el análisis, ya que hallamos una relación entre las velocidades lineales absolutas medidas en ambas bases, pasaremos ahora a obtener la relación existente entre la velocidad angular absoluta expresada en la base del SB y los ángulos de *Euler*  $(\varphi, \theta, \psi)$  que propusimos para expresar las diferentes rotaciones que llevan de un sistema de medida al otro.

Dado como se definió párrafos más arriba las diferentes rotaciones de ángulos  $\varphi, \theta$  y  $\psi$ , se puede notar que la velocidad angular del avión se puede escribir en términos de los versores intervinientes en el proceso de cambio de base según las rotaciones que se definieron como:  $\vec{w} = \psi \hat{k}_I + \theta \hat{j}_1 + \dot{\varphi} \hat{x}_B$ <sup>b</sup>. La intención ahora es expresar esta velocidad angular en términos de la base del SB. Esto se puede lograr observando que conocemos la relación existente entre los versores  $\hat{k}_I$  y  $\hat{j}_1$  con la base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$ . Utilizando la matriz de cambio de base de la ecuación 2.2 se puede obtener fácilmente que  $\hat{k}_I = -\sin \theta \hat{i}_B + \cos \theta \sin \varphi \hat{j}_B + \cos \theta \cos \varphi \hat{k}_B$ . Para el caso del otro versor ( $\hat{j}_1$ ) tenemos que trabajar un poquito más, por ejemplo, una forma de hacerlo es hallar la matriz  $H_1^B$  utilizando las matrices definidas en la ecuación 2.1.

$$H_1^B = H_2^B H_1^2 = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \theta \sin \varphi & \cos \varphi & \cos \theta \sin \varphi \\ \sin \theta \cos \varphi & -\sin \varphi & \cos \theta \cos \varphi \end{pmatrix} \Rightarrow \hat{j}_1 = \cos \varphi \hat{j}_B - \sin \varphi \hat{k}_B$$

Resultado que es razonable dado que una vez hecha la primer rotación, la única rotación que influye sobre el versor  $\hat{j}_1$  es la tercera rotación de ángulo  $\varphi$ . Habiendo obtenido estas relaciones y recordando que  $\vec{w} = \psi \hat{k}_I + \theta \hat{j}_1 + \dot{\varphi} \hat{x}_B$  estamos en condiciones de expresar el resultado final como se muestra en la ecuación

<sup>b</sup>Esto es posible debido a que las velocidades angulares se pueden sumar vectorialmente sin necesidad de establecer un orden en las rotaciones.

2.5.

$$\begin{aligned}\vec{w} = \vec{w}_B &= (\dot{\varphi} - \dot{\psi} \sin \theta) \hat{i}_B + (\dot{\psi} \cos \theta \sin \varphi + \dot{\theta} \cos \varphi) \hat{j}_B + (\dot{\psi} \cos \theta \cos \varphi - \dot{\theta} \sin \varphi) \hat{k}_B \\ &= w_{Bx} \hat{i}_B + w_{By} \hat{j}_B + w_{Bz} \hat{k}_B\end{aligned}\quad (2.5)$$

Esta última relación se puede escribir más amigablemente en forma matricial. La matriz en cuestión será llamada matriz L y su correspondiente inversa M, M se puede hallar analíticamente realizando una considerable cantidad de cuentas. Estas matrices se muestran en la ecuación 2.6, y relacionan la velocidad angular medida en la base del SB con la derivada temporal de los ángulos de *Euler*.

$$L = \begin{pmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \varphi & \cos \theta \sin \varphi \\ 0 & -\sin \varphi & \cos \theta \cos \varphi \end{pmatrix} \quad M = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{pmatrix}\quad (2.6)$$

Finalmente la relación que estábamos buscando queda expresada según la ecuación 2.7.

$$\boxed{\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = M \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix}}\quad (2.7)$$

Hasta ahora se hizo un análisis principalmente cinemático del sistema a controlar. Las ecuaciones que se deducieron hasta el momento no nos dicen nada sobre la dinámica del sistema, sino que sólo nos aportan relaciones cinemáticas entre distintas magnitudes físicas involucradas en el modelo. A efectos de modelar esta dinámica las herramientas utilizadas fueron las cardinales <sup>c</sup>. En las siguientes ecuaciones se hará uso de la regla de derivación que se muestra en la ecuación 2.8, notar que recién aquí se hace explícito que el sistema SB rota con el avión. Esta ecuación vincula las derivadas temporales vistas desde los dos distintos referenciales con la velocidad angular con la que se mueve uno respecto al otro, una cosa importante a hacer notar es que esta ecuación es completamente independiente de la base en que se exprese  $\vec{q}$ .

$$\left( \frac{d\vec{q}}{dt} \right)_I = \left( \frac{d\vec{q}}{dt} \right)_B + \vec{w}_B \times \vec{q}\quad (2.8)$$

Una vez obtenidas estas ecuaciones, que son las referidas al modelado de la dinámica del sistema, estamos en condiciones de afirmar que tenemos modelado el sistema. El conjunto de ecuaciones que aparecen recuadradas en esta sección son las ecuaciones que utilizamos como MVE.

▷ **Primera cardinal:**

$$\vec{F}_{Neta} = \frac{d(m\vec{v}_I)}{dt} = {}^d m \dot{\vec{v}}_I = m[\dot{\vec{v}}_B + \vec{w}_B \times \vec{v}_B]\quad (2.9)$$

$$\boxed{\begin{pmatrix} \dot{v}_{Bx} \\ \dot{v}_{By} \\ \dot{v}_{Bz} \end{pmatrix} = \frac{1}{m} \left[ \vec{F}_{Neta} - m \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times \begin{pmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{pmatrix} \right]}\quad (2.10)$$

<sup>c</sup>Es claro que la física ofrece otras herramientas para realizar este modelado. Para este caso en particular las ecuaciones son fácilmente obtenibles utilizando las cardinales.

<sup>d</sup>Aquí hemos hecho la suposición de que la masa del avión es constante, cosa que no es del todo cierta dado que existe una pérdida de masa debido al consumo de nafta. Consideramos esta pérdida despreciables y modelable dentro de las perturbaciones al sistema.

▷ **Segunda cardinal:**

$$\vec{T}_{Neto} = \frac{d(\vec{L}_O)}{dt} = \frac{d(I_{OB}\vec{w}_B)}{dt} = {}^d I_{OB}\dot{\vec{w}}_B + \vec{w}_B \times I_{OB}\vec{w}_B \quad (2.11)$$

$$\begin{pmatrix} \dot{w}_{Bx} \\ \dot{w}_{By} \\ \dot{w}_{Bz} \end{pmatrix} = I_{OB}^{-1} \left[ \vec{T}_{Neto} - \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times I_{OB} \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \right] \quad (2.12)$$

Es importante en este punto hacer notar la trascendencia de las ecuaciones 2.10 y 2.12. Estas son las ecuaciones que rigen la dinámica del sistema. La comprensión y resolución de estas ecuaciones es fundamental a efectos de los objetivos de este proyecto, se trata sin duda de un sistema no lineal, el cual es complejo de analizar y controlar. Hasta este momento hemos obtenido un sistema de ecuaciones de la forma  $\vec{x} = f(\vec{x}, \vec{F}_{Neto}, \vec{T}_{Neto})$ , usando como vector de estados  $\vec{x} = (x, y, z, \varphi, \theta, \psi, v_{Bx}, v_{By}, v_{Bz}, w_{Bx}, w_{By}, w_{Bz})$ . Es claro que aún no tenemos un modelo completo. Es necesario para esto saber como son  $\vec{F}_{Neto}$  y  $\vec{T}_{Neto}$ , y como varían en situación de vuelo. Este problema merece una sección aparte y será tratado en próximas secciones.

## 2.2. Modelado de la atmósfera y superficie terrestre

A continuación nos plantearemos el modelado de la atmósfera terrestre. Como se explicó en el capítulo anterior, el avión que se tomó como base para este modelado es el *BEAVER*. Dicho avión tiene una altura típica de vuelo de  $1800m$ . En esta sección pretendemos obtener las correcciones necesarias para un avión volando a alturas del orden de los  $1800m$ . Las ecuaciones que obtendremos en esta sección serán válidas independientemente del avión que se utilice, con lo cual si se desea trabajar con otro avión a otra altura de vuelo las herramientas que se mostrarán en esta sección siguen siendo válidas y se podrán hacer sencillamente los arreglos correspondientes, siempre y cuando no se excedan los  $11000m$  de altura, que es la altura donde se considera que el modelo dejar de ser válido.

El modelo de la atmósfera que se utilizó es un modelo completamente estandar, a saber, es el que utilizan todas las agencias aeronáuticas alrededor del globo. El mismo corresponde al modelado de la atmósfera como un gas ideal, donde se tomarán ciertos valores a nivel del mar para la presión y la temperatura. Es de destacar, que estamos sobre hipótesis más que suficientes para que el modelo de gas ideal sea extremadamente adecuado <sup>e</sup>. Otra hipótesis importante es que consideraremos el aire en estado estacionario, es decir, con las presiones definidas por la fluido-estática, no consideraremos variaciones de presión por velocidad del viento en este modelado de la atmósfera. Así mismo se utilizará un valor estandar para la masa molar del aire  $\bar{M}_a$ , donde de ahora en más nos referiremos a las variables con base molar (magnitudes expresadas por mol de sustancia) de la forma  $\bar{\cdot}$ . Recordamos que la constante universal de los gases  $\bar{R}$  tiene un valor de  $\bar{R} = 8,3145kJ/kMolK$ , y que se puede obtener el  $R$  de cada gas como  $R = \frac{\bar{R}}{\bar{M}}$ . De esta manera, los valores utilizados para los parámetros mencionados son los siguientes:

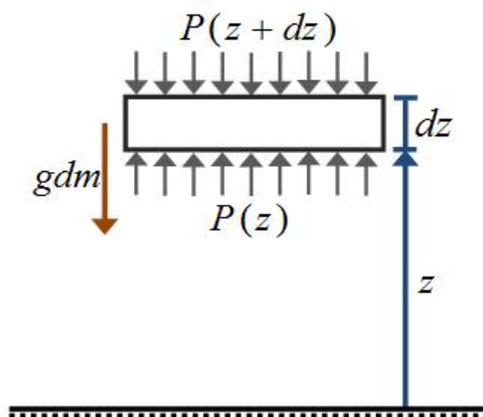
- ▶  $P_0 = 101,312kPa$
- ▶  $T_0 = 15^\circ C = 288,15K$
- ▶  $\bar{M}_a = 28,97kg/kMol \Rightarrow R_a \approx 0,287kJ/kgK$

Estos son los valores estandar que se utilizan para el modelado de la atmósfera. Finalmente, la última

<sup>e</sup>El aire se trata de un gas, y en los rangos de presiones y temperaturas en los que trabajaremos el modelo es adecuado. En particular, pues trabajaremos a presiones muy por debajo de la presión crítica del aire lo cual sería suficiente, de todos modos también las temperaturas involucradas son mayores que la temperatura crítica, y por lo tanto hace más válido aún el modelo. [11].

hipótesis que se realizará es que la temperatura de la atmósfera disminuya a razón de  $6,5K$  por  $km$  de altura, siguiendo una relación lineal con la altura. Esta hipótesis también forma parte del modelado estandar realizado para la atmósfera.

Empezando con la explicación del modelado, nos tomamos un pequeño diferencial de aire de altura  $dz$  y cara lateral de un área  $A$  arbitraria, el cual se encuentra a una altura  $z$  de la forma que se muestra en la figura 2.3. Las fuerzas sobre este diferencial de aire serían: el propio peso del diferencial, sobre la cara inferior del mismo habría una presión  $P(z)$  generando una fuerza hacia arriba, y sobre la cara superior de dicho diferencial una presión  $P(z + dz)$  imponiendo una fuerza hacia abajo. Dado que estamos en condiciones de fluido-estática el diferencial no se mueve, en particular interesa que no tiene aceleración según la dirección que define la coordenada  $z$ .



$$\begin{aligned}
 P(z)A - P(z + dz)A - gdm &= 0 \\
 A[P(z + dz) - P(z)] &= AdP(z) = -gdm \\
 AdP(z) &= -gdm \\
 &= -g\rho(z)dV \\
 &= -g\rho(z)Adz
 \end{aligned}$$

$$\boxed{dP(z) = -g\rho(z)dz} \quad (2.13)$$

Figura 2.3: Modelado de la atmósfera.

La ecuación 2.13 es un resultado clásico de la fluido-estática. Para obtener el resultado final debemos sumarle a este resultado la ecuación derivadas del modelo de gas ideal (ecuación de estado) y la ecuación referida a la variación de la temperatura con la altura.

▷ **Modelo de gas ideal:**

$$\boxed{\frac{P(z)}{\rho(z)} = R_a T(z)} \quad (2.14)$$

▷ **Variación de la temperatura con la altura  $z$ :**

$$\boxed{T(z) = T_0 - \lambda z} \quad (2.15)$$

Donde  $\lambda$  es tal que,  $\lambda = 0,0065K/m$ . Sustituyendo  $\rho(z)$  de la ecuación 2.14 y  $T(z)$  de la ecuación 2.15, en la ecuación 2.13 e integrando se obtiene que:

$$\begin{aligned}
 \int_{P_0}^{P(z)} \frac{dP(z)}{P(z)} &= -\frac{g}{R_a} \int_0^z \frac{dz}{T_0 - \lambda z} = \frac{g}{\lambda R_a} \int_{T_0}^{T_0 - \lambda z} \frac{dq}{q} \\
 \Rightarrow \ln\left(\frac{P(z)}{P_0}\right) &= \frac{g}{\lambda R_a} \ln\left(\frac{T_0 - \lambda z}{T_0}\right)
 \end{aligned} \quad (2.16)$$

De ahora en mas al cociente  $\frac{g}{\lambda R_a}$  lo llamaremos  $\gamma$ . Cabe destacar que  $\gamma$  es un parámetro adimensionado. Para el valor de la gravedad terrestre  $g$  adoptamos en esta cuenta el valor estandar de dicha constante, el cual corresponde a  $g = 9,80665m/s^2$ .

$$\left. \begin{array}{l} g = 9,80665m/s^2 \\ \lambda = 0,0065K/m \\ R_a \approx 0,287kJ/kgK \end{array} \right\} \Rightarrow \gamma = \frac{g}{\lambda R_a} \approx 5,2568$$

Habiendo obtenido el valor de  $\gamma$ , podemos despejar de la ecuación 2.16 una expresión para la variación de la presión con la altura,  $P(z)$ , la cual se muestra en la ecuación 2.17.

$$P(z) = P_0 \left( \frac{T_0 - \lambda z}{T_0} \right)^\gamma \quad (2.17)$$

Y, utilizando la ecuación 2.14, se puede obtener para la variación de la densidad con la altura,  $\rho(z)$ , resultado que se muestra en la ecuación 2.18.

$$\rho(z) = \frac{P_0}{RT_0^\gamma} (T_0 - \lambda z)^{\gamma-1} \quad (2.18)$$

Las ecuaciones 2.15, 2.17 y 2.18 nos permiten hallar una buena estimación para la temperatura, la presión y la densidad del aire a una altura  $z$  genérica respectivamente <sup>f</sup>. Se utilizarán estas dependencias con las variable  $z$  para la construcción del simulador del lazo abierto (modelo *simulink*), es decir, para la construcción del bloque correspondiente al sistema. sin embargo, para el diseño del controlador se despreciará este efecto y se trabajará con valores constantes de estas propiedades termodinámicas a 1800m de altura, por esto interesa obtener los valores de éstas magnitudes a dicha altura.

- ▶  $P_{1800} \approx 81,476kPa$
- ▶  $T_{1800} \approx 276,45K$
- ▶  $\rho_{1800} = \rho \approx 1,0269kg/m^3$

De ahora en más nos referiremos al valor de  $\rho_z$  simplemente como  $\rho$ . Otro valor que será necesario obtener para utilizar posteriormente es el valor de la densidad del aire a nivel del mar  $\rho_0$ , el cual se muestra a continuación en la ecuación 2.19.

$$\rho_0 = \frac{P_0}{R_a T_0} \approx 1,2251kg/m^3 \quad (2.19)$$

El valor de  $\rho$  y de  $\rho_0$  serán importante a la hora de definir las fuerzas y momentos aerodinámicos y del motor. Otro valor que es importante de esta sección es el valor adjudicado a  $g$ .

## 2.3. Definición de los subconjuntos de variables

Es importante definir en este momento los subconjuntos de distintas variables a utilizar. Ya se definió en una sección anterior el vector de estados que se utilizará, el cual se muestra nuevamente en la ecuación 2.20. Por otro lado, interesa definir el vector de entradas  $\vec{u}(t)$  que representará el subconjunto de entradas al

<sup>f</sup>Siempre que no se excedan determinados límites, para alturas menores a 11000m las relaciones siguen siendo válidas. [10].

sistemas que podemos controlar mediante la computadora de a bordo, a saber, la deflexión de las superficies de vuelo y el giro en revoluciones por minuto de la hélice. Este vector se puede apreciar en la ecuación 2.21. Además, existen determinadas variables que representan entradas al sistema y que no podemos controlar como por ejemplo la velocidad del viento que expresaremos en el sistema SB. A este vector de entradas que no podemos controlar lo llamaremos  $\vec{h}(t)$  y se muestra en la ecuación 2.22. Finalmente, existen determinados parámetros del sistema dentro de los que se encuentran parámetros propios del avión y de la atmósferas, los cuales creemos conveniente agrupar en el vector  $\vec{p}$  de modo de hacer más compacta la nomenclatura. El vector  $\vec{p}$  se puede observar en la ecuación 2.23.

$$\vec{x} = (x, y, z, \varphi, \theta, \psi, v_{Bx}, v_{By}, v_{Bz}, w_{Bx}, w_{By}, w_{Bz}) \quad (2.20)$$

$$\vec{u} = (\delta_a, \delta_e, \delta_r, \text{delta}_f, n) \quad (2.21)$$

$$\vec{h} = (v_{vBx}, v_{vBy}, v_{vBz}) \quad (2.22)$$

$$\vec{p} = (P_0, T_{\text{cero}}, R, \lambda, g, \bar{c}, S, b) \quad (2.23)$$

De ahora en más en este trabajo, al hacer referencia al vector de estados  $x$  prescindiremos de la notación  $\vec{x}$ . Obraremos de manera similar al referirnos a los vectores  $\vec{u}$ ,  $\vec{h}$  y  $\vec{p}$  que se definieron anteriormente, a los cuales nos referiremos simplemente como  $u$ ,  $h$  y  $p$  respectivamente. También omitiremos la dependencia explícita con el tiempo ( $t$ ) de ahora en adelante.

## 2.4. Fuerzas y momentos involucrados

Como se mencionó en el párrafo anterior, en esta sección se abordará el tema de como son  $F_{Neta}^{\vec{}}$  y  $T_{Neto}^{\vec{}}$  para un avión en situación de vuelo. Las fuerzas involucradas en este modelo son el propio peso del avión  $\vec{P}$ , las fuerzas aerodinámicas  $\vec{F}_a$  que actúan sobre las superficies de vuelo, y la fuerza que ejerce el motor - o visto de otra manera, la hélice -  $\vec{F}_p$ . Éstas son las fuerzas que aportan a los términos  $F_{Neta}^{\vec{}}$  y  $T_{Neto}^{\vec{}}$  que aparecerán en las ecuaciones 2.10 y 2.12 y que dependen de las distintas variables que componen los vectores  $x$ ,  $u$ ,  $h$  y  $p$  definidos anteriormente. Cabe destacar que éste es un problema de mecánica de los fluidos de alta complejidad. Una pequeña reseña sobre mecánica de los fluidos se puede encontrar en el apéndice B de este trabajo. La forma más general para la  $F_{Neta}^{\vec{}}$  y  $T_{Neto}^{\vec{}}$  es la que se puede observar a continuación en las ecuaciones 2.24 y 2.25 respectivamente.

$$\vec{F}_{Neta}^{Ext} = \vec{F}_a(x, u, h, p) + \vec{F}_p(x, u, h, p) + \vec{P} \quad (2.24)$$

$$\vec{T}_{Neto}^{Ext} = \vec{T}_a(x, u, h, p) + \vec{T}_p(x, u, h, p) \quad (2.25)$$

En las subsecciones siguientes se explicará el modelado realizado para los términos que componen a  $\vec{F}_{Neta}^{\vec{}}$  y  $\vec{T}_{Neto}^{\vec{}}$ .

### 2.4.1. Peso del avión

Para comenzar el análisis de las fuerzas involucradas en el modelo empezaremos con la más sencilla de expresar y evaluar; el peso. Como se mostró párrafos anteriores nos referiremos a él como  $\vec{P}$ . Dado que nuestro análisis no considera curvatura alguna de la tierra ni variaciones de  $g$  con la altura, es claro que el peso del avión es de la forma que aparece en la ecuación 2.26.

$$\vec{P} = m\vec{g} = mg\hat{k}_I \quad (2.26)$$

Como ya hemos mencionado, es nuestro interés expresar las fuerzas en la base del SB. Para esto

podemos utilizar la matriz  $H_I^B$  que se definió en la ecuación 2.2 para rotar el versor  $\hat{k}_I$ . Haciendo esto se obtiene el resultado de la ecuación 2.27.

$$H_I^B \hat{k}_I = -\sin \theta \hat{i}_B + \cos \theta \sin \varphi \hat{j}_B + \cos \theta \cos \varphi \hat{k}_B \quad (2.27)$$

Y finalmente podemos obtener la expresión que utilizaremos para expresar el peso del avión, resultado que se muestra en la ecuación 2.28.

$$\vec{P}(\theta, \varphi, m, g) = -mg \sin \theta \hat{i}_B + mg \cos \theta \sin \varphi \hat{j}_B + mg \cos \theta \cos \varphi \hat{k}_B \quad (2.28)$$

Un comentario importante que cabe realizar en este punto es que el peso es una fuerza cuyo punto de aplicación es  $O$ , razón por la cual no ejerce torque alguno respecto a dicho punto. Como la ecuación 2.12 es obtenida de la segunda cardinal aplicada desde el punto  $O$ , el peso no tiene ninguna contribución a la ecuación de  $T_{Neto}$ .

## 2.4.2. Fuerzas y momentos aerodinámicos

Concentrándonos únicamente en los términos correspondientes a las fuerzas aerodinámicas,  $\vec{F}_a$  y  $\vec{T}_a$ , los mismos tienen la forma que se muestra en las ecuaciones 2.29 y 2.30 [2].

$$\vec{F}_a \propto \frac{\rho |\vec{v}_r|^2}{2} \vec{C}_F(x, u, h, p) \quad (2.29)$$

$$\vec{T}_a \propto \frac{\rho |\vec{v}_r|^2}{2} \vec{C}_T(x, u, h, p) \quad (2.30)$$

Es claro por motivos dimensionales que se debe multiplicar dicha expresión por un factor con dimensiones de superficie o de volumen para los distintos casos, según sea lo que se está calculando una fuerza o un momento. A su vez, las funciones  $\vec{C}_F$  y  $\vec{C}_T$  serán en cada caso distintas funciones de los elementos de los vectores  $x$ ,  $u$ ,  $h$  y  $p$ , definidos anteriormente en las ecuaciones 2.20, 2.21, 2.22 y 2.23.

Es importante destacar que la dependencia de las funciones  $\vec{C}_F$  y  $\vec{C}_T$  en dichas variables diferirá según el avión en consideración dependiendo mayormente de su geometría. Es común en la bibliografía encontrar como aproximación de esta dependencia un desarrollo de *Taylor*, y es ésta la formulación que se ha adoptado en este proyecto. Por supuesto, los coeficientes de la expansión polinomial (desarrollo de *Taylor*) dependerán del avión bajo consideración. A efectos de este proyecto, se asumió una expansión polinomial para un modelo de avión en particular que se explicará en párrafos siguientes. Cabe mencionar que el trabajo realizado durante este proyecto es independiente del avión escogido para su realización, pues es posible realizar un relevamiento de un avión de modo de obtener las magnitudes necesarias para poder aplicar los resultados que se obtuvieron.

Volviendo a un análisis más general de la fuerzas aerodinámicas, observamos que los vectores  $\vec{F}_a$  y  $\vec{T}_a$  se pueden escribir en términos de la base del SB de la forma que se muestra en las ecuaciones 2.31 y 2.32 respectivamente.

$$\vec{F}_a(x, u, h, p) = F_{ax} \hat{i}_B + F_{ay} \hat{j}_B + F_{az} \hat{k}_B \quad (2.31)$$

$$\vec{T}_a(x, u, h, p) = T_{ax} \hat{i}_B + T_{ay} \hat{j}_B + T_{az} \hat{k}_B \quad (2.32)$$

Para todas las fuerzas y torques aerodinámicos se debió adoptar un modelo, a saber, para las magnitudes  $F_{ax}$ ,  $F_{ay}$ ,  $F_{az}$ ,  $T_{ax}$ ,  $T_{ay}$  y  $T_{az}$ , que se definieron en las ecuaciones anteriores. El modelo para estas magnitudes se extrajo de la bibliografía [5][10]. Éste corresponde a expansiones polinomiales a las cuales ya hemos hecho referencia párrafos atrás, la idea básica es la de aproximar las funciones por polinomios de cierto

grado en la distintas variables pudiendo aceptar términos cruzados entre ellas.

Enfocándonos enteramente en como será el modelo para las fuerzas aerodinámicas, se muestra a continuación la forma que tendrán en nuestro modelado las magnitudes  $F_{ax}$ ,  $F_{ay}$ ,  $F_{az}$ ,  $T_{ax}$ ,  $T_{ay}$  y  $T_{az}$  en las ecuaciones 2.33, 2.34, 2.35, 2.36, 2.37 y 2.38 respectivamente. Como ya se ha mencionado en este trabajo, éstas dependen fuertemente de la velocidad del avión relativa al viento que hemos llamado  $\vec{v}_r$ , y que se define como  $\vec{v}_r = \vec{v} - \vec{v}_v$  siendo  $\vec{v}$  la velocidad lineal del avión y  $\vec{v}_v$  la velocidad del viento.

$$F_{ax} = \frac{\rho S \|\vec{v}_r\|^2}{2} C_{ax}(w_{By}, \vec{v}_r, \alpha, \delta_r, \delta_f, c) \quad (2.33)$$

$$F_{ay} = \frac{\rho S \|\vec{v}_r\|^2}{2} C_{ay}(w_{Bx}, w_{Bz}, \vec{v}_r, \beta, \delta_a, \delta_r, b) \quad (2.34)$$

$$F_{az} = \frac{\rho S \|\vec{v}_r\|^2}{2} C_{az}(w_{Bx}, \vec{v}_r, \alpha, \beta, \delta_e, \delta_f, c) \quad (2.35)$$

$$T_{ax} = \frac{\rho S b \|\vec{v}_r\|^2}{2} C_{al}(w_{Bx}, w_{Bz}, \vec{v}_r, \alpha, \beta, \delta_a, \delta_r, b) \quad (2.36)$$

$$T_{ay} = \frac{\rho S \bar{c} \|\vec{v}_r\|^2}{2} C_{am}(w_{Bx}, w_{Bz}, \vec{v}_r, \alpha, \beta, \delta_a, \delta_r, \delta_f, b) \quad (2.37)$$

$$T_{az} = \frac{\rho S b \|\vec{v}_r\|^2}{2} C_{an}(w_{Bx}, \vec{v}_r, \alpha, \beta, \delta_e, c) \quad (2.38)$$

Siendo  $C_{ax}$ ,  $C_{ay}$ ,  $C_{az}$ ,  $C_{al}$ ,  $C_{am}$  y  $C_{an}$  de la forma:

$$\begin{aligned} C_{ax} &= C_{xo} + C_{x\alpha}\alpha + C_{x\alpha^2}\alpha^2 + C_{x\alpha^3}\alpha^3 + C_{xw_{By}} \frac{cw_{By}}{\|\vec{v}_r\|} + C_{x\delta_r}\delta_r + C_{x\delta_f}\delta_f + C_{x\alpha\delta_f}\alpha\delta_f \\ C_{ay} &= C_{yo} + C_{y\beta}\beta + C_{yw_{Bx}} \frac{bw_{Bx}}{2\|\vec{v}_r\|} + C_{yw_{Bz}} \frac{bw_{Bz}}{2\|\vec{v}_r\|} + C_{y\delta_a}\delta_a + C_{y\delta_r}\delta_r + C_{y\alpha\delta_r}\alpha\delta_r \\ C_{az} &= C_{zo} + C_{z\alpha}\alpha + C_{z\alpha^3}\alpha^3 + C_{zw_{By}} \frac{cw_{By}}{\|\vec{v}_r\|} + C_{z\delta_e}\delta_e + C_{z\delta_e\beta^2}\beta^2\delta_e + C_{z\delta_f}\delta_f + C_{z\alpha\delta_f}\alpha\delta_f \\ C_{al} &= C_{lo} + C_{l\beta}\beta + C_{lw_{Bx}} \frac{bw_{Bx}}{2\|\vec{v}_r\|} + C_{lw_{Bz}} \frac{bw_{Bz}}{2\|\vec{v}_r\|} + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r + C_{l\alpha\delta_a}\alpha\delta_a \\ C_{am} &= C_{mo} + C_{m\alpha}\alpha + C_{m\alpha^2}\alpha^2 + C_{mw_{By}} \frac{cw_{By}}{\|\vec{v}_r\|} + C_{m\delta_e}\delta_e + C_{m\beta^2}\beta^2 + C_{mw_{Bz}} \frac{bw_{Bz}}{2\|\vec{v}_r\|} + C_{m\delta_f}\delta_f \\ C_{an} &= C_{no} + C_{n\beta}\beta + C_{nw_{Bx}} \frac{bw_{Bx}}{2\|\vec{v}_r\|} + C_{nw_{Bz}} \frac{bw_{Bz}}{2\|\vec{v}_r\|} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r + C_{nw_{By}} \frac{cw_{By}}{\|\vec{v}_r\|} + C_{n\beta^3}\beta^3 \end{aligned} \quad (2.39)$$

Donde se conocen los valores para los parámetros  $C_{xo}$ ,  $C_{x\alpha}$ ,  $C_{x\alpha^2}$ ,  $C_{x\alpha^3}$ ,  $C_{xw_{By}}$ ,  $C_{x\delta_r}$ ,  $C_{x\delta_f}$ ,  $C_{x\alpha\delta_f}$ ,  $C_{yo}$ ,  $C_{y\beta}$ ,  $C_{yw_{Bx}}$ ,  $C_{yw_{Bz}}$ ,  $C_{y\delta_a}$ ,  $C_{y\delta_r}$ ,  $C_{y\alpha\delta_r}$ ,  $C_{zo}$ ,  $C_{z\alpha}$ ,  $C_{z\alpha^3}$ ,  $C_{zw_{By}}$ ,  $C_{z\delta_e}$ ,  $C_{z\delta_e\beta^2}$ ,  $C_{z\delta_f}$ ,  $C_{z\alpha\delta_f}$ ,  $C_{lo}$ ,  $C_{l\beta}$ ,  $C_{lw_{Bx}}$ ,  $C_{lw_{Bz}}$ ,  $C_{l\delta_a}$ ,  $C_{l\delta_r}$ ,  $C_{l\alpha\delta_a}$ ,  $C_{mo}$ ,  $C_{m\alpha}$ ,  $C_{m\alpha^2}$ ,  $C_{mw_{By}}$ ,  $C_{m\delta_e}$ ,  $C_{m\beta^2}$ ,  $C_{mw_{Bz}}$ ,  $C_{m\delta_f}$ ,  $C_{no}$ ,  $C_{n\beta}$ ,  $C_{nw_{Bx}}$ ,  $C_{nw_{Bz}}$ ,  $C_{n\delta_a}$ ,  $C_{n\delta_r}$ ,  $C_{nw_{By}}$  y  $C_{n\beta^3}$  [5][10]. Es de notar que todos los coeficientes  $C$  son adimensionados.

### 2.4.3. Fuerzas y momentos del motor

Finalmente, debemos hacer un análisis similar pero para la fuerza y el torque del motor. En este caso se puede también escribir estos términos,  $\vec{F}_p$  y  $\vec{T}_p$ , en términos de la base del SB de la misma forma que se hizo en la subsección anterior como se muestra en las ecuaciones 2.31 y 2.32 respectivamente.

$$\vec{F}_p(x, u, h, p) = F_{px}\hat{i}_B + F_{py}\hat{j}_B + F_{pz}\hat{k}_B \quad (2.40)$$

$$\vec{T}_p(x, u, h, p) = T_{px}\hat{i}_B + T_{py}\hat{j}_B + T_{pz}\hat{k}_B \quad (2.41)$$

Nuevamente se debió asumir un modelo para las magnitudes  $F_{px}$ ,  $F_{py}$ ,  $F_{pz}$ ,  $T_{px}$ ,  $T_{py}$  y  $T_{pz}$ . Siendo

consistentes con el modelado obtenido para las fuerzas y momentos aerodinámicos, decidimos extraer el modelo para éstas otras fuerzas y momentos de la misma bibliografía [5][10]. Básicamente, queremos asegurarnos que el modelado entre la geometría del avión y su motor fuese consistente. Esta decisión hace que esto último necesariamente se cumpla dado que el modelado extraído se hizo en base a un avión existente. La forma que tendrán entonces las magnitudes de interés se muestran en las ecuaciones 2.42, 2.43, 2.44, 2.45, 2.46 y 2.47.

$$F_{px} = \frac{\rho S \|\vec{v}_r\|^2}{2} C_{px}(\alpha, \kappa) \quad (2.42)$$

$$F_{ay} = 0 \quad (2.43)$$

$$F_{az} = \frac{\rho S \|\vec{v}_r\|^2}{2} C_{pz}(\kappa) \quad (2.44)$$

$$T_{px} = \frac{\rho S b \|\vec{v}_r\|^2}{2} C_{pl}(\alpha, \kappa) \quad (2.45)$$

$$T_{ay} = \frac{\rho S \bar{c} \|\vec{v}_r\|^2}{2} C_{pm}(\kappa) \quad (2.46)$$

$$T_{az} = \frac{\rho S b \|\vec{v}_r\|^2}{2} C_{pn}(\kappa) \quad (2.47)$$

Siendo  $C_{px}$ ,  $C_{pz}$ ,  $C_{pl}$ ,  $C_{pm}$  y  $C_{pn}$  de la forma:

$$\begin{aligned} C_{px} &= C_{x\kappa}\kappa + C_{x\alpha\kappa^2}\alpha\kappa^2 \\ C_{pz} &= C_{z\kappa}\kappa \\ C_{pl} &= C_{l\kappa\alpha^2}\kappa\alpha^2 \\ C_{pm} &= C_{m\kappa}\kappa \\ C_{pn} &= C_{n\kappa^3}\kappa^3 \end{aligned} \quad (2.48)$$

Donde se conocen los valores para los parámetros  $C_{x\kappa}$ ,  $C_{x\alpha\kappa^2}$ ,  $C_{z\kappa}$ ,  $C_{l\kappa\alpha^2}$ ,  $C_{m\kappa}$  y  $C_{n\kappa^3}$  [5][10]. Nuevamente, es de notar que todos los coeficientes  $C$  son adimensionados.

Como se puede apreciar, las fuerzas y momentos del motor depender fuertemente del parámetro  $\kappa$  el cual se define en la ecuación 2.49.

$$\kappa(n, \vec{v}_r, \rho) = C_1 + C_2 \left[ \frac{a + b\left(\frac{n}{n_0} + 1\right) + (cn + d)\left(1 - \frac{\rho}{\rho_0}\right)}{\rho \|\vec{v}_r\|^3 / 2} \right] \quad (2.49)$$

Donde  $\kappa$  es un parámetro adimensionado el cual es directamente proporcional a la potencia del motor y se obtiene observando el cambio de presión total debido al efecto de la hélice. Es de observar que este modelo es válido para un motor de pistones con el cual se mueve una hélice, lo cual representa una situación igual al avión que disponemos. [5].

De esta forma quedan descriptas entonces las relaciones funcionales que utilizaremos para las fuerzas y momentos aerodinámicos y del motor en el modelado. Éstas corresponden al modelo de avión que estamos utilizando, el *BEAVER* y están sujetas a los comentarios realizados en la sección 2.3.

#### 2.4.4. Resumen de coeficientes

Siguiendo en la misma línea de la subsección anterior, expondremos en esta los valores para los coeficientes necesarios para definir las expansiones polinomiales correspondientes a las fuerzas y momentos aerodinámicos y del motor, los cuales se pueden observar en las tablas 2.1 y 2.2.

<b>Coefficientes para las fuerzas aerodinámicas</b>			
$F_{ax}$	$C_{x_0} = -0,03554$ $C_{x\alpha^3} = -5,162$ $C_{x\delta_f} = -0,09447$	$C_{x\alpha} = 0,00292$ $C_{xw_{By}} = -0,6748$ $C_{x\alpha\delta_f} = 1,106$	$C_{x\alpha^2} = 5,459$ $C_{x\delta_r} = 0,03412$
$F_{ay}$	$C_{y_0} = -0,002226$ $C_{yw_{Bz}} = 0,3666$ $C_{y\alpha\delta_r} = 0,5238$	$C_{x\beta} = -0,7678$ $C_{y\delta_a} = -0,02956$	$C_{yw_{Bx}} = -0,124$ $C_{y\delta_r} = 0,1158$
$F_{az}$	$C_{z_0} = -0,05504$ $C_{zw_{By}} = -2,988$ $C_{z\delta_f} = -1,377$	$C_{z\alpha} = -5,578$ $C_{z\delta_e} = -0,398$ $C_{z\alpha\delta_f} = -1,261$	$C_{z\alpha^3} = 3,442$ $C_{z\delta_e\beta^2} = -15,93$
<b>Coefficientes para los momentos aerodinámicos</b>			
$T_{ax}$	$C_{l_0} = 0,000591$ $C_{lw_{Bz}} = 0,1695$ $C_{l\alpha\delta_a} = -0,08269$	$C_{l\beta} = 0,0618$ $C_{l\delta_a} = -0,09917$	$C_{lw_{Bx}} = -0,5045$ $C_{l\delta_r} = 0,006934$
$T_{ay}$	$C_{m_0} = 0,09448$ $C_{mw_{By}} = -15,56$ $C_{mw_{Bz}} = -0,3118$	$C_{m\alpha} = -0,6028$ $C_{m\delta_e} = -1,921$ $C_{m\delta_f} = 0,4072$	$C_{m\alpha^2} = -2,14$ $C_{m\beta^2} = 0,6921$
$T_{az}$	$C_{n_0} = -0,003117$ $C_{nw_{Bz}} = -0,1112$ $C_{nw_{By}} = 0,1595$	$C_{n\beta} = 0,006719$ $C_{n\delta_a} = -0,003872$ $C_{n\beta^3} = 0,1373$	$C_{nw_{Bx}} = -0,1585$ $C_{n\delta_r} = -0,08265$

Cuadro 2.1: Tabla de coeficientes para las fuerzas y momentos aerodinámicos. [5][10].

<b>Coefficientes para las fuerzas y momentos del motor</b>		
$\kappa$	$C_1 = 0,08696$ $a \simeq -240,14kW$ $c \simeq -0,071kW/rpm$ $n_0 = 2010rpm$	$C_2 = 191,18kg/kWs^3$ $b \simeq 203,43kW$ $d \simeq 300,08kW$
$F_{px}$	$C_{x\kappa} = 0,1161$	$C_{x\alpha\kappa^2} = 0,1453$
$F_{pz}$	$C_{z\kappa} = -0,1563$	
$T_{pl}$	$C_{l\kappa\alpha^2} = -0,01406$	
$T_{pm}$	$C_{m\kappa} = -0,07895$	
$T_{pn}$	$C_{n\kappa^3} = -0,003026$	

Cuadro 2.2: Tabla de coeficientes para las fuerzas y momentos del motor. [5][10].

## 2.4.5. Comentarios sobre el modelado de las fuerzas y momentos

Para finalizar con esta descripción de las fuerzas, expondremos a continuación algunos comentarios respecto al modelo que utilizaremos para las mismas que son de interés en esta etapa de modelado. Esto es a modo de chequeo del modelo, dado que debemos observar la correcta correspondencia entre las ecuaciones que son utilizadas en el MVE y el sistema físico a modelar.

Si observamos las expresiones para las fuerzas y momentos aerodinámicos, podemos notar que éstas tiene una fuerte dependencia de los ángulos  $\alpha$  y  $\beta$ , y las variables de control  $\delta_a$ ,  $\delta_e$  y  $\delta_r$ . Un primer elemento a chequear es que los sentidos positivos utilizados para las variables de control  $\delta$ , que se pueden observar en la figura 1.5, sean consistentes con los sentidos positivos definidos para los momentos aerodinámicos  $T_{ax}$ ,  $T_{ay}$

y  $T_{az}$ , el momento de *roll*, el momento de *pitch* y el momento de *yaw* respectivamente, que están definidos sus sentidos positivos según la regla de mano derecha utilizando la base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$  del SB, así como con los sentidos las fuerzas aerodinámicas definidos positivos según los versores de dicha base. En la figura 1.2 se puede apreciar el sentido positivo para estos momentos. Otros elementos que analizamos de este modelos son: la variación del coeficiente de *lift* en función del ángulo  $\alpha$  y la relación típica entre dicho coeficiente de *lift* y el coeficiente de *drag*, de forma de compararlos contra los resultados teóricos ideales de la bibliografía.

Si observamos la figura 1.5, intuitivamente podemos notar que una variación positiva del ángulo del alerón ( $\delta_a > 0$ ) implica la generación de un momento según  $-x_B$ , es decir en el sentido opuesto al definido positivo para el momento  $T_{ax}$ . En menor medida, aparecería un momento según la dirección  $-k_B$ . En cuanto a su efecto en las fuerzas aerodinámicas, las fuerzas que aparecen en los alerones debido a este movimiento son tales que generan momentos según lo comentado en las oraciones anteriores pero a efectos de la primera cardinal se anulan, con lo cual sería de esperar que debido a esta simetría (y el movimiento conjugado) no aparecieran variaciones en las componentes  $F_{ax}$  y  $F_{az}$ . Lo dicho anteriormente se refleja en el modelo que se obtuvo. El movimiento de los alerones no afectan en las componentes de la fuerza aerodinámica  $F_{ax}$  y  $F_{az}$ , si bien aparece una pequeña componente según  $F_{ay}$ . En cuanto a los momentos, también es modelo es compatible pues los coeficientes  $C_{l\delta_a}$ ,  $C_{l\alpha\delta_a}$  y  $C_{m\delta_a}$  son negativos, reflejando así lo referido respecto a los sentidos escogidos positivos.

Si analizamos ahora el movimiento del elevador, siguiendo con razonamientos similares a los aplicados en el párrafo anterior una variación positiva de  $\delta_e$  implicaría una aparición de una fuerza hacia arriba según el eje vertical del avión (según  $-k_B$ ), esto implica una aparición de un momento según  $-j_B$ , es decir, ambas contribuciones (a fuerzas y momentos) serían en el sentido opuesto al definido positivo para dichas fuerzas y momentos. Esto también es consistente con las ecuaciones del modelado dado que los coeficientes  $C_{z\delta_e}$  y  $C_{m\delta_e}$  son en ambos casos negativos.

Por otro lado, ante una variación de los *flaps* (positiva por definición) debería aparecer una fuerza según  $-k_B$  (los *flaps* se utilizan para aumentar el *lift*), también es de esperar una pequeña fuerza según  $-i_B$ . Éstas dos conjeturas también se verifican dado que los coeficientes  $C_{z\delta_f}$  y  $C_{z\alpha\delta_f}$  son ambos negativos, y  $C_{x\delta_f}$  es negativo y pequeño.

Finalmente, analizando una variación positiva en el movimiento del timón del cola ( $\delta_r > 0$ ), podríamos esperar que aparezca una fuerza según  $j_B$  con su correspondiente momento respecto al versor  $-k_B$ . Nuevamente, en este caso, esto se contraste en el MVE dado que el coeficiente  $C_{y\delta_r}$  es positivo y el coeficiente  $C_{n\delta_r}$  es negativo. Además en el modelo aparece una componente debido al movimiento del timon en la componente  $F_{ax}$  de la fuerza aerodinámica.

Para realizar las comparaciones del modelo obtenido para las fuerzas y momentos aerodinámicos con el coeficiente de *lift*, que denotaremos con  $C_L$  de ahora en más, y el coeficiente de *drag*, que llamaremos  $C_D$  de ahora en adelante, debemos primero definir estos coeficientes.

Los coeficientes  $C_L$  y  $C_D$  se definen en base a un sistema definido por el vector  $\vec{v}_r$ . Este vector  $\vec{v}_r = \vec{v} - \vec{v}_v$  de velocidad del avión relativa al viento define un nuevo sistema de referencia al que denotaremos con la sigla SR<sup>§</sup>, el cual tendrá una base de medida  $(\hat{i}_R, \hat{j}_R, \hat{k}_R)$ . El *lift*  $L$  y el *drag*  $D$  están definidos en los sentidos definidos por la base del sistema SR, esto es así pues éstas son las fuerzas que aparecen sobre el avión cuando se lo deja quieto y se considera como que es el aire que se mueve a una cierta velocidad  $\vec{v}_r$ , lo cual representa una situación análoga a la de un avión en situación de vuelo pero es más sencilla de analizar desde el punto de vista de la mecánica de los fluidos. Siguiendo con este análisis se puede encontrar que el vector fuerza de *lift*  $\vec{L}$  y el vector fuerza de *drag*  $\vec{D}$  son de la forma [2] [14]:

$$\vec{L} = \frac{\rho S |\vec{v}_r|^2}{2} C_L (-\hat{k}_R) \quad (2.50)$$

$$\vec{D} = \frac{\rho S |\vec{v}_r|^2}{2} C_D (-\hat{i}_R) \quad (2.51)$$

Dado que disponemos de la forma de las fuerzas expresadas en la base del sistema SB ( $F_{ax}$ ,  $F_{ay}$

<sup>§</sup>En la bibliografía es sistema de referencia llamado *stability system* y se denota según la sigla SS.

y  $F_{az}$ ), debemos encontrar el cambio de base que lleva la base del sistema SB en la base del sistema SR. Procediendo de manera análoga a la que se expuso en la sección 3.1, para ir del sistema SR al sistema SB se debe efectuar primero una rotación ángulo de  $-\beta$  respecto al versor  $\hat{k}_R$  lo cual define un set de versores que llamaremos  $(\hat{i}_1, \hat{j}_1, \hat{k}_R)$ . Luego se debe realizar una rotación de ángulo  $-\alpha$  respecto al versor  $\hat{j}_1$  - que definió la rotación anterior - para finalmente llegar a la base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$ , culminando así el procedimiento para el cambio de base. Este procedimiento se puede apreciar más claramente ayudado por la figura 1.6. Matricialmente lo explicado se puede sintetizar según la ecuación 2.52 que se muestra a continuación:

$$\begin{pmatrix} \hat{i}_B \\ \hat{j}_B \\ \hat{k}_B \end{pmatrix} = \begin{pmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{pmatrix} \begin{pmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{i}_R \\ \hat{j}_R \\ \hat{k}_R \end{pmatrix} \quad (2.52)$$

Con lo cual la matriz de cambio de base  $H_R^B$  que se obtiene queda de la manera que se puede observar en la ecuación ??.

$$H_R^B = \begin{pmatrix} \cos \alpha \cos \beta & -\sin \beta \cos \alpha & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \beta \sin \alpha & \cos \alpha \end{pmatrix} \quad (2.53)$$

Dado que lo que realmente nos interesa es la matriz de rotación  $H_B^R$ , procederemos análogamente a lo explicado en la sección 3.1 respecto a cambiar los sin por  $-\sin$  e invertir el orden de las rotaciones para hallar dicha matriz. Con lo cual el resultado final para  $H_B^R$  es el que se muestra en la siguiente ecuación, la ecuación 2.54.

$$H_B^R = \begin{pmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix} \quad (2.54)$$

Utilizando esta matriz es que podemos hallar la relación entre las fuerzas  $\vec{L}$  y  $\vec{D}$  con las componentes de la fuerzas expresadas en la base del SB  $F_{ax}$ ,  $F_{ay}$  y  $F_{az}$ . Realizando esta transformación para el caso particular de  $\beta = 0$ , pues queremos comparar contra las variaciones teóricas con que varían  $C_L$  y  $C_D$  como función solamente de  $\alpha$ , se obtiene que:

$$\vec{L} = (F_{ax} \cos \alpha + F_{az} \sin \alpha)(-\hat{k}_R) \quad (2.55)$$

$$\vec{D} = (-F_{ax} \sin \alpha + F_{az} \cos \alpha)(-\hat{i}_R) \quad (2.56)$$

Observando que según las ecuaciones 2.50, 2.51, 2.33 y 2.35 las fuerzas  $\vec{L}$ ,  $\vec{D}$ ,  $F_{ax}$  y  $F_{az}$  comparten el término  $\frac{\rho S |\vec{v}_r|^2}{2}$ , se puede obtener finalmente las ecuaciones que vinculan los coeficientes  $C_L$  y  $C_D$  con  $F_{ax}$  y  $F_{az}$ . Estas relaciones son las que se muestran a continuación en las ecuaciones 2.57 y 2.58.

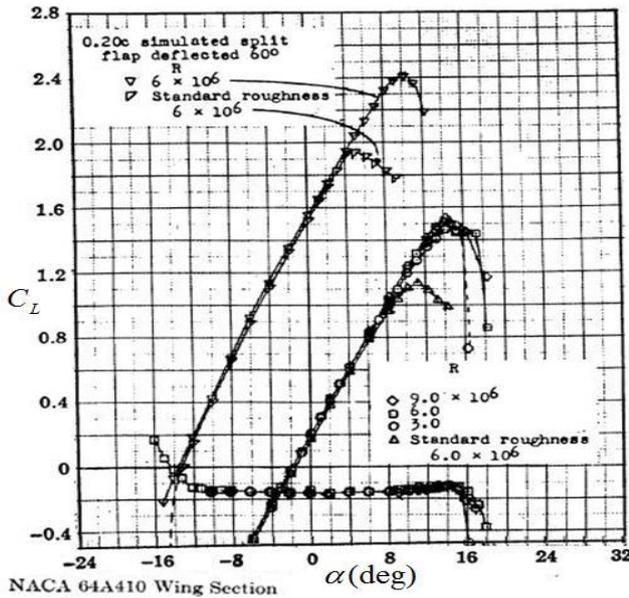
$$C_L = -(-C_x \sin \alpha + C_z \cos \alpha) \quad (2.57)$$

$$C_D = -(C_x \cos \alpha + C_z \sin \alpha) \quad (2.58)$$

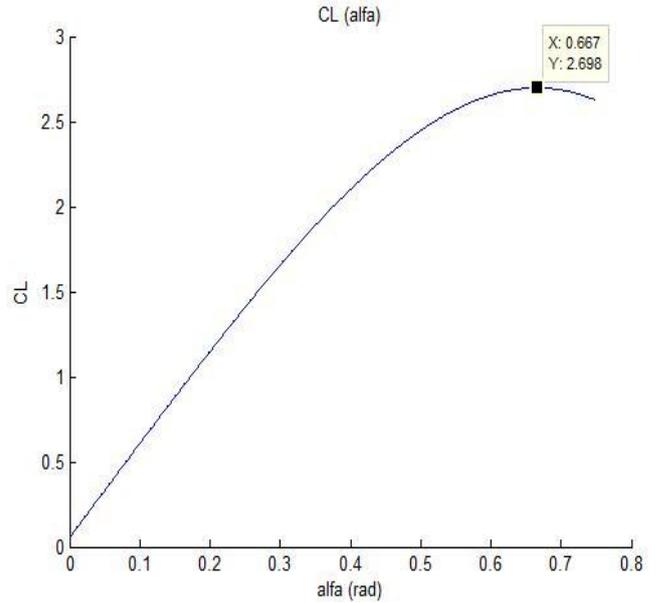
Según éstas dos últimas ecuaciones tenemos una expresión para los coeficientes  $C_L$  y  $C_D$  como función de los parámetros  $x$ ,  $u$ ,  $h$  y  $p$ . No interesa en particular analizar el comportamiento de estas magnitudes como función del ángulo  $\alpha$ , pues disponemos de ciertas gráficas teóricas ideales para el comportamiento funcional de las mismas. En lo que sigue, trabajaremos sólo con la dependencia en  $\alpha$  tomando todas las

variables como constantes o directamente nulas (como por ejemplo para los gráficos que se mostraran a continuación). Es claro que poner los valores de esas variables como constante sólo subiría o bajaría los gráficos, no afectaría de gran manera la forma general de dichos gráficos.

A continuación se muestran en las figuras 2.4 y 2.5 los gráficos obtenidos para el modelo utilizado y los gráficos teóricos para un perfil NACA 64 A 410 [14] (el perfil de ala del *BEAVER* es el NACA 64 A 416). El primer gráfico corresponde a la evolución del coeficiente de *lift* con función del ángulo de ataque ( $C_L(\alpha)$ ), mientras que el segundo es un gráfico comparativo de la evolución del coeficiente de *drag* con la evolución del coeficiente de *lift*, es decir, un gráfico de  $C_D(C_L)$ .

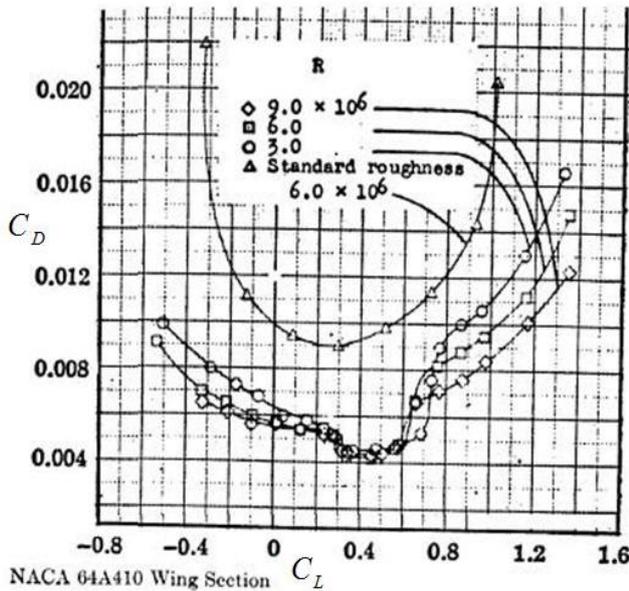


(a)  $C_L(\alpha)$  para el perfil NACA 64 A 410.

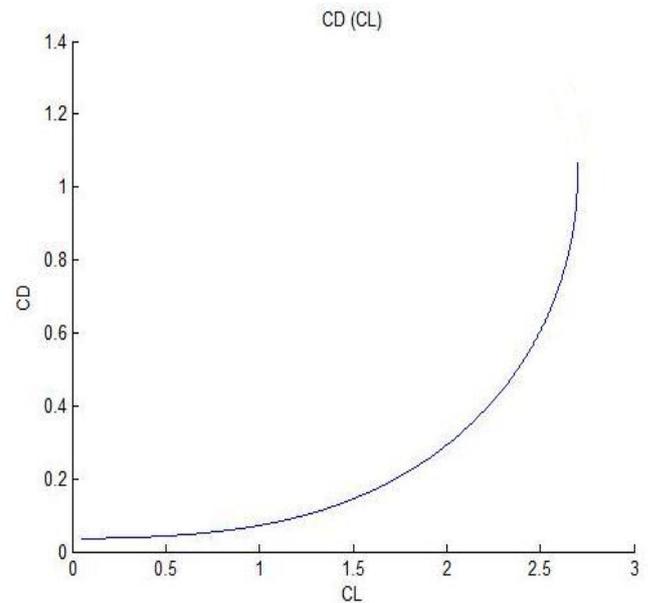


(b)  $C_L(\alpha)$  para el modelo del *BEAVER*.

Figura 2.4: Relación funcional entre  $C_L$  y  $\alpha$  para el modelo del *BEAVER* y el perfil NACA 64 A 410.



(a)  $C_D(C_L)$  para el perfil NACA 64 A 410.



(b)  $C_D(C_L)$  para el modelo del *BEAVER*.

Figura 2.5: Relación entre  $C_D$  y  $C_L$  para el modelo del *BEAVER* y el perfil NACA 64 A 410.

Como se desprende de estas últimas imágenes mostradas, las dependencias funcionales observadas para el modelo obtenido para el *BEAVER* respetan la forma de las funciones teóricas en ambos casos. Evidentemente, varían algunos valores respecto a los teóricos dado que se trata de un relevamiento de un avión existente en la realidad, a saber, las posibles diferencias vienen dadas por: efecto de ala finita, cierta rugosidad de la superficie, perfil no homogéneo en el ala, alguna modificación del perfil estandar para obtener mejores resultados (por ejemplo variando el ángulo de diedro), y también el hecho de que el perfil NACA 64 A 410 no es exactamente el mismo que el del *BEAVER*, aunque en todos los distintos perfiles de ala se observan gráficas muy parecidas entre si, con la misma forma y variando levemente algunos aspectos cuantitativos. En ambos gráficos de la figura 2.4 se tiene la misma forma, se tiene que  $C_L$  crece al crecer  $\alpha$  hasta que se llega a un cierto ángulo  $\alpha_S$  donde se encuentra un máximo de  $C_L$ , luego si se sigue aumentando  $\alpha$  el avión entra en pérdida<sup>h</sup> y el coeficiente de *lift* empieza a disminuir conforme siga aumentando  $\alpha$ . Este ángulo  $\alpha_S$  es el denominado ángulo de *stall* y es una magnitud bien importante en el vuelo de un avión. En ambos gráficos, el correspondiente al modelo y a la gráfica teórica, se puede apreciar dicho fenómeno de *stall*. Observando ahora cualitativamente los gráficos de la figura 2.5, se puede observar que sobre el semiplano  $C_L > 0, C_D > 0$  la forma de ambas gráficas es esencialmente la misma, es decir, ambas tienen forma parabólica.

Respecto a los aspectos cuantitativos de esta comparación, es de destacar que los ordenes de magnitud en los ejes de ambas figuras son muy parecidos. Para ejemplificar esto, observando las gráficas correspondientes a  $C_L(\alpha)$  de la figura 2.4, en la gráfica teórica se puede apreciar un coeficiente de *lift* máximo de  $\approx 2,4$ , mientras que para el modelo del *BEAVER* el coeficiente de *lift* máximo es  $\approx 2,7$ . Por otro lado, si observamos la figura 2.5 se puede apreciar que el coeficiente  $C_D$  crece más rápido para la gráfica del modelo del *BEAVER* que para el modelo teórico, es decir, para un coeficiente de *lift* de  $C_L \approx 0,8$  se tiene para el modelo teórico un coeficiente de *drag* de  $C_{D_{teorico}} \approx 0,012$  mientras que para el modelo del *BEAVER* el mismo es de  $C_{D_{BEAVER}} \approx 0,06$ , lo cual representa un crecimiento más rápido de la parábola aunque los órdenes de magnitud de los resultados son similares. Este fenómeno es en cierta medida razonable dado que es esperable tener un *drag* mayor en la realidad que en el modelo teórico. En lo que respecta al ángulo de *stall*  $\alpha_S$ , podemos ver que para el caso del modelo teórico ideal dicho ángulo se encuentra en el entorno de los  $\approx 15^\circ$  como máximo, en cambio para nuestro avión modelo el mismo se encuentra en los  $\approx 38^\circ$  lo cual representa una distancia importante respecto al marco teórico.

## 2.5. Resumen de efectos no considerados en el modelado

Como todo modelado, el modelo realizado está sujeto a errores ya sean por errores intrínsecos a las mediciones necesarias para la determinación del modelo, así como por efectos que no fueron tomados en cuenta en la elaboración del mismo. El saber qué efectos son relevantes y cuáles pueden ser considerados como perturbación al sistema es donde entra en juego el arte del modelado. En esta sección haremos un listado de los fenómenos existentes y que no fueron considerados a la hora de la obtención del MVE, con una breve explicación de por qué se considera que puede ser despreciable su efecto, y razón por la cual dicho efecto será incluido como una perturbación al sistema, problema a ser resuelto por la lógica de control.

### ▷ Superficie terrestre no plana.

En todo momento se consideró en la obtención de la ecuaciones de movimiento que la tierra era plana. Por ejemplo, en nuestro modelo una trayectoria recta en el plano  $xy$  con  $z = cte$  es una trayectoria en línea recta, pero en la realidad tener  $z = cte$  es en verdad una curva que tiene curvatura igual a la curvatura de la tierra. El avión del que se dispone cuenta con una autonomía de vuelo de media hora aproximadamente, suponiendo una velocidad de  $70m/s$ , este avión sería capaz de alejarse a lo sumo a una distancia de  $126km$ . Es claro que en comparación al perímetro y la curvatura terrestre estos  $126km$  son completamente despreciables. Este efecto debe ser considerado a efectos de vuelos de alta duración donde se recorran largas distancias.

<sup>h</sup>Se empiezan a apreciar turbulencias sobre la parte superior trasera de las alas.

Incluso en este caso no tendría porque ser considerado en el modelado del sistema, dado que las correcciones a realizar podrían solucionarse por el sistema de control, más específicamente, lo que llamaremos el segundo lazo de control.

#### ▷ **Variación de la aceleración $g$ .**

La aceleración gravitatoria  $g$  varía en efecto con la altura  $z$ , además se debe hacer notar que dicha aceleración varía también con la latitud y la longitud que uno se encuentre. Estos efectos no fueron considerados ni en el cálculo del peso propiamente dicho del avión ni en el modelado de la atmósfera. El avión del que se dispone está pensado para volar a una altura de  $100m$ . El modelo del avión que se utilizó en este proyecto fue utilizado como altura de régimen el valor de  $1800m$ . En ambos casos las alturas involucradas son despreciables frente al radio terrestre de  $\approx 6370km$ , lo cual hace despreciable el efecto de la variación de  $g$  para dichas alturas de vuelo. En cuanto al cambio del valor de  $g$  con la latitud y longitud, el mismo no es significativo respecto al valor usado para  $g$ .

#### ▷ **Masa del sistema no constante.**

Este es un efecto que si bien hicimos referencia en el modelado, asumimos en dicho momento que sería despreciable. En efecto, lo es. La variación de masa en el sistema viene por el hecho de que el avión para volar debe consumir combustible, la pérdida total de masa durante el transcurso de una sesión de vuelo es a lo sumo igual a la cantidad de masa del tanque de combustible lleno.

El avión del que se dispone tiene un tanque de combustible de  $750ml$  y se recomienda utilizar para su funcionamiento una mezcla de nafta super con aceite lubricante en relación 40/1 respectivamente, dado estas proporciones, la densidad del combustible será esencialmente igual que la densidad de la nafta super, la cual es de  $\rho_{NS} \approx 0,745kg/l$ . En este caso tenemos una masa total de combustible para el avión de  $m_c \approx 558,75g$ , dicha pérdida de masa es completamente despreciable frente a la masa total del avión que es de  $12kg$ . Para el caso del *BEAVER* esta diferencia no es tan grande, el mismo tiene un tanque de combustible de  $512l$ , asumiendo el mismo tipo de combustible<sup>i</sup>, se tiene que la pérdida de masa total para el avión que se utiliza como modelo es de  $m_c \approx 381,44kg$ , de la misma manera esta pérdida de masa fue despreciada frente a la masa total del *BEAVER* con combustible de  $\approx 2669,671kg$  en este modelado. Despreciar esta pérdida de masa es práctica estandar en el modelado del vuelo de aviones.

Es claro que despreciar este término es siempre válido en recorridos cortos. Si el error por hacer esta aproximación fuese demasiado grande se podría ir actualizando el modelo cada determinada cantidad de tiempo (o de longitud recorrida) y considerarlo válido para un rango de tiempo (o de distancia a recorrer) determinado. Este tipo de estrategias no se consideraron en el transcurso de este proyecto pues esta aproximación no representa una complicación mayor, las trayectorias recorridas son reajustadas por la lógica de control.

#### ▷ **Variación de la temperatura de forma más compleja.**

En el modelado de la atmósfera se asumió que la temperatura decrecía linealmente con la altura  $z$ . Es claro que la temperatura de la atmósfera no es únicamente función de la variable  $z$ , depende también de muchos otros aspectos, por ejemplo velocidad del viento, latitud y longitud, constitución atmosférica, etc. Dado las alturas a las que estamos trabajando se considera este modelo como aceptable. De todas formas se considera un modelo estandar de la atmósferas para alturas menores a  $11000m$ , y en este proyecto se trabajará a altura mucho menores.

---

<sup>i</sup>Utilizando nafta común no hace mucha diferencia la densidad pues  $\rho_{NC} \approx 0,725kg/l$ , respecto a los órdenes de magnitud del cálculo no existe diferencia alguna.

## 2.6. Modelo en variables de estado completo

Por último, y para finalizar este capítulo, expondremos el MVE completo a modo de resumen. Éste es el modelo con el cual trabajaremos de aquí en más en el desarrollo de esta monografía. El MVE queda entonces de la siguiente manera:

$$\begin{aligned}
 \dot{x} &= \cos \theta \cos \psi v_{Bx} + (\sin \theta \cos \psi \sin \varphi - \sin \psi \cos \varphi) v_{By} + (\sin \theta \cos \psi \cos \varphi + \sin \psi \sin \varphi) v_{Bz} \\
 \dot{y} &= \cos \theta \sin \psi v_{Bx} + (\sin \theta \sin \psi \sin \varphi + \cos \varphi \cos \psi) v_{By} + (\sin \theta \sin \psi \cos \varphi - \sin \varphi \cos \psi) v_{Bz} \\
 -\dot{z} &= -\sin \theta v_{Bx} + \cos \theta \sin \varphi v_{By} + \cos \theta \cos \varphi v_{Bz} \\
 \dot{\varphi} &= w_{Bx} + \sin \varphi \tan \theta w_{By} + \cos \varphi \tan \theta w_{Bz} \\
 \dot{\theta} &= \cos \varphi w_{By} - \sin \varphi w_{Bz} \\
 \dot{\psi} &= (\sin \varphi / \cos \theta) w_{By} + (\cos \varphi / \cos \theta) w_{Bz} \\
 v_{\dot{B}x} &= \frac{1}{m} [F_{xa} + F_{xp} - mg \sin \theta] - v_{Bz} w_{By} + v_{By} w_{Bz} \\
 v_{\dot{B}y} &= \frac{1}{m} [F_{ya} + F_{yp} + mg \cos \theta \sin \varphi] - v_{Bx} w_{Bz} + v_{Bz} w_{Bx} \\
 v_{\dot{B}z} &= \frac{1}{m} [F_{za} + F_{zp} + mg \cos \theta \cos \varphi] - v_{By} w_{Bx} + v_{Bx} w_{By} \\
 \begin{pmatrix} w_{\dot{B}x} \\ w_{\dot{B}y} \\ w_{\dot{B}z} \end{pmatrix} &= I_{OB}^{-1} \left[ \begin{pmatrix} T_{ax} \\ T_{ay} \\ T_{az} \end{pmatrix} + \begin{pmatrix} T_{px} \\ T_{py} \\ T_{pz} \end{pmatrix} - \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times I_{OB} \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \right]
 \end{aligned}$$

(2.59)

Donde las magnitudes  $F_{xa}$ ,  $F_{za}$ ,  $T_{ya}$ ,  $F_{xp}$ ,  $F_{zp}$  y  $T_{yp}$  son funciones de los subconjuntos de variables y parámetros  $x$ ,  $u$ ,  $p$  y  $h$ . Éstas relaciones funcionales son las que se muestran en las ecuaciones 2.33, 2.34, 2.35, 2.36, 2.37, 2.38, 2.42, 2.43, 2.44, 2.45, 2.46 y 2.47 respectivamente.

La matriz  $I_{OB}$  es la que se muestra en la ecuación 1.5, cuya inversa, la matriz  $I_{OB}^{-1}$  se muestra en la ecuación 1.6. Los demás parámetros de este MVE han sido definidos durante el transcurso de este capítulo en las distintas subsecciones.

Con esto damos fin a la etapa de modelo, y procederemos de ahora en más a trabajar sobre los otros aspectos que fueron analizados durante el transcurso de este proyecto.

## Simulación del lazo abierto

Luego de obtenido el MVE, se procedió a la construcción de un simulador. La principal función del simulador consiste en la resolución numérica de las ecuaciones del MVE para distintos valores de las condiciones iniciales y las entradas al sistema  $u$  y  $h$  definidas por la ecuaciones 2.21 y 2.22 respectivamente <sup>a</sup>. De esta manera, este simulador representa un modelo del comportamiento del sistema, es decir, lo que llamaremos el lazo abierto (sistema sin controlador).

Este simulador es esencial a fin de poder observar el comportamiento del sistema en lazo abierto, o sea, realizar distintas pruebas sobre la dinámica del sistema para distintos valores de las acciones  $u$  de control y para diferentes valores de las entradas no controlables  $h$  (distintas velocidades para el viento). También será importante a la hora de la validación del controlador a diseñar, es decir, para constatar el correcto funcionamiento del lazo cerrado. Este modelo del sistema será además muy útil para el propio diseño del controlador como se explicará en el capítulo 6. Por todas éstas utilidades, es fundamental que dicho simulador refleje de la manera más fiel al sistema real, y en particular que las trayectorias generadas por este simulador sean confiables, razón por la cual el simulador debió ser testeado exhaustivamente desde ejemplos sencillos hasta la realización de pruebas más complicadas. Dado que fue utilizado en otras etapas del proyecto, este *software* fue también probado indirectamente a medida que transcurrían las demás etapas del mismo. Cabe hacer notar que muchas de las pruebas realizadas (de las cuales sólo se muestra en este informe un grupo reducido) pueden escapar un poco a la intuición en el sentido "¿cuál sería la trayectoria razonable para el vuelo del avión dado el valor de los parámetros, las entradas, las condiciones iniciales, y las fuerzas y los momentos aplicados sobre el rígido?". Esta complicación nace básicamente por la complejidad de las fuerzas y momentos aerodinámicos y del motor, por poner un ejemplo, un movimiento en el timón de cola  $\delta_r$  afecta las componentes de la fuerza aerodinámica  $F_{ax}$  y  $F_{ay}$ , y también las componentes del momento aerodinámico  $T_{ax}$  y  $T_{az}$ .

Pero no sólo se debe tener en cuenta el hecho de que las trayectorias predichas por el simulador deben reflejar lo más fielmente la realidad física del sistema, sino que también hay que cuidarse de los errores numéricos. Estos errores numéricos deben ser tomados en cuenta de forma crítica, y sino se cuidan puede llegar a haber diferencias bastante grandes entre las trayectorias que se obtendrían analíticamente (si fuera posible obtenerlas) de las ecuaciones que dominan la dinámica del sistema y las trayectorias que se obtienen del simulador realizando la simulación numérica. A efectos de cuidar estos errores, se corroboró el simulador desarrollado contra algunos bloques que vienen en el *Aerospace Block Library* de las nuevas versiones de Matlab.

Es importante en este punto recalcar el hecho que el simulador entregado fue enteramente realizado por los integrantes de este equipo incluyendo todas las *scripts*, las funciones y los archivos *.mdl* (a los cuales nos referiremos de ahora en más simplemente como MDLs). A continuación, en las próximas secciones, se detallan los procesos de construcción del simulador del lazo abierto y se muestran algunas de las pruebas realizadas sobre el mismo.

<sup>a</sup>Recordamos que el vector  $u$  es el vector de entradas que podemos controlar desde la lógica de control y el vector  $h$  representa entradas al sistema de las cuales no tenemos control alguno, como por ejemplo, la velocidad del viento.

### 3.1. Desarrollo del simulador

El primer enfoque para el desarrollo del simulador fue claro, realizar un programa sobre la plataforma *Matlab* que resolviera  $\dot{x} = f(x, u)$ <sup>b</sup> utilizando un algoritmo estandar de resolución de Ecuaciones Diferenciales Ordinarias (EDO de ahora en adelante) como puede ser el algoritmo de *Runge - Kutta*. La principal diferencia entre la primer versión del simulador que se intentó desarrollar con las versiones posteriores es que en esa primer versión no se utilizó *simulink*.

#### 3.1.1. Primer versión del simulador

La primer versión del simulador prescindía por completo de la herramienta *simulink* que proporciona *Matlab*, y simplemente utilizaba el método *ode45.m* (que implementa el método de resolución *Runge-Kutta*) de *Matlab* a los efectos de la resolución numérica. Esta implementación fue descartada dado que adolecía del defecto que detallamos a continuación: por la manera en que se debe invocar el método de resolución numérica *ode45*, el código precisaba que la función  $f(x, u)$  fuera exclusivamente función de la variable  $x$ . Esta condición impone pues que la entrada  $u$ , estuviese definida como función de matlab, y dependiera sólomente de la variable tiempo  $y$ , en todo caso, del vector de estados  $x$ . Es decir, esto limita la dependencia de la entrada en un determinado instante solamente al valor del estado en ese mismo instante, limitando así la dinámica de futuros lazos de control a una dependencia únicamente algebraica sobre el estado. Es de observar que las simulaciones sobre esta primer versión (versión 1.0) del simulador eran parecidas a lo que se obtendría después con las versiones posteriores y, a pesar de que tenía un tiempo de espera apreciable, las simulaciones corrían razonablemente bien. Para subsanar esta dificultad se abandonó este enfoque por el abordaje de la creación del modelo en *simulink*, que provee un entorno de desarrollo mucho mas afín a la resolución de lazos de control más complejos.

#### 3.1.2. Versión del simulador en *simulink*

Finalmente se adoptó la estrategia de elaborar un modelo *simulink* basado en la ecuaciones del MVE. De esta manera se construyeron todos los bloques necesarios para su elaboración. En la figura 3.1 se puede apreciar dicho archivo *.mdl*.

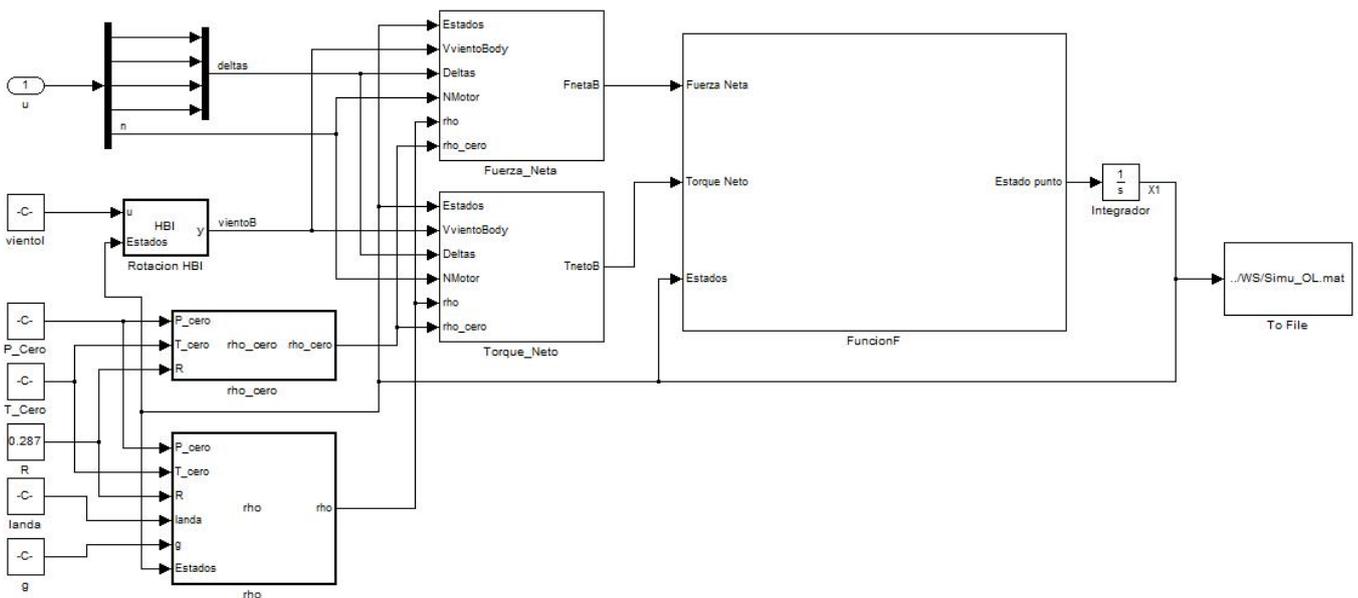


Figura 3.1: Modelo MDL elaborado en base a las ecuaciones del MVE.

<sup>b</sup>Omitimos el vector  $h$  por razones de compactidad y de delimitar las variables más relevantes de dicha ecuación, es claro que está ecuación también depende del vector  $p$  que se definió en la ecuación 2.23, el cual también omitimos.

En esta estructura se puede observar de forma clara que la dependencia de la función  $f(x, u, h, p)$  se puede escribir de la siguiente manera:

$$\dot{x} = f(x, u, h, p) = f^*(x, F, T) \quad \text{con} \quad \begin{cases} F = f_1(x, u, h, p) \\ T = f_2(x, u, h, p) \end{cases} \quad (3.1)$$

en términos de  $F$  y  $T$ , nuevas variables que representa la fuerza neta y el momento neto respectivamente. Es decir, la dependencia de  $f$  con las variables  $u$ ,  $h$  y  $p$  entra al sistema por la forma de las fuerzas y momentos. Aunque claro está, de ahora en más nos referiremos a este sistema de la forma  $\dot{x} = f(x, u)$  por razones de compactadas y relevancia.

Se puede observar en la figura 3.1 que se deben cargar al MDL los parámetros concernientes a  $h$  y  $p$ , a saber, los valores para la entrada  $\vec{v}_v$  de velocidad del viento (en este caso expresado en términos de la base del SI) y también para las variables  $P_0$ ,  $T_0$ ,  $g$ ,  $R$  y  $\lambda$ . También será importante setear los datos relativos al modelo del avión dentro de los distintos bloques, en particular esto se debe hacer dentro de los bloques dedicados al cálculo de la fuerza neta y el momento neto como se explicará más adelante. Inicialmente dichos bloques se encuentran cargados con los datos correspondientes al modelo del *BEAVER*. Como se verá más adelante, el *software* integrado que se entrega con este proyecto tiene la posibilidad de cambiar fácilmente de modelo de avión siempre y cuando se sigan las especificaciones relativas a la ubicación de los parámetros necesarios que se encuentran en el manual de usuario. Además se deben setear las condiciones iniciales para la simulación en el integrador, o sea, en el bloque *integrador* que se observa sobre la parte superior derecha de dicha figura.

El bloque *funcionF* es el encargado de representar las ecuaciones del MVE, y tiene la forma que tendría  $f^*$ . La figura 3.2 muestra la implementación de dicho bloque, donde se pueden apreciar los sub-bloques que la componen. El primer bloque *rotaciónHIB* representa la ecuación 2.4, éste es el cambio de base entre el vector  $\vec{v}_B$  y el vector  $(\dot{x}, \dot{y}, -\dot{z})$ . El segundo bloque es el encargado de realizar la rotación correspondiente a la ecuación 2.7 mediante la matriz  $M$ , este es el cambio de base entre el vector  $\vec{w}_B$  y el vector  $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$  correspondiente a las derivadas temporales de los ángulos de *Euler*. Finalmente, los dos últimos sub-bloques, los bloques *Primera Cardinal* y *Segunda Cardinal* son los encargados de representar las ecuaciones 2.10 y 2.12 del MVE respectivamente, cuyas variables de entrada son  $F$ ,  $T$  y  $x$ .

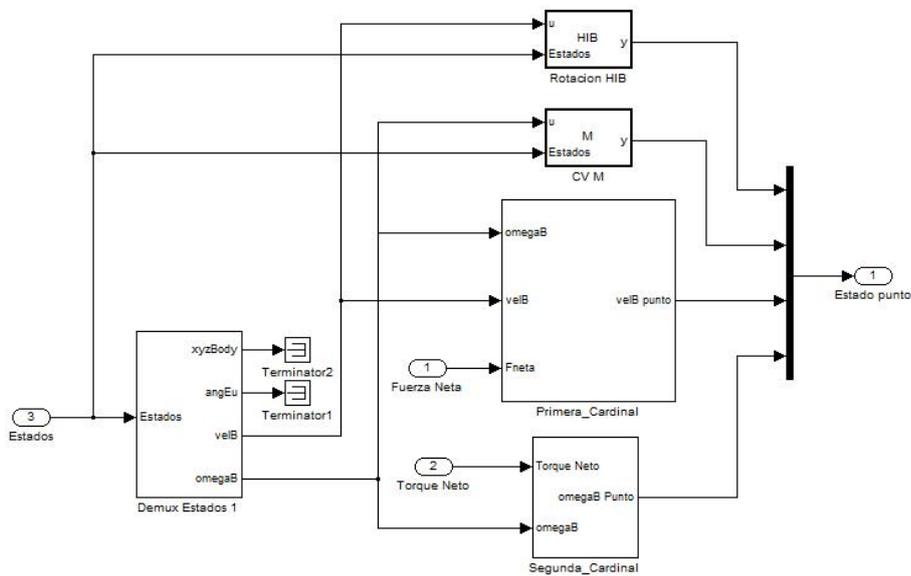


Figura 3.2: Modelo MDL elaborado para el bloque *funcionF*.

Continuando con los bloques de interés, se muestran a continuación en la figura 3.3 la forma de los bloques que implementan la fuerza neta y el momento neto. El bloque *Fuerza Neta* está compuesto por tres grandes sub-bloques, estos son: el bloque *Peso B* cuya misión es expresar el vector peso del avión en términos de la base del SB, el bloque *FAero B* representa las fuerzas aerodinámicas sobre las superficies de vuelo según se las describió en la subsección 2.4.2, y finalmente el bloque *FT Motor* que implementa la

fuerza y el momento ejercido por el motor según lo descrito en la subsección 2.4.3. Por otro lado, el bloque Torque Neto cuenta sólo de dos sub-bloques principales, dichos sub-bloques son: **TAero B** que representa a los momentos aerodinámicos y el sub-bloque **FT Motor** ya mencionado.

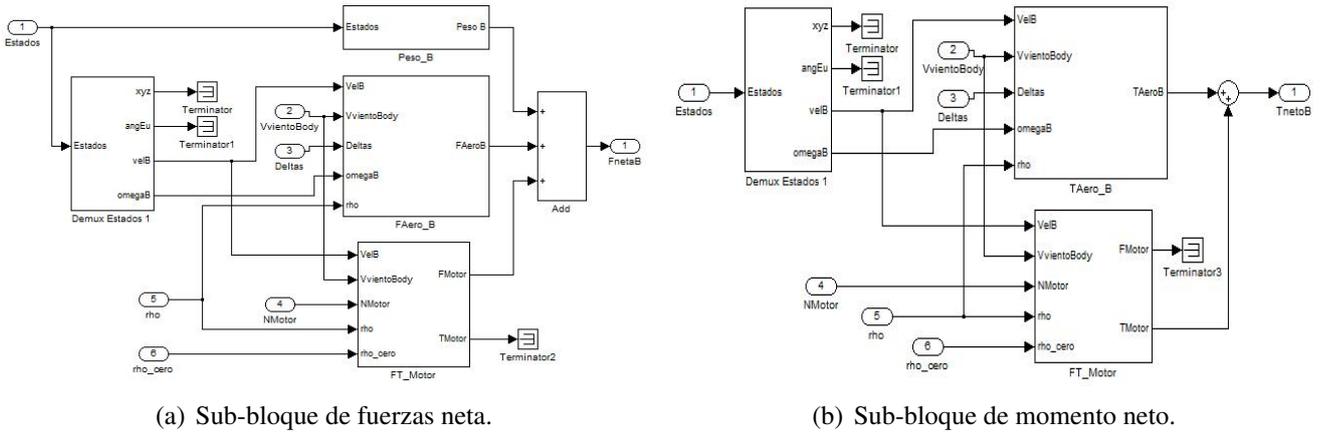


Figura 3.3: Modelos MDLs que implementan la fuerza neta y el momento neto sobre el sistema.

Estos dos bloques representan la suma vectorial de las fuerzas involucradas en el modelado según la sección 2.4. A continuación, en la siguientes figuras, se muestran los sub-bloques que se detallaron en el párrafo anterior.

- ▷ **Peso B:** Cambio de base del peso expresado en el SI como  $mg\hat{k}_I$  a la base del SB. Ver ecuación 2.28.

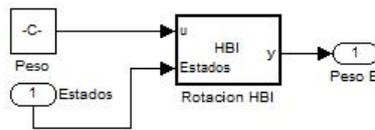


Figura 3.4: Modelo MDL elaborado para el bloque **Peso B**.

- ▷ **FAero B:** Genera el vector de fuerza aerodinámica. Ver subsección 2.4.2.

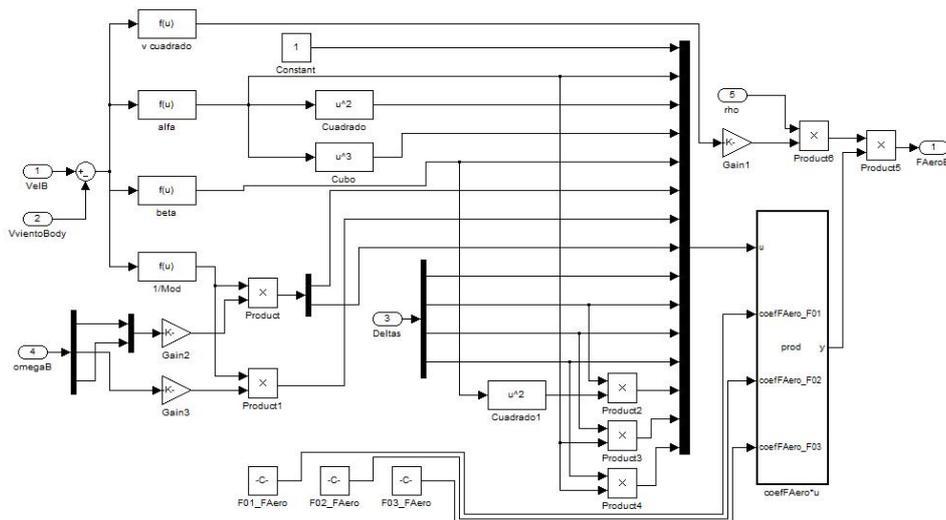


Figura 3.5: Modelo MDL elaborado para el bloque **FAero B**.

▷ **TAero B:** Genera el vector de momento aerodinámico. Ver subsección 2.4.2.

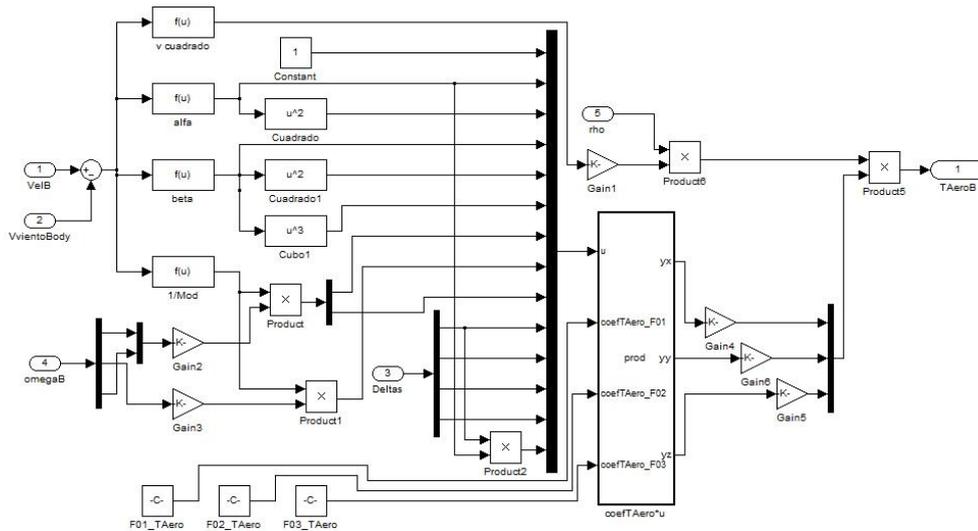


Figura 3.6: Modelo MDL elaborado para el bloque TAero B.

▷ **FT Motor:** Genera los vectores de fuerza y momento del motor. Ver subsección 2.4.3.

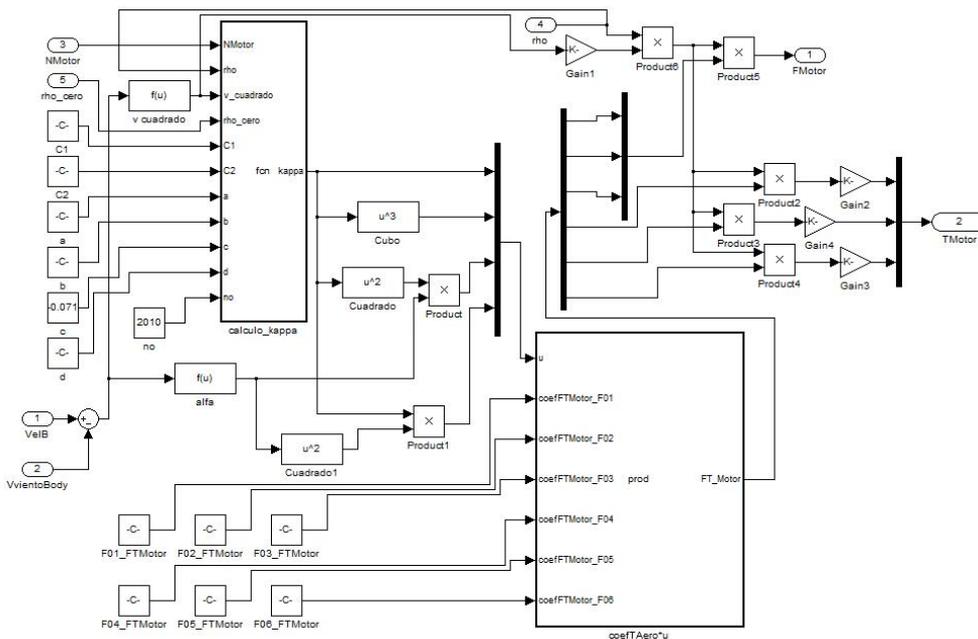


Figura 3.7: Modelo MDL elaborado para el bloque FT Motor.

Para correr los bloques que fueron mostrados en las figuras 3.4, 3.5, 3.6 y 3.7 se deben setear algunas constantes. Tanto las funciones *.m* como el *software* integrado están pensados para tomar dichos datos desde distintos archivos *.txt* que deberán estar ubicados acordemente en la carpeta correspondiente, esto se explicará más en detalle en el manual de usuario y en el capítulo 7 de esta documentación.

La constante a setear en el bloque **Peso B** es el vector  $(0, 0, mg)$  dentro de la caja **Peso**. En los bloques **FAero B**, **TAero B** y **FT Motor** se deben setear distintas ganancias (bloques de la forma **Gain i**) a los valores de  $b$ ,  $b/2$ ,  $\bar{c}$  y  $S/2$ , y también se deben setear las matrices de coeficientes para las fuerzas y momentos según sea el caso. Otras constantes que es necesitan cargar son los parámetros del motor, a saber, los parámetros  $\{C_1, C_2, a, b, c, d, n_0\}$  correspondientes al modelado del motor y que son utilizados para el

cálculo de  $\kappa$ , la cual responde a la forma de la ecuación 2.49. Todos estos parámetros se encuentran seteados en los MDLs en los valores correspondientes el *BEAVER*, podrían cambiarse utilizando otros archivos *.txt* (lugar donde se encuentran los valores utilizados, dentro de la carpeta 'Beaver') o bien utilizando variables cargadas en el *workspace* de *Matlab*. Se entregan los MDLs para ambas modalidades. En el manual de usuario de profundizará más sobre los distintos tipos de utilización y sobre los archivos entregados.

Las matrices de cambio de base  $H_I^B$  y  $H_B^I$ , y la matriz de cambio de coordenadas  $M$  fueron implementadas como funciones embebidas y son representadas por los bloques HIB, HBI y M que aparecen en las figuras mostradas durante la presente sección.

Finalmente, los últimos dos bloques que nos quedan por comentar son los bloques rho y rho cero. Estos bloques son simplemente funciones embebidas y su funcionalidad es clara; el bloque rho cero calcula el valor de  $\rho_0$  según la ecuación 2.19, mientras que el bloque rho calcula la variación de la densidad del aire con la altura  $\rho(z)$  según el modelado realizado para la atmósfera, a saber, según la ecuación 2.18.

## 3.2. Modelos implementados para el lazo abierto

El modelo MDL que se describió en la sección anterior es el modelo más completo del sistema en lazo abierto que implementamos. Éste es el MDL que utilizamos como simulación del sistema pues, entendimos que era el más exacto que podíamos construir dada las hipótesis de modelado. Pero también se construyeron otros MDLs para este sistema que se usaron para distintos objetivos.

Dichos MDLs se contruyeron básicamente a efectos del diseño y construcción del controlador lineal. Como se explicará en los capítulos 5 y 6, mediante un proceso de linealización de las ecuaciones del MVE se obtuvieron ecuaciones lineales en torno a dos trayectorias de interés; una trayectoria de vuelo rectilíneo a velocidad constante y una trayectoria de giro circular en el plano  $xy$  a velocidad angular constante, para luego proceder con el diseño del controlador lineal que se utilizará para seguir dichas trayectorias.

Para el caso del diseño del controlador para las trayectorias del tipo de la primera trayectoria mencionada en el párrafo anterior, se utilizó el MVE explicado en el capítulo 2 con la simplificación de que el valor de  $\rho(z)$  fuese constante con la altura e igual al valor que tendría  $\rho$  a  $1800m$ . De esta manera se usó para  $\rho$  el valor de  $\rho_{1800}$ . Esta hipótesis simplifica de gran manera la linealización del MVE para luego proceder al diseño del controlador. Como es natural, en la medida que no varíe mucho la altura de vuelo este modelo se parecerá más al modelo que se utilizó para el sistema real. En cuanto al diseño del controlador para lograr seguir trayectorias de giro circular en el plano a velocidad angular constante, se mantuvo la hipótesis sobre el  $\rho(z)$  que se explicó anteriormente pero se tuvo que modificar tres de las ecuaciones del MVE a fin de obtener un sistema SLIT luego del proceso de linealización. Con lo cual, como se modificaron las primeras tres ecuaciones del MVE, cambia la linealización del mismo.

En ambos casos se construyeron MDLs que reflejaran los cambios efectuados sobre el MVE. Estos MDLs se utilizaron para poder comprobar las linealizaciones analíticas de los dos sistemas frente a la linealización numérica de *Matlab*. Es decir, dado que las ecuaciones involucradas son grandes y complejas, de alguna manera debíamos asegurarnos que todas las derivadas que realizamos estuvieran correctamente realizadas e implementadas. De esta manera, los MDLs construidos en esta instancia fueron a fin de poder cotejar el correcto cálculo de las linealizaciones para ambos casos. Como se explicará en el capítulo ?? los resultados obtenidos fueron ampliamente satisfactorios.

## 3.3. Pruebas del simulador

Las pruebas del simulador del lazo abierto se fueron efectuando durante el proceso de construcción del mismo. Es por esto que las primeras pruebas que mostraremos en esta sección son bastante básicas, incluso el proceso de testeo fue un aprendizaje en sí mismo, dado que a medida que se iban realizando las pruebas

obteníamos una mayor comprensión e intuición sobre el sistema y el MVE. A continuación, en las siguientes subsecciones enumeramos algunas de las pruebas hechas y las conclusiones obtenidas de las mismas.

### 3.3.1. Prueba 1: Caída libre

Esta fue la primer prueba realizada sobre la primer versión del simulador del lazo abierto que se construyó. Creemos que la misma representa una prueba bastante natural, razón por la cual la incluimos en esta documentación. De todas formas, esta prueba no será útil para ilustrar algunas conclusiones que hemos obtenido sobre la performance de nuestro simulador para el lazo abierto.

La prueba que se realizó fue la siguiente: se consideró una esfera (para el tensor de inercia  $I_{OB}$  se utilizó el de una esfera), y se la dejó bajo la acción de su propio peso y de un momento constante según la dirección  $\hat{i}_B$  (para esta prueba  $\hat{i}_B = \hat{i}_I$ ). Las condiciones iniciales fueron todas nulas, salvo la velocidad lineal según  $\hat{i}_B$  y la altura inicial. Las especificaciones para esta prueba serían entonces:

Condiciones iniciales		Otras especificaciones
$x_0 = 0$	$v_{Bx0} = 20m/s$	$F_{Neta} = cte = mg\hat{k}_I$
$y_0 = 0$	$v_{By0} = 0$	$T_{Neto} = cte = T_{Nx}\hat{i}_B = 5Nm\hat{i}_B$
$z_0 = 1000m$	$v_{Bz0} = 0$	Tiempo de simulación: $t = 10s$
$\varphi_0 = 0$	$w_{Bx0} = 0$	Tensor de inercia: $I_{OB} = 10Idkgm^2$
$\theta_0 = 0$	$w_{By0} = 0$	
$\psi_0 = 0$	$w_{Bz0} = 0$	

Cuadro 3.1: Tabla de especificaciones de la prueba 1.

Como se trata de un caso sencillo es fácil saber como debería reaccionar este sistema. Tomando en cuenta los primeros 6 elementos del vector de estados, la evolución de los estados  $x$ ,  $z$  y  $\varphi$  debería ser según las ecuaciones 3.2, 3.3 y 3.4 respectivamente, mientras que el resto de las variables deberían permanecer nulas.

$$x(t) = v_{Bx0}t \quad \Rightarrow \quad x(t)|_{t=10s} = 200m \quad (3.2)$$

$$z(t) = z_0 - \frac{gt^2}{2} \quad \Rightarrow \quad z(t)|_{t=10s} \approx 509,675m \quad (3.3)$$

$$\varphi(t) = \frac{T_{Nx}t^2}{2I_{xx}} \quad \Rightarrow \quad \varphi(t)|_{t=10s} = 25rad \quad (3.4)$$

Básicamente queremos observar que el modelo realizado sea consistente para casos sencillos, además de evaluar algún tipo de *performance* frente a los errores numéricos. Los resultados obtenidos para la simulación del lazo abierto en las condiciones de la tabla 3.2 son los que se exponen a continuación en la figura 3.8. Como se puede apreciar en dicha figura se obtuvieron las parábolas buscadas, tanto en el plano  $xz$  como para  $\varphi(t)$ . Además, se observan las rectas correspondientes en el plano  $xy$  e  $yz$ , y también para los demás ángulos de Euler  $\theta$  y  $\psi$  donde se debería cumplir  $\theta(t) = cte = 0$  y  $\psi(t) = cte = 0$ . Extrayendo datos de dicha figura tenemos que:

$$\begin{aligned} x(t = 10s)_{sim} &= 200m \\ z(t = 10s)_{sim} &= 509,5m \\ \varphi(t = 10s)_{sim} &= 25m \end{aligned} \quad (3.5)$$

Lo cual demuestra que las trayectorias son las adecuadas para esta prueba. También se puede notar amplificando las gráficas de la figura 3.8 una pequeña desviación sobre el final de las trayectorias de las trayectorias sobre  $y = 0$ ,  $\varphi = 0$  y  $\psi = 0$ . El orden de esta desviación es de  $2 \times 10^{-5}$  como se puede observar

mejor en la figura 3.9, dicha desviación la consideramos despreciable frente a los 200m de avance del avión según el eje  $x$ . Este error se lo adjudicamos a los errores numéricos.

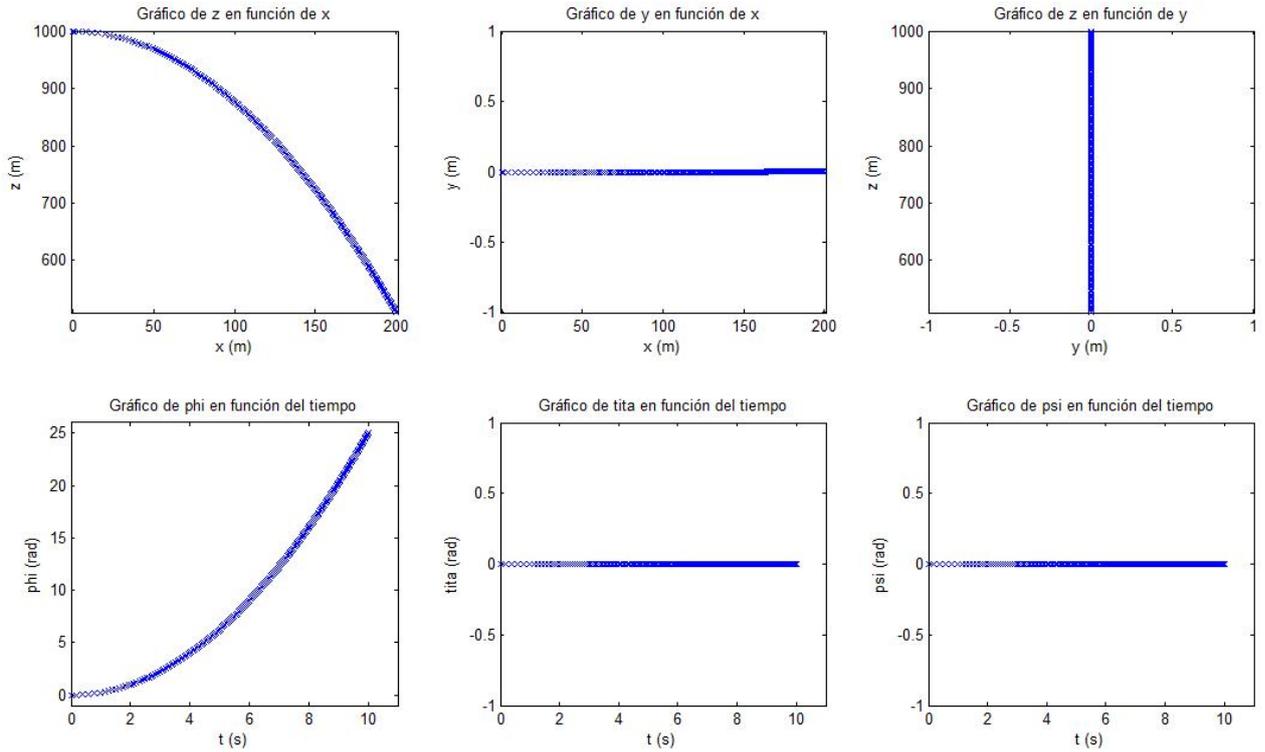


Figura 3.8: Trayectorias predichas por el simulador para la prueba 1.

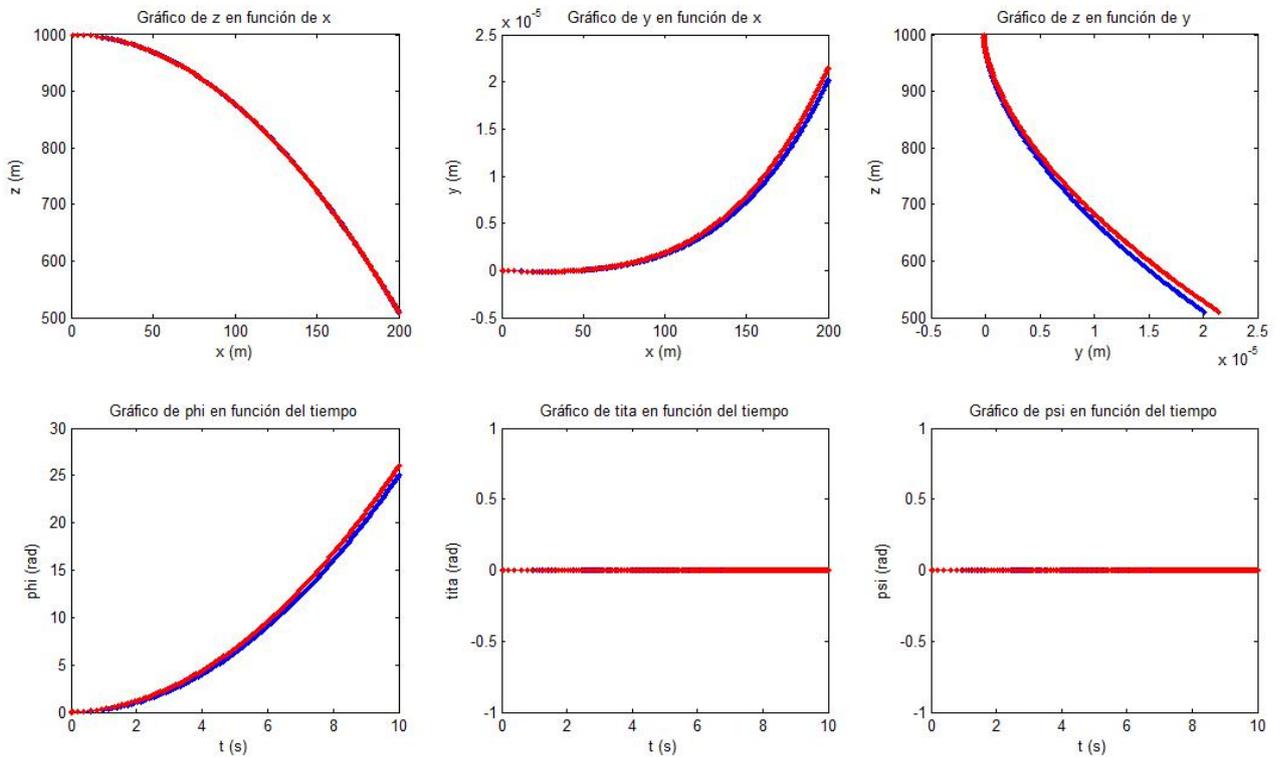


Figura 3.9: Trayectorias para la prueba 1. **Azul:** simulador implementado. **Rojo:** utilizando el sistema 6DoF de *Matlab*.

En esta última figura, la figura 3.9, se muestra una comparación de la *performance* de nuestro simulador y un MDL construido a partir del bloque 6DoF del *Aerospace Block Library* de *simulink* para



# Generador de trayectorias

---

La generación de trayectorias es una parte clave de cualquier sistema de navegación autónomo. El tipo de trayectorias generadas determinará en última instancia que estrategias de control podrán utilizarse y la complejidad de las mismas. Por lo tanto, se requiere a la hora de construir el sistema generador de trayectorias, tomar en cuenta las consecuencias que este tendrá sobre el sistema de control y evaluar todo el sistema como uno, en otras palabras, no se puede pensar independientemente el controlador y el generador de trayectorias pues están íntimamente ligados.

## 4.1. ¿Qué entendemos por generador de trayectorias?

En una primera instancia, problema que debe resolver el generador de trayectorias es: dados  $N$  puntos en el espacio (waypoints de ahora en más), generar una curva a seguir en el espacio de posiciones, pasando por los waypoints especificados y en el orden especificado. Los puntos especificados a priori podrían, además de tener un orden predeterminado para seguirlos, tener un requerimiento en cuanto al tiempo en que se debe llegar a ese punto (parametrización explícita en el tiempo). Extendiendo el concepto de trayectoria, se podría incluso especificar una curva a seguir en el espacio de velocidades.

De manera aún más general, podemos entender por trayectoria a un conjunto de valores (curva) en el vector de estados (o algún sub-vector del mismo) que se desea que el sistema sea capaz de seguir. Otra vez, esta evolución puede estar explícitamente parametrizada en el tiempo o no. De esta definición surge una primera decisión clave para un sistema de generación de trayectorias: definir cual es el sub-conjunto de variables del espacio de estados que se desea controlar y cuales quedan libres para ser modificadas por el sistema de control.

## 4.2. Elementos a considerar para la construcción del generador de trayectorias

A efectos de diseñar un generador de trayectorias debemos poner previamente especial atención en los requerimientos que debe cumplir y en algunas características deseables para el mismo. Dentro de estos aspectos, observamos que hay algunos que son intrínsecos al generador de trayectorias y otros que tiene más que ver con la relación que éste tendrá con el controlador a diseñar y la capacidad computacional disponible.

Dentro de los aspectos relativos a generador de trayectorias propiamente dicho podemos mencionar que las especificaciones a tener en cuenta son las siguientes:

- ▷ **Tipo de recorrido que se busca del avión.** Por ejemplo, debe definirse si la navegación será bi o tri-dimensional. También debe decidirse cuales serán los inputs del usuario, si esto consistirá en un conjunto de waypoints ordenados o no, con un tiempo máximo de pasada o no, entre otros aspectos a definir.

- ▷ **Restricciones del sistema físico que impiden recorrer una trayectoria arbitraria.** Entre otros, hay que considerar un máximo radio de curvatura que puede realizar el vehículo con acciones de control aceptables, máxima velocidad de vuelo, máxima velocidad de ascenso, y otras más básicas como que exista continuidad de la posición y velocidad del vehículo (3ra Ley de Newton).
- ▷ **Criterio de trayectoria óptima.** A efectos de elegir una buena trayectoria entre varias posibles que cumplan con los objetivos trazados en los puntos anteriores, se puede definir o no qué variables del sistema se desean optimizar según algún criterio. Casos típicos serían los de elegir las trayectorias más cortas, las que consuman menos combustible, las que tengan menores acciones de control, o las que tengan los mayores radios de giro, por mencionar algunas de las posibles elecciones.

Como decíamos párrafos atrás, no sólo los items mencionados en el listado anterior son los que hay que tener en cuenta, sino que nace necesariamente una necesidad de estimar el costo computacional que tiene la generación de trayectorias (en especial si se desea hacer en la computadora de abordo, con prestaciones bastante más limitadas que la de un PC de escritorio) y también se debe considerar el costo que tendrá el sistema de control necesario para lograr seguir las trayectorias implementadas.

Entre los elementos relacionados con estos costos se encuentran:

- ▷ **La naturaleza del sistema a controlar.** El sistema a controlar es inherentemente no lineal, y por lo tanto las estrategias de control sobre el sistema tienen un costo computacional mayor que las estrategias de control lineal. Sin embargo, existen trayectorias sobre las que se puede linealizar nuestro sistema; a saber, los vuelos en línea recta sobre el plano horizontal y vertical, y las trayectorias circulares a velocidad constante en el plano horizontal. Esto además de reducir el costo computacional permite el uso de un gran conjunto de herramientas de control lineal, que son de simple aplicación y con las que estamos más familiarizados.
- ▷ **Complejidad computacional.** Dado que los problemas de optimización con restricciones son problemas computacionalmente complejos, se pueden buscar soluciones con una menor cantidad de restricciones de las que tiene el problema real, dejando que el mismo sistema físico y el control sigan la trayectoria con las restricciones naturales que aparecen. Un ejemplo de esto podría ser no preocuparse por que exista continuidad en la velocidad de la trayectoria, y que cualquier discontinuidad (imposible en el sistema real debido a las leyes físicas) se resuelva con un control adecuado que adapte la velocidad suave pero rápidamente.
- ▷ **Compromiso optimalidad - complejidad computacional.** En la misma línea que el punto anterior, se pueden buscar optimizaciones casi sin restricciones, para luego imponer las restricciones en el mismo programa generador de trayectorias, sobre la trayectoria óptima previamente encontrada. Esto necesariamente conduce a soluciones que respetan todas las restricciones que se desee, pero que son subóptimas. En resumen, es posible hacer un trade-off entre optimalidad y complejidad computacional de la solución.

Otra observación que es pertinente mencionar es que es casi imprescindible contar con un sistema de re-generación de trayectorias, a utilizar para calcular nuevas trayectorias en los casos que el vehículo se aleja demasiado de una trayectoria a seguir. Esto último hace que el costo computacional de generar una trayectoria pase a ser un elemento crítico.

### 4.3. El generador de trayectorias implementado

Tomando en cuenta todas las consideraciones mencionadas en el punto anterior, procedimos a la construcción de un generador de trayectorias 3D que se ajuste a las especificaciones principales sobre la

función del generador de trayectorias. Además, tuvimos que definir y resolver muchas de las interrogantes que se plantearon también en la sección anterior.

En particular, el punto desarrollado sobre las trayectorias sobre las cuales se puede linealizar el sistema puede parecer en primera instancia algo limitador en cuanto a las posibles trayectorias a generar. Esto en realidad no es del todo cierto, ya que cualquier trayectoria podría aproximarse infinitesimalmente por una concatenación de las trayectorias mencionadas. Sin embargo, esta estrategia implicaría tener un punto de operación que varía permanentemente, con la consiguiente necesidad de recalcular las posiciones de las superficies de vuelo y otras variables de control. Como se dice a continuación, finalmente optamos por hacer una concatenación finita de trayectorias linealizables, ya que representaban un bajo costo computacional y una manera más sencilla de realizar el control.

### 4.3.1. Especificación funcional del generador de trayectorias.

Como resultado de todas estas consideraciones, en primera instancia decidimos proceder con un generador de trayectorias que concatenara rectas y círculos, y que operara según los puntos que se exponen a continuación.

1. Las trayectorias serán una curva en el sub-espacio de estado de 6 variables: posición y velocidad. Esto implica una dependencia implícita con el tiempo, pues se indica la velocidad del avión, pero no es absoluta en términos de que no existe una posición correcta para cada instante de tiempo. De este modo se toman únicamente 6 de las 12 variables que pertenecen al vector de estados, permitiéndose la manipulación de las otras 6 por parte del sistema de control, y se elimina la necesidad de hacer un control explícito en la variable tiempo.
2. La navegación será tridimensional, y los inputs del usuario serán un conjunto ordenado de waypoints que deben respetar ciertas reglas. Entre otros, debe haber una distancia mínima -a definir- entre waypoints (de manera que no se exijan radios de curvatura excesivos) y la diferencia de altura entre waypoints sucesivos debe estar acotada. En una segunda instancia podría pensarse en la generación de trayectorias en los casos que los waypoints no cumplen con las restricciones necesarias, por ejemplo no pasando por algún o algunos waypoints, y obtener la trayectoria a utilizar mediante alguna minimización que permita encontrar una trayectoria aceptable según algún otro criterio.
3. Se exigirá que la trayectoria tenga continuidad en posiciones pero -en principio- no se considerarán restricciones de continuidad de la velocidad. Sí se impondrá una máxima velocidad de vuelo.
4. El conjunto de waypoints se proyectará sobre un plano horizontal para resolver en primera instancia un problema en 2D. Esto presenta muchas facilidades adicionales y resta parámetros libres, lo que permite minimizar la complejidad computacional.
5. Para la proyección 2D de  $N$  waypoints, se creará una trayectoria de  $N - 1$  rectas y  $N - 2$  círculos, de manera de tener una sucesión recta-círculo-recta-círculo-...-recta.
6. El tratamiento de puntos a distinta altura se resolverá de la siguiente manera: se clasificarán los waypoints según conjuntos de waypoints con alturas similares, y se los tratará en esos subconjuntos como pertenecientes a un mismo plano horizontal. Los cambios de altura se agregarán sobre la solución 2D agregando tasas de trepado a los segmentos rectos. (ILUSTRAR ESTE PUNTO?)
7. En una primera implementación, se impondrá también que las trayectorias concatenadas sean tangentes en el sub-espacio de posiciones, o en el caso de la concatenación entre una trayectoria en el plano horizontal y otra en el plano vertical, se buscará que la concatenación respete la tangencia en la proyección de las trayectorias sobre el plano horizontal. De esta manera se evita la necesidad de minimizar en el conjunto de todas las trayectorias posibles, pero necesariamente se pierde optimalidad.

Partiendo de la base de estas especificaciones, pasamos a la etapa de la construcción propiamente dicha del generador de trayectorias 3D.

### 4.3.2. Análisis del problema geométrico en 2D.

Para empezar a atacar el problema, nos concentramos en la generación de la trayectoria geométrica (es decir, la trayectoria solamente en el subespacio de posiciones), sin considerar de momento al tiempo como variable (implícita o explícita). Estrictamente hablando no nos deshacemos de la cinemática de la situación ya que existen al menos dos elementos dinámicos que deben ser considerados: El sentido del recorrido (la imposición de una "flecha" de tiempo) ya que, por ejemplo, para una recta y un círculo tangentes existen dos posibles segmentos de recta a recorrer, por lo que es un requerimiento conocer en que sentido se están recorriendo dichas curvas. El otro elemento cinemático.<sup>a</sup> tener en cuenta es la restricción del radio mínimo de giro que puede describir el vehículo en cuestión (que surge de un análisis dinámico del vuelo, como una restricción sobre la mayor velocidad angular que puede existir en vuelo de crucero), ya que poco sentido tendría tratar de generar trayectorias que ignoraran por completo estas restricciones reales <sup>a</sup>

Con los elementos elegidos según la descripción funcional, la familia de trayectorias cuenta en principio con 2 grados de libertad por recta y 3 grados de libertad por círculo, siendo  $N - 1$  rectas y  $N - 2$  círculos, dando un total de  $5N - 8$  grados de libertad. Al imponer las restricciones de tangencia entre un círculo y la recta que le antecede y la que le precede, se pierden  $2(N - 2)$  grados de libertad (dos ecuaciones por cada círculo que haya en la trayectoria), quedando un total de  $3N - 4$  grados de libertad efectivos para la familia de trayectorias considerada. En un segundo análisis podemos ver que existen dos parámetros libres adicionales, que determinan donde comienza la primera recta y donde termina la última, dando un total de  $3N - 2$  grados de libertad efectivos. A grandes rasgos, esto significa que se tiene la libertad de fijar 3 parámetros por waypoint existente y encontrar una solución al problema. Cabe notar que estrictamente esto no es cierto, pues dependiendo de los parámetros que se fijen entre los  $9N - 14$  posibles, no siempre existirá una solución para el sistema de ecuaciones que queda planteado, pues el mismo no es lineal en todas las variables).

Evidentemente si se fijan arbitrariamente los  $3N - 2$  parámetros, no existe ninguna garantía de que la solución encontrada -de existir- sea una 'buena' solución. Es por esto que entra en juego la idea de hacer una optimización en el espacio de las soluciones posibles, por lo que es importante conocer cuantos son los parámetros independientes del sistema. Nótese que hasta ahora para el análisis de la trayectoria no se ha usado nunca la restricción de que la misma pase por los waypoints. Esto se debe a que no se hará esta exigencia sino que se contemplará la distancia a los waypoints entre los costos a minimizar en la optimización. Los motivos para esto son dos: el principal es que el despeje analítico en el caso de exigir el paso por los waypoints es engorroso y no facilita la búsqueda del resultado óptimo en el espacio de soluciones. El otro motivo es que en ciertas circunstancias exigir pasar por un conjunto de puntos estrictamente puede resultar en una trayectoria 'mala' desde otros puntos de vista (como por ejemplo el largo total de la trayectoria) pero sin embargo no representar ninguna ganancia real, pues la precisión del sistema de seguimiento es mucho menor que la precisión que se exige de la trayectoria.

A partir del análisis desarrollado en esta sección pasaremos al planteo del problema, comenzando por la elección de la representación para hacer dicho planteo y continuando por los procesos de optimización, generalización a 3D y post-procesamiento de la solución.

### 4.3.3. Representación elegida.

La elección de la representación para caracterizar los elementos de nuestra trayectoria fue una tarea que subestimamos inicialmente. Al igual que nosotros en un comienzo, es posible que el lector se pregunte ¿qué complicación puede surgir a la hora de elegir como representar un conjunto de rectas y otro de círculos?

---

<sup>a</sup>Como veremos más adelante, este dato se usa como insumo para esta etapa pero sin imponer la restricción estrictamente. Por más información referirse al manual de usuario.

En primera instancia parecería que la opción natural sería caracterizar a las rectas según  $y = a*x + b$ , y a los círculos según  $(x - xc)^2 + (y - yc)^2 = R^2$ . Esta representación sin embargo contiene algunos problemas: los puntos de intersección (o de comienzo y fin de la sección "útil" de cada recta y círculo), que son muy útiles y necesarios para evaluar la trayectoria dada y controlar al vehículo alrededor de la misma, deben ser despejados mediante la resolución de un sistema no lineal. Adicionalmente, no contienen explícitamente el sentido en que se deben recorrer los tramos y quizás la mayor de sus faltas es la no continuidad entre las rectas representadas y los parámetros que las describen (por ejemplo la representación de rectas en el entorno de ser verticales, con  $a = Inf$ ). Esto es un gran problema si se pretende hacer una búsqueda en el espacio de estos parámetros, pues no se puede buscar partiendo de la premisa de que rectas parecidas tienen representaciones parecidas (condición necesaria para cualquier algoritmo de optimización que se quiera usar).

Aspirando a elegir una representación que no tuviera ninguno de los problemas anteriores, finalmente elegimos la siguiente:

Las rectas fueron parametrizadas según  $v$ , estando  $v$  contenido en el intervalo  $[0, v_{max_i}]$   
 $(x, y) = (x_{r_i}, y_{r_i}) + v \cdot (\cos(phi_i), \sin(phi_i))$

Y los círculos según un parámetro  $w$ , estando  $abs(w)$  contenido en el intervalo  $[0, w_{max_j}]$   
 $(x, y) = (x_{c_j}, y_{c_j}) + R_j \cdot (\cos(w + theta_j), \sin(w + theta_j))$

De esta manera queda planteado un sistema con 4 variables por recta y 5 por círculo, dando  $9N - 14$  grados de libertad. De las restricciones que se imponen de tangencia y continuidad, surgen las siguientes restricciones:

Continuidad  $4(N - 2)$  ecuaciones:

$$(x_{r_i} + 1, y_{r_i} + 1) = (x_{c_i}, y_{c_i}) + R_i \cdot (\cos(w_{max_i} + theta_i), \sin(w_{max_i} + theta_i))$$

$$(x_{r_i}, y_{r_i}) + v_{max_i} \cdot (\cos(phi_i), \sin(phi_i)) = (x_{c_i}, y_{c_i}) + R_i \cdot (\cos(theta_i), \sin(theta_i))$$

Tangencia  $2(N - 2)$  ecuaciones:

$$sign(w_{max_i}) \cdot (-\sin(theta_i), \cos(theta_i)) = (\cos(phi_i), \sin(phi_i))$$

$$sign(w_{max_i}) \cdot (-\sin(w_{max_i} + theta_i), \cos(w_{max_i} + theta_i)) = (\cos(phi_i + 1), \sin(phi_i + 1))$$

Un rápido conteo permite ver que el número de grados de libertad efectivos se mantiene en  $3N - 2$  como se estimó en la sección anterior.

#### 4.3.4. Optimización de la solución.

Como se dijo anteriormente, la etapa de optimización es clave en el proceso de hallar una trayectoria adecuada y en nuestro caso también es la etapa en la que se 'hace' que la misma cumpla una de sus especificaciones funcionales más importantes: la de pasar por los waypoints.<sup>b</sup> Antes de profundizar en como implementamos esta etapa, es necesario comprender como funciona el algoritmo de optimización con ecuaciones no lineales.

En particular el algoritmo con el que trabajamos hace una búsqueda lineal para minimizar una cantidad dada, pero los conceptos son extrapolables a otros algoritmos similares. En primera instancia, es necesario construir una función de costo -que depende de todas las variables del espacio de búsqueda en principio- y que representa el costo.<sup>o</sup> "cuán mala" de una solución dada. Luego, a partir de una 'semilla' (valor inicial para las variables o simplemente una trayectoria inicial en el espacio de trayectorias posibles), se busca minimizar dicho costo, iterando de la siguiente forma: en cada paso se calcula numéricamente el gradiente de

<sup>b</sup>Notar que esto no tiene que ser necesariamente así. Es posible, pero se descartó por motivos que ya se explicaron, hacer un despeje analítico de los parámetros de la trayectoria para forzar que la misma pase por los waypoints y recién luego buscar una optimización de acuerdo a algún conjunto de criterios.

la función en ese punto y calcula un nuevo punto hacia el cual moverse en función del resultado. Idealmente el proceso termina cuando el gradiente es tan pequeño (y por lo tanto lo son los movimientos en la búsqueda) que no tiene sentido seguir buscando <sup>c</sup>. De manera más general, distintos algoritmos de optimización utilizan distintos criterios para elegir la dirección de movimiento en la búsqueda y no necesariamente la dirección del gradiente. En cualquier caso, este proceso es tanto más complejo computacionalmente cuanto más grande sea la dimensión del espacio de búsqueda.

Procedimos entonces a implementar directamente una minimización en el espacio de  $9N - 14$  parámetros dado por la representación que se describió en la sección anterior, incluyendo las ecuaciones o vínculos entre parámetros como parte de lo que debe minimizarse (idealmente esas ecuaciones o vínculos deben respetarse estrictamente). Este acercamiento a la minimización tiene la ventaja de no requerir más programación que la del algoritmo de minimización y la función a minimizar, donde se pueden considerar directamente las ecuaciones vínculo sin necesidad de despejarlas analíticamente. Para ejemplificar esta situación, supongamos que queremos resolver el siguiente sistema:

$$x + y - 6 = 0 \text{ (Ec1A)}$$

$$x - y = 0 \text{ (Ec2A)}$$

La solución de este sistema es fácilmente despejable y coincide con  $x=y=3$ . Este resultado se puede obtener como se dijo, despejando analíticamente el sistema, o mediante resolución de un sistema lineal de ecuaciones, lo cual es fácilmente programable y tiene un costo cuadrático en el número de variables (se requiere solamente invertir una matriz). Sin embargo, otra estrategia podría ser ingresar en un sistema iterativo, como el que usa para minimizar o hallar soluciones de sistemas que no se pueden despejar analíticamente. Para esto, podríamos considerar una función de costo

$$J = (x + y - 6)^2 + (x - y)^2$$

(que no es otra cosa que la suma de las ecuaciones vínculo al cuadrado), y una semilla para iterar, por ejemplo,  $x = 2, y = 1$ . De acuerdo al proceso descrito anteriormente, se iteraría hasta llegar a una solución tan cercana a  $x=y=3$  como se quiera, pero no exactamente ésta. La mayor ventaja de este mecanismo de resolución es la posibilidad de resolver el sistema sin despejar analíticamente las ecuaciones, o sin siquiera contar con ecuaciones que sean despejables analíticamente. Sin embargo, como aprendimos durante el desarrollo de dicho método, este camino también tiene la desventaja de ser computacionalmente complejo, y de acuerdo a la bibliografía, lo es exponencialmente en el número de variables.

En el problema que describíamos de minimizar en  $9N - 14$  variables, esto se reflejaba en no poder manejar más de 5 waypoints (31 parámetros) en una computadora de escritorio con prestaciones bastante más generosas que las que pueden estar disponibles a bordo del avión. Es necesario resaltar especialmente este punto, pues seguramente mediante una programación más eficiente sea posible lograr trabajar con un número mayor de waypoints. Sin embargo, consideramos que el hecho mismo de que con 5 waypoints el sistema fallara para una computadora de escritorio implicaba de alguna manera una "intratabilidad" del problema en un procesador como el que actualmente se posee en el avión de la UdelaR.

Intentando superar este problema, la solución final consistió en tomar la parametrización que se presentó, que tiene  $9N - 14$  grados de libertad, pero **optimizando en un espacio de  $3N - 2$  parámetros variables, que fueron los que determinamos como efectivamente libres en nuestro estudio anterior y despejando los otros  $5N - 12$  de manera analítica**, lo que minimiza significativamente el costo computacional del algoritmo optimizador.

En ambos casos, la estrategia para lograr una buena y pronta (y en este contexto pronta es sinónimo de computacionalmente barata) convergencia radicaba en la elección de la semilla para el algoritmo. Como criterio para esta semilla, construimos de manera 'ad hoc' una solución que si bien no pretendía ser óptima en ningún sentido, cumplía con todas las restricciones del problema y era, en principio, similar al tipo de solución

---

<sup>c</sup>El criterio de parada para el algoritmo puede variar pero esencialmente siempre el concepto es este: las 'mejoras' en el costo son tan pocas que se asume que se ha llegado a un mínimo local. En términos numéricos dicho criterio se puede expresar en un tamaño mínimo para el módulo del gradiente, o en variaciones relativas de la función de costo o del vector de variables.

que esperabamos fuera la devuelta por el generador. Es clave aquí hacer énfasis en la necesidad de que la semilla cumpliera con las restricciones básicas del problema (como que los radios de giro no fueran menores que el radio mínimo que es posible seguir por parte del avión), ya que de otra manera la convergencia quedaba seriamente comprometida. Con este objetivo en mente, construimos la siguiente semilla para el problema: primero tomamos el conjunto de las rectas que unían a 2 waypoints consecutivos (una trayectoria de círculos de radio nulo), y luego incluimos los círculos que eran tangentes a dos rectas consecutivas y que el punto de tangencia estuviera a una distancia dada desde los waypoints. Finalmente de "agrandan" los radios de la trayectoria resultante hasta que todos alcancen al menos el valor del radio mínimo al que puede girar el avión.

### **Función de costos a minimizar.**

La elección de la función de costos es en sí mismo un elemento crítico para que las soluciones halladas sean adecuadas. Deben elegirse en principio dos grupos de elementos: las 'cantidades' a minimizar y la función que las cuantifica, y los coeficientes de pesos relativos entre las distintas cantidades. Un simple ejemplo de esto sería: si deseamos maximizar los radios de giro (la 'cantidad'), podríamos minimizar la norma de  $1/R$ , siendo  $R$  el vector de todos los radios (una posible función), o bien podríamos minimizar  $1/\min(R)$  y luego ponderarla por un coeficiente 'alfa' para darle un peso relativo adecuado a las otras cantidades que se desean minimizar. A continuación se enlistan las cantidades elegidas para minimizar, junto con la función que las cuantifica y finalmente se hace un análisis de la función conjunta y de los coeficientes adecuados para lograr los resultados buscados.

- ▷ **El costo de distancia a los waypoints.** En primera instancia lo razonable sería minimizar la distancia entre la trayectoria planeada y los waypoints dados, de manera de asegurarnos que la trayectoria efectivamente pase por los mismos. Una manera fácil de implementar esto es sumar a la función de costos la siguiente ecuación, para cada círculo posible:

$$((x - xc)^2 + (y - yc)^2 - R^2)^2(1)$$

Esta ecuación mide la distancia al cuadrado desde un círculo a un punto, pero no toma en cuenta que no todo el círculo puede estar comprendido en la trayectoria. En particular, tuvimos que modificar esta ecuación de manera de agregar un costo adicional a los puntos que cayeran fuera del arco que nos interesa:

$$((x - xc)^2 + (y - yc)^2 - R^2)^2 + k.(\theta - \text{angulo}([x - xc, y - yc]))^2$$

Donde  $\theta$  es el ángulo que marca el centro del arco de interés y  $k$  una constante apropiada que vincula el peso entre la diferencia en el ángulo y la diferencia en el radio.

Dado que el primer y último punto no 'tienen' un círculo que pase por ellos, se debe sumar la contribución de la distancia de la primera y última recta al primer y último waypoint respectivamente:

$$(x_{r0} - x_1)^2 + (y_{r0} - y_1)^2 + (x_{rN} - 1 + v_{N-1} \cdot \cos(\text{phi}(N - 1)) - x_N)^2 + (y_{rN} - 1 + v_{N-1} \cdot \sin(\text{phi}(N - 1)) - y_N)^2$$

- ▷ **El costo de largo de trayectoria.** En nuestro sistema, dado los grados de libertad que tenemos, sería posible exigir que se pase por todos los waypoints y aún así tener  $2N$  grados de libertad sobrantes. Es por esto que es necesario pensar en que la trayectoria sea la más corta posible de toda la familia de soluciones existentes. En realidad, dado que no exigimos pasar exactamente por todos los waypoints, el conjunto de trayectorias posibles es aún más grande. Para minimizar el largo de la trayectoria se incluye un costo asociado al mismo que, de acuerdo con nuestra parametrización, se puede calcular como:

$$\text{sum}(v) + \text{sum}(\text{abs}(w.R))$$

- ▷ **El costo de los radios de giro.** En nuestro sistema nunca se incluyó la restricción física de que los radios de giro fueran mayores que ciertas cantidades, mayoritariamente debido al costo computacional de resolver problemas de minimización con restricciones. Para evitar que el generador devolviera radios

de giro por debajo de lo aceptable, se incluyó un costo asociado a los mismos que en realidad es inversamente proporcional a los radios que son menores a  $R_{min}$ . La cuantificación de este costo se hizo según:

$$R_{min}^2 - \min(R, R_{min})^2$$

Esta función si bien no es infinitamente derivable, sí es continua y también lo es su derivada, lo que hace que sea suficientemente bien comportada como para incluirla en la minimización.

- ▷ **El costo de las tasas de trepado.** De igual manera que con los radios de giro, existe una restricción real sobre el sistema físico de la máxima tasa de trepado que se puede soportar. Si bien esto en principio no está contemplado en la resolución del problema bi-dimensional, cuando se generaliza a resolver problemas tri-dimensionales surge el requerimiento de hacer que estas tasas estén acotadas superiormente. Cabe notar que la posibilidad de que efectivamente estén acotadas las tasas de trepado dependerá de los waypoints ingresados, y el margen para manipular estos valores es en realidad limitado. Para lograr acotar las tasas de trepado, pero sin conocer realmente la diferencia de alturas entre puntos consecutivos (al menos en la sección del programa que realiza la optimización en un plano), se asoció un peso inversamente proporcional al largo de los segmentos de recta en los que el avión deberá realizar un ascenso:

$$sum(v)$$

Cabe notar que en el caso de que se trabaje con las segmentación en planos para waypoints a distinta altura, esta sumatoria se hace solo en los segmentos de recta que implican un cambio de altura. El gran debe de esta ponderación es que valora de igual manera todos los segmentos con cambio de altura, ya sea dicho cambio de 100m como de 10cm.

- ▷ **El costo de tener  $v_{max} > 0$ .** Al igual que con los radios, para que la trayectoria sea coherente es necesario que se respete una desigualdad con los parámetros  $v_{max}$ . Para garantizar esto, se agregó un costo asociado al valor de  $v_{max}$  según la ecuación:  $\min(v_{max}, 0)^2$ . De alguna manera es posible ver que esta función redundante con el costo de las tasas de trepado y si bien posiblemente nunca sea realmente un costo si el sistema funciona adecuadamente, parece razonable dejarlo como "salvaguarda". También es necesario notar que esta desigualdad -a diferencia de la de los radios- debe mantenerse siempre y es independiente del avión con el que se vuela.

Cabe destacar que tanto para las tasas de trepado como para los radios de giro y los  $v_{max}$ , si bien se busca no superar ciertas cotas impuestas por el sistema físico, en realidad no se puede garantizar que la optimización cumpla con las desigualdades deseadas. Para lograr esto, una vez que se obtiene el resultado del generador de trayectorias se imponen las condiciones necesarias si estas no fueran cumplidas, sacrificando optimalidad de la solución. Por detalles adicionales de este proceso ver las secciones siguientes.

- ▷ **La ponderación de los costos.** Finalmente para lograr una buena optimización es necesario hacer una ponderación de los distintos costos de manera de balancearlos adecuadamente. Básicamente lo que se decide con esto es la resolución de los compromisos entre los distintos objetivos de la minimización: ¿es aceptable agrandar un radio de giro a costas de alargar la trayectoria? El compromiso se cuantifica mediante coeficientes para cada uno de los cuatro ítems anteriores. El valor adecuado para estos coeficientes se halló experimentando con distintas combinaciones.

## El algoritmo de minimización.

La minimización una vez que se tiene la función de costos antedicha se implementó usando una función de Matlab muy común, llamada `fminsearch`. Dicha función toma un conjunto de parámetros (condición de parada, máximo número de iteraciones, etc.) y la función de costos e intenta hallar un mínimo mediante el

algoritmo descrito en las secciones anteriores. Sin embargo, para lograr una buena convergencia fue necesario implementar el algoritmo de manera "inteligente", ya sea eligiendo adecuadamente las variables en las que iterar, como aplicando el algoritmo sucesivas veces para lograr una fácil convergencia.

Respecto al conjunto de variables en las que se itera, la elección no es trivial y resultó ser un elemento fundamental para facilitar cierto procesamiento de la trayectoria. A modo de recuento, inicialmente se tomó como variables de iteración al conjunto  $xr_i, yr_i, phi_i, v_{N-1}$  y si bien este conjunto ofrece algunas propiedades deseables como el desacoplamiento entre distintos segmentos de la trayectoria (la recta  $i$ -ésima y el círculo  $i$ -ésimo quedan determinados con solo conocer los valores  $xr_i, yr_i, phi_i, xr_{i+1}, yr_{i+1}, phi_{i+1}$ ), también tiene algunas desventajas, como que no es sencillo el despeje de las cantidades que figuran en la función de costos (distancias a los waypoints, radios de giro, largos de los segmentos de recta) y por lo tanto es difícil garantizar, entre otras cosas, que la semilla expresada en estas variables cumpliera con la restricción de los radios de giro (nótese que lo difícil no es el chequeo de si cumple, sino la construcción de la misma). Como un conjunto de variables alternativo (y que finalmente fue el utilizado) se tomó el conjunto  $xr_0, yr_0, phi_0, v_i, w_i, R_i$ , que tiene una relación bastante más directa con las cantidades que aparecen en la función de costos (y con la construcción de la trayectoria total), pero que sin embargo pierde la ventaja del desacoplamiento entre distintos tramos de la trayectoria (por ejemplo, para conocer el último segmento de recta de la trayectoria es necesario conocer todo el conjunto de parámetros variables). Debe notarse como en cualquiera de los casos el número de parámetros para iterar es  $3N - 2$ .

Finalmente para la llamadas al propio algoritmo de minimización, ocurrió la idea de realizar una minimización por etapas: dados  $N$  waypoints, encontrar inicialmente (siempre a partir de la semilla) una trayectoria óptima para los 3 primeros waypoints, y luego usar esa trayectoria encontrada como una semilla mejorada para optimizar con 4 waypoints y así sucesivamente hasta encontrar la solución óptima para los  $N$  waypoints dados. Los motivos para esto fueron esencialmente dos: el primero es que la minimización en muchos parámetros puede tomar mucho tiempo e incluso converger a soluciones 'malas' (o al menos peores de lo que hace de otra manera) si la semilla no es suficientemente adecuada. El otro motivo surgió del conjunto de variables que se usaron para iterar: si dichas variables no permiten el desacople de la trayectoria en tramos, surge que, por ejemplo, el valor  $v_0$  al modificarse altera toda la trayectoria y por lo tanto hace que el sistema sea desde el punto de vista numérico algo más inestable. Es necesario notar que esta minimización "de a pasos" efectivamente mostró mejores resultados que la minimización en todos los waypoints juntos partiendo de la semilla que se tenía.

#### **4.3.5. Tratamiento de los puntos a distinta altura (generalización del programa a 3D).**

Como se dijo anteriormente, a pesar de habernos planteado la resolución de un generador de trayectorias en un espacio geométrico tridimensional, resolvimos inicialmente la generación de trayectorias en un espacio bidimensional. La mayor ventaja de esto era librarnos de algunos parámetros y variables, reduciendo así la complejidad computacional. Adicionalmente, las trayectorias alrededor de las cuales se puede linealizar el sistema son rectas o círculos contenidos en un plano horizontal, lo que ya da una idea de la mayor simpleza que tiene el problema bidimensional.

Para lograr llevar el problema a tres dimensiones, se tomó la solución bidimensional, y en particular entre dos waypoints a distinta altura, se tomó el segmento de recta que hay entre ellos. En este segmento, calculamos la velocidad de ascenso necesaria para que el avión pase de la altura del waypoint anterior a la altura siguiente y la asignamos al mismo. De este modo logramos un manejo de las variaciones de altura que reutiliza toda la trayectoria generada para dos dimensiones con solo algunas modificaciones mínimas.

Para simplificar aún más esta etapa, no se consideraron todas las diferencias de altura entre waypoints consecutivos, sino que se clasificó a los mismos en grupos de waypoints consecutivos con altura similar (con una tolerancia en estas variaciones de altura) y se trató a cada uno de estos grupos como si estuvieran contenidos en un plano. De este modo se reduce aún más la complejidad computacional de la etapa, pero más importantemente se reducen los cambios de altura que implican trayectorias más exigentes del punto de vista de las entradas del sistema, en particular la velocidad de giro de la hélice. De todos modos la implementación

se realizo de tal manera que es posible considerar la separación por planos o no, según un parámetro fijado por el usuario.

#### **4.3.6. Post-procesamiento de la solución optimizada.**

Una consecuencia del desarrollo que se hace para obtener una trayectoria optimizada es que no necesariamente se respetan ciertas restricciones físicas que el sistema tiene. Las más relevantes de estas son: los radios mínimos de giro, la tasa máxima de ascenso/descenso, y la continuidad de la velocidad en la trayectoria. Para solucionar este inconveniente, se procedió a hacer un post-procesamiento de la solución que se obtuvo hasta el momento (y que denominaremos ".optimizada").

Básicamente el post-procesamiento consiste en "forzar" todas las cantidades anteriormente mencionadas para que sus valores queden dentro de los rangos aceptables para el sistema físico. Obviamente esto se hace con el costo de perder optimalidad en la solución, pues no aseguramos que mediante este forzamiento se siga pasando tan cerca de los waypoints como antes, o que la trayectoria no se alargue incesantemente. La motivación para hacer esto a pesar de la pérdida de optimalidad radica en el costo computacional de los problemas de optimización con restricciones, un costo que consideramos excede largamente la optimalidad perdida en el proceso.<sup>d</sup>

En el caso de los radios de giro, si alguno de los mismos no alcanzara el valor del radio mínimo, se fuerza su valor manteniendo los segmentos de las rectas tangentes al círculo en cuestión aunque cambiando su largo, y manteniendo todo el resto de la trayectoria incambiado.

Para el caso de las tasas de ascenso y descenso la cuestión no es tan trivial, pues disminuir una tasa de ascenso en particular y mantener el resto causa que se vea afectada toda la trayectoria que precede al segmento cuya tasa de ascenso es cambiada. Para ajustar esto es necesario entonces hacer un recálculo de todas las tasas de ascenso para que, sin obviar las restricciones, se llegue a las alturas de vuelo deseadas.

En cuanto a las discontinuidades en la velocidad del sistema, que surgen cuando se generaliza la trayectoria a 3D y por lo tanto necesariamente se pasa de velocidades contenidas en planos horizontales (cuando se recorren círculos) a velocidades con componente según  $\hat{z}_I$  no se implementó ninguna solución. Esto tiene dos justificaciones: por un lado el controlador que termino por implementarse mostró buena tolerancia a estas discontinuidades haciendo innecesario su tratamiento previo y por otro lado tampoco las discontinuidades podían eliminarse del todo, aunque si podrían haber sido suavizadas mediante el agregado de componentes según  $\hat{z}_I$  para la velocidad en los círculos (lo que efectivamente los hubiera convertido en tramos helicoidales).

### **4.4. Análisis de los resultados**

En esta sección hacemos un estudio de los resultados obtenidos con el generador programado, haciendo énfasis en definir indicadores cuantitativos para realizar las medidas.

#### **4.4.1. Criterios de éxito.**

En primera instancia se procedió a definir los criterios de éxito, entendiendo a los mismos como el conjunto de magnitudes que se quiere cuantificar para comparar respecto de lo esperado. A continuación se enlistan las magnitudes elegidas (y no las funciones cuantificadoras de las mismas).

---

<sup>d</sup>Se llegaron a probar incluso algunos algoritmos que realizan minimizaciones con restricciones y aunque los mismos daban resultados muy buenos y en muy poco tiempo, su código era propietario (no solo el software que lo implementaba sino el código en sí) y el costo de adquirirlos no era particularmente bajo. Sin embargo el equipo del proyecto considera que los resultados obtenidos con el algoritmo que se utilizó -si bien no garantizan la optimalidad con restricciones- muestra resultados más que satisfactorios.

- ▷ **Distancias a los waypoints pequeñas.** Quizás el más obvio de los criterios de éxito, para que la trayectoria sea aceptable debe pasar suficientemente cerca de los waypoints dados por el usuario.
- ▷ **Radios de giro suficientemente grandes.** Un criterio de éxito que surge de la restricción física del sistema. Debe tenerse en cuenta que si los radios de giro quedan por debajo de cierta cota, se los fuerza aun valor mínimo, con lo que en realidad este se convierte en un criterio que siempre debe satisfacerse (posiblemente a costas de otros). Sin embargo nos interesa evaluar cuan lejos de estos valores mínimos y –que como dijimos, se fuerzan- estaba la solución optimizada devuelta por el programa. De este modo se cuantifica la suboptimalidad de la trayectoria final.
- ▷ **Largo de la trayectoria no excesivo.** Lo deseable siempre será que la trayectoria sea de lo más corta posible, y surge por lo tanto el criterio de comparar contra el mínimo largo de la trayectoria dados los waypoints.
- ▷ **Tasas de trepado acotadas.** Otra restricción física del sistema que debe cumplirse para que una trayectoria sea realizable por el sistema físico. Una vez más, esta condición se fuerza sobre la trayectoria resultante si fuera necesario, a costas de obtener una trayectoria subóptima. Del mismo modo, nos interesa evaluar cuan lejanos eran los valores propuestos después de la optimización, con respecto a los valores mínimos aceptables. Al igual que en el caso de los radios de giro, esto es un indicador de la suboptimalidad de la trayectoria final y no de que efectivamente estemos exigiendo tasas de ascenso/descenso por encima de lo posible.
- ▷ **La solución para N+1 puntos 'incluye' la solución para N puntos.** Este criterio de éxito surge más como una prueba de buen comportamiento del programa que como una condición deseable del mismo. Con este criterio nos queremos referir a que, si la optimización es buena, el resultado de la trayectoria para pasar por los primeros N waypoints, cuando se dan N+1 waypoints, debería ser muy similar al resultado que se obtiene cuando simplemente se ingresan los primeros N waypoints como entrada al programa.

#### 4.4.2. Indicadores cuantitativos.

Se procedió entonces a definir indicadores cuantitativos de los criterios anteriormente mencionados.

- ▷ **Distancias a los waypoints pequeñas.** En este caso simplemente consideramos la máxima distancia euclídeana entre la trayectoria trazada y los waypoints dados por el usuario. Esta magnitud esperamos que nunca exceda los  $5m$  para satisfacer las condiciones de diseño<sup>6</sup>.
- ▷ **Radios de giro suficientemente grandes.** Aquí consideramos una triple cuantificación, el radio mínimo de la solución 'óptima', el radio medio y la mediana del conjunto de radios. La justificación radica en que nos interesa tanto tener una noción de cuan lejos están los radios en general respecto de lo deseado y, en particular, el peor caso. Valores aceptables para estas cantidades sería un 95 % de la cota para el radio mínimo y el 100 % de la cota para la media y mediana. Las cotas tan exigentes surgen de que en el sistema real se requiere que se respete una desigualdad estricta, y aunque las mismas se aseguran en el 'post-procesamiento', idealmente la trayectoria no debería alejarse mucho de la óptima.
- ▷ **Largo de la trayectoria no excesivo.** Para esto consideramos el largo -mediante una norma euclídeana- de la trayectoria, menos la sumatoria de la distancia entre un waypoint y el siguiente. Básicamente esto es el largo de la trayectoria dada menos el mínimo largo que la trayectoria debe tener para pasar por todos

---

<sup>6</sup>Como se comenta en la sección de resultados, esta cantidad está algo comprometida (en particular la distancia a los dos primeros waypoints) por el punto de comienzo y la dirección inicial de la trayectoria, elementos que si bien pueden ser optimizables, fueron dejados fijos en la implementación usada a la hora del testeo.

los waypoints<sup>f</sup>. Un valor aceptable para esta cantidad sería el 20 % del largo mínimo de la trayectoria. Esta cantidad surge de un análisis más bien arbitrario de cuán larga puede ser la trayectoria real.

- ▷ **Tasas de trepado acotadas.** Al igual que en el caso de los radios, consideramos el valor mínimo, mediano y medio de la tasa de trepado en la solución 'óptima'. Los motivos para la elección y los valores aceptables también son los mismos. Cabe notar aquí que el valor de las tasas de trepado en la solución 'óptima' no es demasiado modificable en el proceso de optimización (depende fuertemente de los waypoints), pero se puede lograr algún tipo de minimización.
- ▷ **La solución para N+1 puntos 'incluye' la solución para N puntos.** En la última implementación del software de generación de trayectorias este comportamiento casi que se impone sobre el sistema, pues se usa la solución para N waypoints como semilla para la solución de N+1 waypoints, con lo cual pierde sentido la cuantificación de este indicador.

### 4.4.3. Casos de prueba y resultados

Para evaluar los resultados del generador se construyeron conjuntos de waypoints bastante arbitrarios, que no siempre cumplían con las restricciones pretendidas para los mismos (distancia mínima entre puntos, etc.), y también se evaluó el uso de distintos coeficientes de ponderación para la función de costos, de modo de visualizar como se ven afectados los distintos 'equilibrios' que implica hallar una solución óptima. En lo que sigue se detallarán cualitativamente (y mínimamente de forma cuantitativa) los resultados obtenidos. En caso de querer indicadores más amplios o detalles particulares de las condiciones de testeo, se recomienda la lectura del apéndice C.

En la figuras que siguen se muestran algunos casos de prueba para el generador de trayectorias y la salida obtenida finalmente: *Test1<sub>sem</sub>.fig* *Test1<sub>opt</sub>.fig* *Test1<sub>Fin</sub>.fig* *Test3<sub>sem</sub>.fig* *Test3<sub>opt</sub>.fig* *Test3<sub>Fin</sub>.fig*

Como se aprecia en la figura los resultados son satisfactorios, siendo las trayectorias muy similares a las esperadas. De los ensayos realizados se desprenden resultados que cumplen en general con los estándares cuantitativos fijados anteriormente. En cuanto a las distancias a los waypoints, se lograron resultados que indican que la distancia media a los waypoints puede perfectamente mantenerse menor a los 20m para todos los waypoints. En general los mayores problemas surgen cuando se quiere imponer sobre la trayectoria la dirección inicial, pues esto condiciona fuertemente el paso por los dos primeros waypoints, o cuando los waypoints son muy cercanos entre sí.

En cuanto a los radios de giro, los resultados fueron más que satisfactorios, ya que la desigualdad deseada ( $R > R_{min}$ ) se respetó casi sin excepciones, y cuando las hubo, las mismas no superaban el 1 %. Es necesario indicar que estos resultados dependieron fuertemente de la implementación que finalmente se hizo para la optimización, donde los radios de giro en la 'semilla' se forzaron al valor de  $R_{min}$  antes de comenzar la minimización.

El largo total de las trayectorias generadas no siempre satisfizo el criterio fijado a priori de ser únicamente 20 % mayor que el largo mínimo, pero en todos los casos fue cercano al mismo. De todas maneras consideramos exitoso el criterio debido a la arbitrariedad con la que se había elegido el parámetro.

Las tasas de trepado fueron posiblemente el elemento más conflictivo en la generación de trayectorias. Por un lado, las tasas de trepado quedan casi dadas ya en el momento de ingresar los waypoints al generador, ya que en cierta forma estamos pidiendo -al pasar de un waypoint al siguiente- que el avión modifique su altura de vuelo según la altura de los distintos waypoints. Sin embargo, es posible afectar algo las tasas de trepado pues lo que sí es modificable es la distancia en la que el avión tiene que cambiar su altura de vuelo. En estas circunstancias consideramos que fue más que aceptable el comportamiento del generador durante los ensayos, que permitió afectar de cierta manera las tasas de trepado (y mantenerlas dentro de las cotas admisibles). Sin

<sup>f</sup>El largo mínimo aquí lo definimos como el largo de la trayectoria que va desde un waypoint al siguiente mediante un segmento de recta. Si bien este largo es mínimo en cierto sentido, pueden perfectamente existir trayectorias que sean más cortas, siempre y cuando no se exija a las mismas pasar estrictamente por todos los waypoints.

embargo todo el tratamiento que se hizo está basado en el hecho de que los cambios de altura solo se dan en los tramos rectos de la trayectoria, lo cual no tiene que ser necesariamente así. Queda pendiente para desarrollos futuros el hecho de trabajar con cambios de altura en los tramos circulares (que pasarían a ser helicoidales) lo que en principio permitiría obtener mejores resultados (trayectorias más suaves y tasas de trepado menores).

A modo de comentario final cabe destacar como una 'buena' trayectoria en general se obtiene mediante el compromiso entre todas las cantidades que se mencionaron, lo que hace necesario poner énfasis en la elección de los coeficientes de ponderación para la función de costos. Por ejemplo, agrandar el peso de la distancia a los waypoints necesariamente trae aparejado trayectorias más largas, así como tratar de minimizar las tasas de trepado reduce los radios de giro, y disminuir el largo de la trayectoria aleja a la misma de los waypoints y reduce también los radios.

#### **4.4.4. Conclusiones**

Los resultados que se han mostrado a lo largo de esta sección así como otros que pudimos ver durante la incontable cantidad de ejecuciones del generador de trayectorias que hemos hecho durante el proyecto, nos dejan la sensación de que el mismo es suficientemente robusto y provee soluciones más que adecuadas. En particular los pocos problemas que quedarían para solucionar vinculados con el mismo serían: mejorar los tiempos de ejecución para una implementación eficiente en un procesador de características más reducidas que las de un PC de escritorio, y agregar algún tipo de tratamiento o mejora del algoritmo cuando los waypoints que se dan son demasiado cercanos (este es uno de los pocos puntos en los que un algoritmo propietario probado resultó tener resultados evidentemente mejores que los del algoritmo que se terminó por usar.)

---

## Capítulo 5

# Linealización

---

Hasta ahora se venía discutiendo sobre como es el MVE completo - aunque, claro está, con las salvedades de los efectos no considerados que se detallaron en la sección 3.6. -. Este modelo, como se puede desprender fácilmente del resumen de ecuaciones realizado en la sección 3.7., es un sistema de alta complejidad en cuanto a que se trata de un sistema de orden doce (doce variables de estado, doce componentes del vector  $x$ ) y en extremo no lineal, el cual será muy útil a efectos de la construcción del simulador, lo cual se explicará en un capítulo aparte. Sin embargo, a efectos del diseño y la construcción del controlador, encontramos que un primer abanico de herramientas a utilizar son las que nos brinda la teoría de control lineal. Pero, claro está, para esto es necesaria la construcción de un nuevo sistema; un sistema linealizado. Por suerte no estamos trabajando con un sistema intrínsecamente no lineal, por lo cual podemos encontrar linealizaciones en torno a determinadas trayectorias. Para aclarar este último punto, no existe una linealización genérica del MVE presentado, y el mismo tampoco es linealizable para cualquier trayectoria, sólo para algunas trayectorias sencillas que se explicarán más adelante. De esta manera, para poder trabajar con el sistema buscaremos linealizar las ecuaciones halladas alrededor de una trayectoria estacionaria, es decir, trabajaremos con las ecuaciones que cumplen las pequeñas perturbaciones alrededor de estas trayectorias.

Recordamos que la linealización jacobiana para un sistema de la forma:

$$\dot{x} = f(x, u) \quad (5.1)$$

Se obtiene realizando el desarrollo de *Taylor* alrededor de vectores  $x_0$  y  $u_0$  que determinan el punto de operación del sistema, o más genéricamente, la trayectoria de operación pues en principio tanto  $x_0$  como  $u_0$  podrían incluso ser funciones del tiempo, y que cumplen:

$$\dot{x}_0 = f(x_0, u_0) \quad (5.2)$$

Entonces, derivando para los doce elementos del vector de estado  $x$  que denotaremos con los subíndice  $i$  y  $j$ , para los cuatro elementos del vector de entradas controlables  $u$  los cuales denotaremos con el subíndice  $k$ , y evaluando dichas derivadas en el punto de operación resulta que la linealización se obtiene de la siguiente manera:

$$\dot{\tilde{x}}_i \simeq \sum_{j=1}^{12} \left( \left. \frac{\partial f_i(x, u)}{\partial x_j} \right|_{(x_0, u_0)} \tilde{x}_j \right) + \sum_{k=1}^4 \left( \left. \frac{\partial f_i(x, u)}{\partial u_k} \right|_{(x_0, u_0)} \tilde{u}_k \right) \quad (5.3)$$

Siendo  $\tilde{x}_j$  y  $\tilde{u}_k$  pequeños apartamientos de la trayectoria de operación  $x_{j0}$  y  $u_{k0}$  respectivamente, y quedan definidos por  $\tilde{x}_j = x_j - x_{j0}$  y  $\tilde{u}_k = u_k - u_{k0}$ . De modo de clarificar el conjunto de los  $\{x_i\}$  es de la forma  $\{x_i\} = \{x, y, z, \varphi, \theta, \psi, v_{Bx}, v_{By}, v_{Bz}, w_{Bx}, w_{By}, w_{Bz}\}$ , mientras que el conjunto de los  $\{u_k\}$  es de la forma  $\{u_k\} = \{\delta_a, \delta_e, \delta_r, n\}$ . Aplicando este procedimiento se obtiene finalmente un sistema linealizado de la forma:

$$\dot{x} = A|_{(x_0, u_0)} \tilde{x} + B|_{(x_0, u_0)} \tilde{u} \quad (5.4)$$

Siendo nuevamente  $\tilde{x}$  y  $\tilde{u}$  pequeños apartamientos de la trayectoria de operación  $x_0$  y  $u_0$  respectivamente, y quedan definidos por  $\tilde{x} = x - x_0$  y  $\tilde{u} = u - u_0$ . Usando este procedimiento para el MVE presentado en el capítulo anterior, se obtienen las matrices  $A$  y  $B$  para la linealización de dicho modelo.

## 5.1. Trayectorias estacionarias y puntos de operación

Como se desprende de lo expresado anteriormente, otro concepto importante que aparece en esta instancia es el concepto de trayectoria estacionaria. Estrictamente lo que estamos buscando son trayectorias tales que si linealizamos alrededor de las mismas, nos quede un sistema lineal invariante en el tiempo (SLIT de ahora en más). Dado que nuestro MVE no depende explícitamente del tiempo, para obtener un SLIT es necesario que las derivadas antes mencionadas en las variables de estado  $x$  y en las entradas  $u$  no varíen en el tiempo al ser evaluadas en las trayectorias de operación  $x_0$  y  $u_0$ . Por lo tanto, llamamos trayectorias estacionarias, y denotaremos como  $x_d$  y  $u_d$ , a aquellas soluciones del MVE que hacen que las ecuaciones 2.4, 2.7, 2.10 y 2.12 sean invariantes en el tiempo.

Dada la dependencia de las ecuaciones con la velocidad lineal y la velocidad angular expresadas en la base del SB, se puede ver que para las trayectorias buscadas se deben anular las ecuaciones 2.10 y 2.12. Es claro que no buscamos puntos de equilibrio del sistema, es decir soluciones para las cuales la derivada del vector de estados se anule para todo tiempo, dado que para esto sería necesario anular las ecuaciones 2.4 y 2.7, obteniéndose por lo tanto la trayectoria trivial (e imposible para una situación real) en que el avión se mantiene en reposo para todo tiempo. Buscamos pues, trayectorias para las cuales la velocidad lineal y la velocidad angular se mantienen constantes en el SB <sup>a</sup>.

Matemáticamente esto sería lo que se expone en las ecuaciones 5.9 y 5.10 que se muestran a continuación, lo cual constituye la característica fundamental de lo que llamaremos trayectorias estacionarias:

$$\boxed{\dot{v}_B = 0 \quad \Rightarrow \quad v_{Bx} = 0, v_{By} = 0, v_{Bz} = 0} \quad (5.5)$$

$$\boxed{\dot{w}_B = 0 \quad \Rightarrow \quad w_{Bx} = 0, w_{By} = 0, w_{Bz} = 0} \quad (5.6)$$

Si imponemos las restricción de las ecuaciones 5.9 y 5.10 sobre las ecuaciones 2.10 y 2.12, observamos que implica anular el lado izquierdo es estas últimas dos ecuaciones, con lo cual se obtiene que se debe cumplir el resultado que se muestra en las ecuaciones 5.7 y 5.8.

$$F_{Neta}^{\vec{}} - m \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times \begin{pmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{pmatrix} = 0 \quad (5.7)$$

$$T_{Neta}^{\vec{}} - \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times I_{GB} \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} = 0 \quad (5.8)$$

Con este criterio, las trayectorias estacionarias posibles para este sistema son las que aparecen en la lista que se muestra a continuación. Observamos que esto es posible debido a que las ecuaciones dinámicas 2.10 y 2.12 no dependen explícitamente de la posición en el sistema inercial ni de los ángulos de *Euler*. En este punto, para que esta última afirmación sea correcta para todas las trayectorias estacionarias listadas, deberemos despreciar la velocidad del viento respecto de la velocidad del avión para el segundo ítem.

▷ Vuelo en línea recta velocidad constante

<sup>a</sup>Lo que si serían puntos de equilibrio para las ecuaciones 2.10 y 2.12, es decir, restringiéndonos a un subespacio del MVE

▷ Giro a velocidad angular constante

Sin embargo, debe notarse que las ecuaciones cinemáticas 2.4 y 2.7 sí dependen explícitamente de los ángulos de *Euler* (no así de la posición en el SI) por lo que estas trayectorias no pueden ser calificadas todavía de estacionarias. Un estudio más detallado de la ecuación 2.7 permite ver que existe dependencia de solamente dos de los tres ángulos de euler, por lo que si se tiene que  $\theta = cte$  y  $\varphi = cte$ , entonces esta ecuación pasa a ser invariante en el tiempo. En lo que sigue haremos un estudio de las trayectorias mencionadas anteriormente y veremos en que casos se podrán considerar como trayectorias estacionarias.

## 5.2. Vuelo en línea recta a velocidad constante

La primer trayectoria estacionaria que vamos a analizar es el vuelo en línea recta a velocidad constante en el SI, o también llamado vuelo rectilíneo nivelado. Dicha trayectoria estacionaria se caracteriza por las condiciones:

$$\boxed{\vec{v}_B = cte \quad \Rightarrow \quad v_{Bx} = cte, v_{By} = cte, v_{Bz} = cte} \quad (5.9)$$

$$\boxed{\vec{w}_B = cte = 0 \quad \Rightarrow \quad w_{Bx} = 0, w_{By} = 0, w_{Bz} = 0} \quad (5.10)$$

Estas condiciones, claro está, cumplen con las ecuaciones 5.9 y 5.10 correspondientes a la definición de una trayectoria estacionaria. Como estamos bajo las hipótesis de  $w_{Bx} = w_{By} = w_{Bz} = 0$ , las ecuaciones 2.10 y 2.12 se reducen a  $F_{Neta}^{\vec{}} = 0$  y  $T_{Neto}^{\vec{}} = 0$ . Esta condición implica necesariamente que el vuelo rectilíneo nivelado no va a ser posible para un valor cualquiera de velocidad, ni tampoco para un valor cualquiera de giro de la hélice por ejemplo. Dado que tanto la fuerza de *lift* como la fuerza de *drag* dependen del módulo de la velocidad relativa al viento, va a ser necesario, por un lado suficiente velocidad como para que el *lift* pueda contrarrestar al peso así como una potencia del motor suficiente como para 'vencer' al *drag*, a lo que se deben tener en cuenta las limitaciones físicas del motor. Sin embargo, las magnitudes recién mencionadas no son las únicas involucradas en estas dos ecuaciones. Elaborando un poco más sobre este tema, se tienen para la fuerza neta y el torque neto las ecuaciones 5.11 y 5.12.

$$F_{Neta}^{\vec{}} = \vec{F}_a + mg\hat{k}_I + \vec{F}_p = 0 \quad (5.11)$$

$$T_{Neto}^{\vec{}} = \vec{T}_a + \vec{T}_p = 0 \quad (5.12)$$

Donde tenemos, para las distintas fuerzas y momentos, las relaciones funcionales dadas según las ecuaciones definidas en el capítulo anterior, en la sección 3.5. de fuerzas y momentos involucrados. Otra cosa que cabe mencionar es que se debe recordar que los ángulos de *Euler* son necesarios para obtener la expresión de la velocidad del viento en el SB, con lo cual al introducir la velocidad del viento como variable en el problema se están introduciendo dependencia con los ángulos de Euler. Sin embargo, en esta trayectoria estacionaria se verá que los ángulos de *Euler* son constantes, y por lo tanto esto no introducirá una dependencia en el tiempo, y por lo tanto no deberemos para esta trayectoria hacer ninguna suposición sobre la velocidad del viento.

$$F_{Aero}^{\vec{}} = F_{Aero}^{\vec{}}(v_{Bx}, v_{By}, v_{Bz}, w_{Bx}, w_{By}, w_{Bz}, |\vec{v}_{VientoB}|(\varphi, \theta, \psi), \alpha, \beta, \delta_{elevador}, \delta_{alerones}, \delta_{timon}) \quad (5.13)$$

$$F_{Motor}^{\vec{}} = F_{Motor}^{\vec{}}(v_{Bx}, v_{By}, v_{Bz}, |\vec{v}_{VientoB}|, n) \quad (5.14)$$

$$T_{Aero}^{\vec{}} = T_{Aero}^{\vec{}}(v_{Bx}, v_{By}, v_{Bz}, w_{Bx}, w_{By}, w_{Bz}, |\vec{v}_{VientoB}|(\varphi, \theta, \psi), \alpha, \beta, \delta_{elevador}, \delta_{alerones}, \delta_{timon}) \quad (5.15)$$

$$T_{Motor}^{\vec{}} = T_{Motor}^{\vec{}}(v_{Bx}, v_{By}, v_{Bz}, |\vec{v}_{VientoB}|, n) \quad (5.16)$$

$$m\vec{g} = mg\hat{k}_I = mg(-\sin\theta\hat{i}_B + \cos\theta\sin\varphi\hat{j}_B + \cos\theta\cos\varphi\hat{j}_B) \quad (5.17)$$

Estas últimas relaciones son válidas en general, aunque en el caso que se está tratando, de movimiento rectilíneo nivelado, se impone que  $w_{Bx} = w_{By} = w_{Bz} = 0$  esto no será así en otros casos que se tratarán mas adelante para los cuales también se utilizaran estas ecuaciones.

Si miramos las dependencias de las fuerzas, se puede ver claramente que las ecuaciones 5.11 y 5.12 involucran a las siguientes variables: los ángulos de Euler, las componentes de la velocidad del rígido en el SB, las posiciones de las superficies de vuelo, la velocidad de la hélice y la velocidad del viento. Observamos pues, que para que pueda existir el vuelo recto nivelado, se deben cumplir dos ecuaciones vectoriales que relacionan 13 variables distintas sin vínculos entre ellas. Resumiendo, tenemos tres ecuaciones vectoriales correspondiente a 6 ecuaciones algebraicas para un conjunto de 13 variables. Es claro que de este sistema de ecuaciones solo será posible despejar a lo sumo 6 de estas variables en función de las demás, estas otras deberán entonces ser impuestas de alguna manera para hallar los valores de las 6 variables que se pueden despejar y de esa manera obtener las condiciones en que se realizará el vuelo a velocidad constante, rectilíneo y nivelado. Asumiendo que lo que interesa es despejar para una trayectoria rectilínea dada (velocidad en el sistema inercial conocida), podemos agregar la ecuación de cambio de base del sistema inercial al sistema body, lo que vincula la  $v_B$  con  $v_I$  y los ángulos de Euler, subiendo así a 9 el número de ecuaciones y a 16 el número de variables.

Nuestro interés en particular es, dada una determinada situación de vuelo, hallar como deben ser las acciones de control para que esta sea posible, por lo cual las posiciones de las superficies de vuelo (deltas) y la velocidad de la hélice serán variables que se despejaron de las ecuaciones 5.11 y 5.12 a partir de ciertas condiciones de vuelo fijas de antemano. A partir de este análisis, las condiciones de vuelo que se fijaran serán: la velocidad del viento, la velocidad del avión expresada en el sistema inercial y el ángulo  $\psi$ , dejando 6 variables para 6 ecuaciones.

En relación a la velocidad del viento, parece razonable que la velocidad del viento sea fija dado que es una característica ambiental externa al sistema, si bien podría variar en el tiempo, es claro que no tiene sentido despejarlo como si fuera una variable que podemos controlar. Por otro lado, el sistema es prácticamente indiferente sobre el ángulo  $\psi$  salvo a través de la velocidad del viento en el SB, por lo que también es razonable que este ángulo se fije de antemano. La velocidad del avión en el sistema inercial es lo que determina la situación de vuelo buscada por lo cual es un dato necesario para determinar el problema, y el ángulo  $\psi$  puede ser impuesto en una posición razonable conociendo la velocidad del avión en el sistema inercial <sup>b</sup>. En conclusión, fijadas estas condiciones de vuelo y resolviendo el sistema de ecuaciones algebraico no-lineal dado por las ecuaciones 5.11 y 5.12, obtenemos las seis restantes variables del vuelo:  $\theta, \varphi, \delta_a, \delta_e, \delta_r, n$ .

### 5.3. Giro a velocidad angular constante

Por giro a velocidad constante entendemos las trayectorias que admiten  $w_B = cte$  y  $v_B = cte$ . Como ya se dijo a lo largo de las secciones anteriores, para que el giro a velocidad constante pueda ser una trayectoria estacionaria se requieren de algunas hipótesis adicionales: a efectos de que las ecuaciones 2.10 y 2.12 sean independientes del tiempo, la velocidad del viento debe ser nula o despreciable respecto de  $v_B$ ; también a efectos de que la ecuación 2.7 sea invariante en el tiempo, no puede haber variación temporal de  $\theta$  o  $\varphi$ ; y finalmente para que 2.4 sea invariante en el tiempo se debe pedir que  $\psi$  también sea constante. La primera de estas hipótesis puede tomarse ya que, más allá de que sea una realidad, tomando la velocidad del viento como una entrada externa al sistema, contará como una perturbación en nuestro modelo. Sin embargo, las otras dos hipótesis juntas implican, usando 2.7, que  $w_{Bx} = w_{By} = w_{Bz} = 0$ , lo cual nos llevaría nuevamente a la

<sup>b</sup>Estrictamente el ángulo  $\psi$  no es una entrada conocida de antemano y bien podría dejarse libre a la hora hallar la solución al sistema de ecuaciones que por ser no lineal no necesariamente guarda una relación entre el número de ecuaciones necesarias para resolverlo y variables incógnita. Sin embargo hemos decidido fijarlo para simplificar la resolución del sistema.

situación de vuelo a velocidad constante, lo que es contrario a lo que se busca.

Quedamos entonces ante dos opciones: o bien restringimos las trayectorias de trabajo a vuelo a velocidad constante, lo cual limita fuertemente las posibilidades de vuelo controlado para el avión, o bien modificamos nuestro MVE para que admita otras linealizaciones invariantes en el tiempo. Nosotros optamos por la segunda opción ya que nos permite una mayor flexibilidad de trabajo con el avión, aunque también trae sus complicaciones adicionales. En particular exploramos dos opciones cuyos resultados e implementación se detallan en capítulos posteriores pero cuyos detalles generales explicamos aquí.

La primera opción es la de directamente eliminar la ecuación 2.4 del MVE, lo que es posible pues en nuestro modelo ni las ecuaciones de cinemática ni las de dinámica dependen explícitamente de la posición del avión. La consecuencia que esto trae aparejado es que renunciamos a trabajar posteriormente controlando directamente la posición como variable, y siendo que en el mejor de los casos podríamos controlar su derivada (la velocidad), se obtiene un error integral en la posición. Esto no es completamente cierto pues usando control integral (es decir buscando controlar la integral de nuestro vector de estados) podríamos recuperar las variables, pero esto volvería a traer el mismo problema: no es posible encontrar un sistema lineal invariante en el tiempo para realizar el control. Trabajando con una variante de la idea de control integral surge la segunda posibilidad: cambiar la posición por otra variable de similares características en el MVE, pero cuya cinemática sea invariante en el tiempo para esta trayectoria. Dependiendo de como se implemente esta segunda opción puede tener problemas más o menos similares a la anterior, pero esto se describe en el capítulo de control lineal junto con los resultados obtenidos. En nuestro caso procedimos a sustituir la ecuación 2.4, pasando a medir la posición para el avión desde el centro del círculo que se encuentra recorriendo, y expresando dicha posición en el sistema solidario al avión. Para esta nueva expresión de la posición se obtiene la siguiente ecuación de cinemática:

$$\dot{x}_B = v_B + w_B \times x_B$$

Esta ecuación sustituye a 2.4 para la trayectoria en estudio y permite efectivamente lograr un sistema invariante en el tiempo cuando se linealiza. Cabe notar aquí que los nuevos elementos del vector de estados miden exactamente la misma cantidad que antes (la posición) pero desde un referencial distinto que da lugar a la invarianza temporal.

Para cualquiera de los dos casos (ignorar la ecuación 2.4 o sustituirla por la ecuación, trabajamos en el marco de las primeras dos hipótesis que se mencionaron ignorando temporalmente la tercera de ellas. Quedamos entonces con la necesidad de linealizar las ecuaciones 2.7, 2.10 y 2.12 para obtener el SLIT deseado. El tratamiento de la ecuación 2.4 dependerá de las opciones mencionadas en el párrafo anterior, pero pasaremos a linealizar también la nueva ecuación a efectos de contar con un modelo completo para poder realizar el control.

Aceptando que estamos buscando  $w_B = cte$  y  $v_B = cte$ , las ecuaciones ?? y ?? se sostienen también para esta situación. Hasta aquí tenemos 6 de las mismas ecuaciones del caso anterior, pero 13 variables ya que  $w_B$  es desconocida. Debe considerarse también la ecuación que vincula  $v_I$  con  $v_B$  (OTRA VEZ FALTA REF DE ECUACION), pero para eliminar la dependencia con  $\psi$ , que es una función del tiempo, se evaluará dicha expresión solo para  $\psi = 0$  sin pérdida de información por la simetría de la situación. La última ecuación que debe tomarse para resolver el problema es 2.7, llegando a un total de 12 ecuaciones y 19 variables.

Al igual que en el caso anterior, no todas estas variables son incógnitas: como dijimos al principio de esta sección,  $\theta$  y  $\varphi$  deben ser constantes, lo que elimina sus derivadas como incógnitas; también la velocidad del viento debe ser considerada nula para que el sistema sea LIT, por lo que tampoco es una incógnita; finalmente la condición de vuelo buscada queda determinada por la velocidad angular de giro  $\dot{\psi}$  que debe ser dada y la velocidad del avión en cualquier punto de dicho giro (en nuestro caso usaremos la velocidad cuando  $\psi = 0$  pero como se dijo antes esto no conlleva pérdida de generalidad)<sup>c</sup>. Se obtiene así un sistema de 12

<sup>c</sup>Cabe notar aquí que la velocidad del avión expresada en el sistema inercial para esta situación puede separarse en dos componentes claramente independientes:  $v_{Iz}$  que es constante en la trayectoria ya que la velocidad angular será según el eje  $k_I$ , y por otro lado la velocidad proyectada sobre el plano determinado por los versores  $i_I$  y  $j_I$ , donde la naturaleza de la trayectoria impone que tenga características de un vuelo circular uniforme, es decir, módulo constante

ecuaciones y 12 incógnitas para determinar la situación de vuelo deseada, pudiéndose despejar las variables  $(\theta, \varphi, \delta_{elevador}, \delta_{aleronas}, \delta_{timn}, n, \vec{w}_B, \vec{v}_B)$  y con ellas determinar completamente el sistema linealizado.

Se desprende del análisis anterior que las trayectorias a las que nos referimos como "giro a velocidad constante" son en realidad trayectorias que comprenden situaciones de vuelo helicoidales, con  $v_{Iz} = cte$  y  $w_{Iz} = cte$  siendo el vuelo en círculos un caso especial de dicha trayectoria cuando  $v_{Iz} = 0$ .

A modo de conclusión debe resaltarse que si bien trabajaremos con los dos tipos de trayectoria desarrollados en este capítulo para poder implementar las herramientas de control sobre sistemas lineales invariantes en el tiempo, la segunda trayectoria no puede considerarse estrictamente estacionaria (al menos no con el MVE completo como se planteó en el capítulo de Modelado) por lo que deberá prestarse especial atención al tratamiento dado para dichas trayectorias durante el capítulo de control.

## 5.4. Software de linealización y trimado

Llamamos trimado al proceso de resolver un sistema de ecuaciones, en general no lineal, para el cual se hallan las condiciones en las que sucede una trayectoria estacionaria. Básicamente consiste en un resolvidor del sistema de ecuaciones no lineales que queda planteado a partir de las condiciones que se describieron en las secciones anteriores, donde se detalla el sistema a resolver en cada caso junto con las ecuaciones que lo definen y las incógnitas a despejar. En términos del MVE, con este proceso de trimado se determinan completamente los vectores  $x$  y  $u$  para las situaciones de vuelo deseadas.

A modo de ejemplo, para el caso del vuelo a velocidad constante, el sistema de ecuaciones a resolver es atacado por la rutina *trimVcte.m*, que es una función de matlab que toma como parámetros los valores de las variables que especifican las condiciones antes indicadas (la velocidad del viento, el módulo de la velocidad requerida y el ángulo  $\psi$  especificado) y devuelve los valores de las restantes variables de vuelo para las que sucede la trayectoria estacionaria (las entradas del sistema, los ángulos de Euler restantes, la expresión de la velocidad y velocidad angular en el sistema Body). Dicha rutina utiliza la función *fsolve.m* de matlab para resolver sistemas de ecuaciones no lineales, que viene con el Optimization Toolbox. Cabe destacar que es necesario procurarle una condición inicial de iteración razonablemente cercana a la solución al método para que este asegure la convergencia (en nuestro caso la semilla que se usa fue impuesta luego de resolver el sistema una vez para un conjunto de parámetros estándar).

De manera totalmente análoga funciona la rutina *trimWcte.m* para el caso de trayectorias circulares, y de manera general la rutina *trimTsv.m* que permite el ingreso de una sucesión de puntos de operación que debieran ser vuelos a velocidad constante o giros a velocidad angular constante, y devuelve la sucesión de entradas, ángulos de Euler y demás variables que deben ser determinadas. Una vez que se utilizan estas funciones para obtener los valores de  $x$  y  $u$ , corresponde pasar al proceso de linealización propiamente dicho.

La linealización en nuestro programa se implementó mediante la derivación analítica de las ecuaciones por parte nuestra y luego su programación en Matlab. El motivo principal para realizar la linealización en forma analítica a pesar de que se podrían haber hecho las derivaciones numéricas, radicaba en la futura implementación del controlador en un computador de capacidad más reducida. La derivación numérica, además de ser computacionalmente más cara que la evaluación directa de las funciones que se derivan analíticamente, requiere de conocer un modelo completo del sistema, que si bien llegamos a programar en Matlab, nunca fue la intención de portarlo al computador donde se fuera a eventualmente implementar el sistema de control.

Una vez hechas las derivaciones analíticas, lo que se programó fueron las funciones *linealizarMVE.m* y *linealizarMVE\_modif.m* que calculan dichas funciones derivadas para el MVE presentado en el capítulo de modelado, y para la versión modificada que se introdujo en este capítulo a efectos de obtener un sistema SLIT para las trayectorias con velocidad angular constante. Dichas funciones reciben como entradas los vectores  $x$  y  $u$  y retornan las matrices  $A$  y  $B$  correspondientes al sistema linealizado. A modo de verificar que las derivaciones estuvieran bien hechas, se contrastaron los resultados de estas funciones con lo que resulta de usar la linealización de Matlab correspondiente a los sistemas implementados en Simulink.

# Diseño del controlador

---

Este capítulo lo dedicaremos a estudiar las distintas estrategias de control probadas para el avión y los resultados obtenidos para las mismas. Las estrategias implementadas fueron diversas y algunas de ellas quedaron truncadas en su estudio debido principalmente a dos elementos: el éxito obtenido con otras estrategias y los límites de tiempo del Proyecto. Para lo que sigue se asumirán los resultados de los capítulos de Modelado, Linealización y Generador de Trayectorias.

## 6.1. Gain Scheduling

Como se adelantó en el capítulo de Generación de Trayectorias, la estrategia para trazar las mismas consiste en una concatenación de dos tipos de trayectorias simples y diferenciadas. Estas trayectorias son segmentos de recta y círculos en un plano horizontal, y fueron elegidas por la particularidad de ser trayectorias alrededor de las que la linealización queda invariante en el tiempo (salvando los cambios de modelo necesarios que fueron manifestados en el capítulo de Linealización).

De aquí se deduce lo que va a ser una de las primeras hipótesis para la estrategia de control: realizar el control como una sucesión de controladores que van cambiando a lo largo de la trayectoria, junto con el tipo de trayectoria local deseada. Como ya dijimos, en una primera instancia este control se hace utilizando control lineal alrededor del punto de operación deseado y cambiando dicho punto a lo largo de la trayectoria en la sucesión antedicha. Esta estrategia de control se denomina Gain-Scheduling, y de manera general consiste en, para un sistema no-lineal, linealizar alrededor de varias situaciones o puntos de operación y cambiar de un punto a otro en función del valor de algún conjunto de variables. Sin embargo dicha estrategia es generalizable a controladores no-lineales, pudiendo usar una estrategia tipo Gain-Scheduling para separar el sistema en distintas situaciones (puntos de operación) y usar cualquier tipo de control -no necesariamente lineal- alrededor de los mismos.

En nuestro caso, los puntos de operación alrededor de los cuales se trabajará el sistema son infinitos pero, como ya dijimos, se agrupan en dos conjuntos: los vuelos en línea recta (que quedan determinados por la velocidad del avión y la velocidad del viento) y los vuelos en círculos horizontales (que quedan determinados por la velocidad angular y la velocidad del avión en algún punto de la trayectoria, asumiendo velocidad del viento nula <sup>a</sup>). La variable que se eligió para determinar el paso de una trayectoria a otra es el tiempo, y dicha elección surge naturalmente de la evaluación de la trayectoria que se quiere seguir, ya que conociendo la velocidad angular y velocidad del avión, solo resta conocer el tiempo transcurrido para poder establecer los puntos de cambio de trayectoria.

Todos los puntos de operación se obtienen del generador de trayectorias (ver capítulo dedicado) y en lo que sigue nos dedicaremos a estudiar las distintas estrategias de control que se utilizaron para controlar alrededor de dicho conjunto de puntos de operación.

---

<sup>a</sup>Si bien asumir la velocidad del viento nula parece ser limitante en cuanto al tipo de trayectoria, esta hipótesis es necesaria para que se pueda realizar una linealización invariante en el tiempo alrededor de la trayectoria.

## 6.2. Control lineal

Como se dijo en el capítulo de Linealización, un conjunto de herramientas que poseíamos de antemano para aplicar al control del avión son las de que surgen del control lineal. Este conjunto de herramientas es bastante extenso y permitía sin mayor necesidad de profundizar conocimientos, intentar una primera implementación del controlador. Para lo que sigue se asumirá que, en los distintos puntos de operación del sistema, se cuenta con un modelo SLIT de la situación de vuelo y se partirá desde allí. Esto es:

$$\dot{x} = A\tilde{x} + B\tilde{u}$$

Como ya se vió en el capítulo de linealización, para cada una de las situaciones de vuelo, la existencia del modelo depende fuertemente de como se midan (desde que referencial) las variables de interés. Asumiendo que se harán las modificaciones pertinentes que se indicaron en el capítulo de linealización cuando sea necesario, asumamos en adelante que el modelo SLIT existe para todas las condiciones de vuelo deseadas.

### 6.2.1. Realimentación de estados

En las condiciones de vuelo deseadas existe, a partir del generador de trayectorias, un vector de estados que se desea seguir. Es por esto que surge naturalmente la idea de realimentar la diferencia existente entre el vector de estados deseado y el vector de estados real, es decir, modificar levemente las acciones de control respecto del punto de operación para eliminar las perturbaciones en el vector de estados. Esto se puede expresar matemáticamente como muestra la siguiente ecuación:

$$u_{t\text{echo}} = -Kx_{t\text{echo}}$$

En esta ecuación,  $K$  es la matriz de realimentación que tiene dimensiones transpuestas a  $B$ . Para un sistema SLIT, si se asume una realimentación estática ( $K$  es una matriz de elementos constantes), los polos del sistema realimentado pueden calcularse fácilmente como los valores propios de la matriz  $A - B.K$ . A partir de esto existen varios métodos para calcular la realimentación necesaria de un sistema de modo de colocar los polos del sistema total donde se deseen (root locus) o incluso existen métodos para darle más o menos peso a la realimentación de ciertas entradas y también para darle más o menos peso a las perturbaciones en ciertos elementos del vector de estados (LQR). En cualquier caso es importante considerar el hecho de que se busca el seguimiento de una señal de referencia que varía en el tiempo (más allá de los cambios de punto de operación, existe también una variación permanente de la posición, por ejemplo). Esto requiere entonces que el controlador, más allá de asegurar la estabilidad, pueda asegurar la convergencia de la señal de error a 0, es decir, pueda asegurar que se haga un seguimiento de la señal de referencia.

Para las trayectorias que consideramos en este trabajo, la mayoría de los elementos del vector de estado se mantienen constantes a lo largo de un punto de operación ( $v_B, w_B, \theta, \varphi$  y, en el caso de las rectas,  $\psi$ ), mientras que otras, como la posición expresada en el sistema inercial ( $x_I$ ) varían en el tiempo. Dichas variaciones serán siempre en forma de rampa (derivada constante), por lo que interesa evaluar que el sistema pueda seguir dichas señales sin error asintótico. Previamente al estudio de los distintos tipos de controladores y su diseño, es interesante hacer un análisis de la controlabilidad del sistema.

Por controlabilidad del sistema se entiende a la posibilidad de llevar el sistema desde un estado cualquiera a otro estado cualquiera en un tiempo finito. Sin perder generalidad, se puede ver que dicha condición es equivalente a la posibilidad de llevar el sistema desde un estado cualquiera al origen en un tiempo finito. Usando algunos resultados que se presentan en la bibliografía (REF: OGATA), podemos ver que un sistema como el nuestro es completamente controlables si la matriz  $[B, AB, A^2B, A^3B, \dots, A^{11}B]$  que es -en nuestro caso-  $12 \times 60$ , tiene rango 12. A continuación se detallan los mecanismos utilizados para obtener el controlador y se realiza un análisis de sus ventajas y desventajas.

## Root Loci

Como se dijo anteriormente, por root loci nos referimos al procedimiento de calcular una realimentación para el sistema que coloque los polos del sistema realimentado en lugares predeterminados. Este tipo de mecanismos permite realizar un diseño ajustando características tales como la respuesta en el tiempo, el sobretiro, etc., (considerando las salidas de a una). Una manera fácil de implementar dicho procedimiento en Matlab es mediante una función tal como *place*, y que en realidad no es otra cosa que una función que resuelve un sistema algebraico de ecuaciones no lineales donde hay más incógnitas que ecuaciones. En nuestro caso habría que determinar  $5 \times 12$  elementos de la matriz  $K$  (que corresponden a la realimentación de las 5 entradas en cada uno de los 12 elementos del vector de estados), a partir de las 12 ecuaciones que surgen de imponer:

$$|A - BK - \lambda.I| = 0$$

Tomando  $\lambda$  el valor de los 12 polos que se quieren imponer. Si bien este es un mecanismo fácil de utilizar para lograr la estabilidad e incluso imponer algunas características más sobre el sistema como ya se dijo, no es intuitivo desde el punto de vista de "ver" que es lo que está pasando con el sistema real, qué variables de estado estamos realimentando, cómo, cuál es la relación entre las distintas entradas que pueden moverse, cómo controlar la magnitud de las acciones de control, etc. Si se quiere se puede pensar esto como un root locus en un sistema de 12 salidas, lo que en sí complica bastante algún post-procesamiento de la solución, como puede ser intentar limitar las acciones de control (que necesariamente están limitadas en el sistema real), o intentar ajustar la velocidad de respuesta en ciertas variables de estado. A modo de ejemplo, podemos pensar que si exigimos polos tales que la respuesta del sistema sea muy rápida, necesariamente esto causará que se necesiten acciones de control muy grandes, lo que en el sistema real puede implicar que sean necesarios actuadores muy fuertes o que directamente el sistema se aleje del punto alrededor del cual fue linealizado, lo que claramente no es deseable.

Surge por lo tanto una necesidad de diseñar un controlador que pueda asegurar la convergencia y estabilidad del sistema, así como proveer de alguna forma una limitación al tramaño de las acciones de control. Dado que el mecanismo de ubicar los polos del sistema posee 60 grados de libertad y tan solo 12 ecuaciones, es fácil ver que sobran grados de libertad para agregar exigencias al sistema. Una solución sencilla que permite introducir este tipo de consideraciones -y otras- es la metodología de LQR.

Para completar un sistema de diseño por colocación de polos, falta definir los criterios con los cuales se ubicaran dichos polos. Esta tarea, que en principio puede parecer trivial (definir polos en el semiplano negativo, suficientemente rápidos y con pocas oscilaciones) es en realidad un desafío considerable. Por un lado, la ubicación en sí misma de los polos es difícil de decidir, pues debe de existir un equilibrio entre las exigencias que se hacen a la respuesta temporal del sistema y las limitaciones físicas que sabemos que existen. Por ejemplo, no es razonable exigirle al sistema controlado tiempos de respuesta que sabemos exceden largamente los tiempos característicos de respuesta para un sistema mecánico de nuestras dimensiones (polos en el orden de algunos Hz). De la misma manera, pretender que todos los polos sean reales negativos (no existan raíces complejas conjugadas al polinomio característico) también es algo exigente y en principio no es necesario para el sistema realimentado. Por otro lado, está la ambigüedad que surge al tener más grados de libertad que ecuaciones, lo que implica que la determinación de los polos mediante un algoritmo cualquiera está fijando implícitamente algunas otras características del sistema que pueden no ser deseables. A modo de ejemplo, se probó diseñar un controlador mediante LQR (ver sección siguiente) y calcular los polos del sistema realimentado, para luego diseñar un controlador mediante la colocación de los polos del sistema en los mismos lugares. Los resultados mostraban dos controladores con características totalmente distintas (distintas matrices de realimentación), que si bien podrían haber sido igualmente útiles de haber estado en presencia de un sistema verdaderamente lineal y sin otras limitaciones físicas, terminaron por dar resultados bien diferentes. Esto prueba que la sola asignación de polos no es suficiente para el diseño de un controlador adecuado, al menos cuando el sistema dista bastante de ser modelo SLIT para el cual se hace el diseño.

Como primer criterio para la asignación de polos se tomaron conjuntos de polos reales negativos, donde el polo dominante estuviera en el orden de 1 Hz, y el resto de los polos se mantuvieran entre 1 y 2

décadas por detrás de este. Luego, mediante ensayo y error se probaron otras configuraciones para los polos, con polos complejos conjugados, con respuestas de polo dominante más lentas y también más rápidas. En la sección de análisis de los resultados se detallan más exhaustivamente las situaciones de prueba y sus resultados, pero nos interesa resaltar que para todos los resultados obtenidos -ya fueran buenos o malos-, colocar los polos se convertía más en un 'arte' que en un procedimiento sistemático que permitiera exportar el mecanismo a otras situaciones y/o plantas a controlar. El primer diagnóstico que hicimos para esta situación fue que, dado el alto número de grados de libertad que tiene el problema en comparación con las restricciones que estábamos imponiendo, la función *place* de Matlab necesariamente estaba 'tomando' algunas decisiones por nosotros que desembocaban en un controlador con características distintas a las deseables, más allá de la colocación de los polos en el sistema realimentado. Procedimos entonces a utilizar la ya mencionada metodología de LQR para diseñar el controlador.

## Linear Quadratic Regulator

El segundo procedimiento que utilizamos para hallar la realimentación de estados para el modelo SLIT fue el de LQR y consiste en encontrar también una matriz de realimentación  $K$ , pero mediante un mecanismo absolutamente distinto al anterior: se elijen dos matrices  $Q(12x12)$  y  $R(5x5)$  (que deben ser definidas positivas), y se minimiza la siguiente función:

$$J = \int_0^t (x^T Q x + u^T R u) dt.$$

De esta manera se obtiene un controlador óptimo desde el punto de vista de la función de costo definida anteriormente. Las ventajas de este nuevo sistema son múltiples: por un lado no solo permite el diseño en base a anular la señal de error entre el estado deseado y el estado real, sino que también permite considerar la minimización de las acciones de control necesarias para hacer esto. También permite de alguna manera ponderar cuáles son las variables del vector de estado en las que se quiere tener más tolerancia, y cuales son las entradas en las que se está dispuesto a tener acciones de control más grandes. Por ejemplo, si pensamos las matrices  $Q$  y  $R$  como diagonales, cada uno de los términos de la diagonal ponderaría el error cuadrático medio en una de las variables del vector de estado o en una de las entradas. De esta manera estaríamos, de hecho, haciendo una minimización del error cuadrático integral esperado en las variables de estado y entradas (de allí su nombre), ponderando los errores en cada variable más o menos que los errores en el resto de las variables. Una consecuencia de una ponderación de ese estilo es que se pueden elegir variables de estado en las que los errores son menos tolerables y otras donde la convergencia puede ser más lenta. Algo similar ocurre con las entradas, donde ponderándolas de distinta forma se puede permitir que algunas varíen más y otras menos, o visto desde otro punto de vista, 'concentrar' la energía de la señal de realimentación en algunas variables de entrada y no en otras (algo que es muy útil, ya que por como se miden las deflexiones de las superficies de vuelo, nos interesa que las mismas no superen valores de fracciones de radian, mientras que para las RPM del motor las variaciones admisibles son del orden de cientos).

Antes de adentrarnos en la implementación del método para este caso particular, corresponde hacer un desarrollo sobre la solución matemática para hallar la matriz de realimentación. La minimización planteada anteriormente, sustituyendo  $u = -K \cdot x$  y realizando algunas cuentas matriciales, puede replantearse como la solución de la siguiente ecuación diferencial matricial, que toma la forma de la ecuación de Ricatti:

$$A^t P(t) + P(t)A - P(t)BR^{-1}B^t P(t) + Q = -\dot{P}(t) \quad K = R^{-1}B^t P$$

Donde deben imponerse las condiciones finales para hallar la solución. Existe un caso particular para dicha solución, cuando el horizonte de tiempo es infinito (es decir, se integra en todo el tiempo). En ese caso la solución se obtiene a partir de la siguiente ecuación algebraica matricial:

$$A^t P + PA - PBR^{-1}B^t P + Q = 0$$

$$K = R^{-1}B^t P$$

Para el caso en estudio, dado que los puntos de operación se mantienen por un período finito de tiempo, correspondería trabajar con la primera solución. Sin embargo, dada la complejidad computacional adicional que requiere hallar dicha solución y el hecho de que, a los efectos prácticos, el tiempo de operación alrededor de un punto de operación es mucho mayor al tiempo característico asociado a los polos que queremos para el sistema realimentado (de otra manera no habría garantía de convergencia para cada punto de operación), consideramos más apropiado usar la segunda solución.

Como se desprende de lo anterior, el controlador que se diseñe mediante LQR depende fuertemente de las matrices  $Q$  y  $R$  que se elijan, tópico que aún no hemos atacado en esta documentación. Comenzaremos por la matriz  $Q$ , que cuantifica la minimización en términos del vector de estados. Por simpleza tomamos una matriz diagonal, ya que permite visualizar fácilmente como se ponderan las distintas variables de estado, aunque esto no tiene porque tomarse así (e implica una pérdida de generalidad). Haremos lo mismo para la matriz  $R$ , tomándola diagonal.

Partiendo de esta base, cabe hacer la pregunta: ¿Cuáles de las variables del vector de estado deben o es más importante que tengan errores menores? Surgen entonces dos tipos de respuesta: por un lado, está la escala en la que se miden las distintas variables del vector de estados. Por ejemplo, en términos de los ángulos de Euler, variaciones de pocas décimas de radian son errores significativos, mientras que para la posición, errores de algunos metros son tolerables y esperables. Por otro lado, debe considerarse que variables afectan de mayor manera el sistema a controlar.

A modo de ilustrar este último punto, es necesario que volvamos a la perspectiva del MVE completo. Dicho modelo tiene dependencias no lineales en casi todas las variables del vector de estados y a la hora de realizar la linealización se asume una hipótesis clave, la de trabajar alrededor de un punto de operación sin 'alejarse demasiado' del mismo. Pero conceptos como 'alejarse demasiado' son relativos, no solo en su cuantificación para el total del sistema, sino en su cuantificación para cada variable de estado. Otra vez a modo de ejemplo tomemos la posición y los ángulos de Euler. Mientras que perturbaciones de décimas de radianes en los ángulos de Euler alteran completamente el modelo (cambia toda la linealización del mismo, pues cambian las expresiones de  $w_B$  y  $v_B$  si asumimos que  $w_I$  y  $v_I$  se mantienen para el nuevo estado), perturbaciones en la posición, por grandes que sean, no cambian en absoluto el MVE completo o su linealización<sup>b</sup>. Esto nos arroja sobre un hecho clave para la elección de la matriz  $Q$ : los errores en la posición no son significativos para la linealización y por lo tanto tampoco lo son para que la realimentación sea adecuada; mientras que errores en las otras variables si lo son, y en especial en los ángulos de Euler. Casi que inmediatamente se deduce que las ponderaciones de los ángulos de Euler en la matriz  $Q$  son las que deben valer más, y las de la posición las que deben valer menos. Una receta en estas líneas que dió buenos resultados, es la de tomar la matriz  $Q$  como sigue:

$$Q = \begin{pmatrix} 0,1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0,1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0,1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 100, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 \end{pmatrix} \tag{6.1}$$

Obviamente existen variantes a esta matriz que también dieron resultados aceptables, pero esta fue una de las más robustas en funcionamiento. Por ejemplo, aumentar la ponderación asignada a la posición aumenta la precisión en el seguimiento de dichas variables pero el sistema pierde en robustez (perturbaciones

<sup>b</sup>Esto en realidad sucede para las variables  $x$  e  $y$ , mientras que para  $z$  el MVE completo -considerando variaciones de la densidad del aire con la altura por ejemplo- puede llegar a modificarse levemente.

demasiado grandes llevan a la inestabilidad).

Para la matriz  $R$ , las consideraciones a hacer son similares. En particular es relevante la escala de las distintas magnitudes y los órdenes de magnitud de las señales de realimentación que podemos considerar 'tolerables'. En particular las señales para las superficies de vuelo deberían estar acotadas por algunas décimas de radian, y la señal del motor en algunos cientos de RPM, lo cual marca claramente los distintos órdenes de magnitud. Para lograr que efectivamente las señales se mantuvieran en dichos rangos, una matriz  $R$  que funcionó fue la siguiente:

$$R = \begin{pmatrix} 1000, 0, 0, 0, 0 \\ 0, 100, 0, 0, 0 \\ 0, 0, 1000, 0, 0 \\ 0, 0, 0, 100, 0 \\ 0, 0, 0, 0, 1e-4 \end{pmatrix} \quad (6.2)$$

Nuevamente la matriz es solo un ejemplo de matrices posibles para el sistema, en particular buscar una señal de realimentación demasiado pequeña puede causar que los tiempos de respuesta sean bastante más lentos que los esperados. Si se acepta la posibilidad de que las señales 'saturen' en cierto valor (como es nuestro caso donde la saturación se implementó de hecho en el sistema a simular), entonces es posible trabajar con matrices que producen realimentaciones 'óptimas' que son más grandes que las aceptables pero que nunca llegan como tales a los actuadores.

Para terminar la sección correspondiente al método del regulador cuadrático lineal o LQR,

Para cualquiera de los dos procedimientos anteriores (asignación de polos y LQR) el resultado es -a grandes rasgos- el mismo: un sistema con realimentación de estados estática que queda estable alrededor de la trayectoria deseada. Corresponde entonces hacer algún tipo de pruebas con la realimentación y ver bajo que error en las condiciones iniciales se alcanza la convergencia. Pero previo a eso cabe hacer una distinción entre las dos situaciones de vuelo que en principio deseamos controlar.

## 6.2.2. Cambio de variables de estado para el vuelo en círculos

Tal como se dijo en el capítulo de la linealización, el MVE presentado puede ser linealizado alrededor de ambas trayectorias, pero es solo sobre el vuelo a velocidad constante que el sistema que se obtiene es invariante en el tiempo y se puede por lo tanto implementar el control como se dice en los párrafos anteriores. Para el caso de las trayectorias con velocidad angular constante (círculos), el sistema no queda invariante en el tiempo si se considera a la posición expresada en el sistema inercial como parte del vector de estados, por lo que para dar el tratamiento antes descrito es necesario dejar a la posición por fuera del sistema en principio. Esto trae aparejada la consecuencia de que, mediante una realimentación estática, no es posible controlar la posición (aunque si la velocidad) por lo que la misma podría en principio diverger durante la trayectoria (a menos que se haga control integral o que los polos del sistema sean tan rápidos que el error asintótico en los círculos sea despreciable).

A efectos de evitar esto y obtener un vector de estados que considere la posición del sistema, pero que a su vez al linealizarlo alrededor de las trayectorias con velocidad angular constante quede un sistema invariante en el tiempo, procedimos a hacer un cambio de variables para nuestro vector de estados. El cambio de variable en sí es menor: medir la posición del avión en las trayectorias circulares a partir del centro del círculo que se describe, y luego expresar el nuevo vector en la base solidaria al avión (Body). Este cambio de variable intuitivamente pasa el sistema a uno invariante en el tiempo, ya que a lo largo de todo el círculo esta nueva magnitud (que denominaremos  $x_B$ ) se debería mantener constante y por lo tanto su derivada nula. Esto se ve en una de las ecuaciones que ya se presentó en el capítulo de linealización:

$$\dot{x}_B = v_B + w_B \times x_B$$

Con este nuevo vector para medir la posición (sustituyendo la posición expresada en el sistema inercial y manteniendo así un vector de estados de 12 elementos), obtuvimos la nueva linealización del sistema y con dicha linealización aplicamos los métodos descritos anteriormente para controlar el avión alrededor de estas trayectorias. Es importante notar que la diferencia entre una representación del sistema y otra radica solamente en la elección de una base de coordenadas para la posición, pero físicamente las cantidades que están presentes en el vector de estados siguen siendo las mismas. De esta forma no es necesario cambiar ni el concepto de trayectoria (tan solo hay que expresar la trayectoria deseada en el nuevo sistema de coordenadas para compararla contra la trayectoria real), ni la estrategia de realimentación de estados.

### 6.2.3. Modelo del sistema para el análisis de los controladores

Antes de pasar a hacer un primer análisis de los resultados obtenidos con los controladores descritos anteriormente, es necesario establecer claramente las condiciones en las que se harán las pruebas de los mismos. Como se vio en el capítulo de modelado, existe una dicotomía entre lo que es el sistema real a controlar y lo que podría llamarse un modelo más 'preciso' del mismo, y el modelo con el cual se trabajó para la linealización y el diseño de controladores. A modo de ejemplo, para este último modelo se habían despreciado efectos tales como la variación de la densidad del aire con la altura, ya que no permitía tener un sistema SLIT en las condiciones de vuelo deseadas, pero sabemos que en realidad dicha dependencia existe. Pasaremos a explicitar entonces las diferencias entre el modelo utilizado para las simulaciones del sistema de control y el modelo que se linealizó y sobre el cual se hizo el diseño del control.

Como ya se dijo, una diferencia entre ambos modelos es la de tener densidad del aire en la atmósfera variable con la altura o no. Por otro lado también está la consideración de la variabilidad de la masa del avión a medida que se consume combustible en vuelo????, y posiblemente una de las mayores diferencias surja para el modelo en trayectorias circulares, para las cuales se había despreciado la velocidad del viento expresada medida desde el referencial inercial, elemento que se retoma para las simulaciones.

Otra diferencia sustancial entre el modelo utilizado para experimentar y el usado para el diseño del controlador son los efectos de saturación. Dado que en el sistema real las entradas están necesariamente acotadas, y seguramente a la hora de implementar cualquier mecanismo de control sea necesario hacer una limitación explícita de dichas acciones para evitar romper los actuadores (además de que posiblemente ya exista una saturación en la etapa de conversión digital/analógico necesaria para dar órdenes a los actuadores), se implementaron bloques de saturación en las entradas al sistema simulado.

### 6.2.4. Agregado de dinámica para controlar errores grandes en la posición

Por la naturaleza de la trayectoria que queremos seguir (concatenación de segmentos de rectas y círculos), si bien es deseable que la posición sea una variable controlada en todo el trayecto, de poder garantizar que durante los tramos rectos la posición se corrija a su valor deseado -sin perjuicio de que durante los tramos circulares ésta se aleje de dicho valor-, podríamos garantizar que la posición es controlada a lo largo de la trayectoria. El problema aquí es que el cambio de punto de operación entre un círculo y una recta puede ser muy grande, en especial cuando no se realiza control de posición sobre el tramo de círculo, entonces cuando el punto de operación pasa a la recta, nos encontramos con un 'escalón' en la diferencia entre el estado real y el deseado, de un tamaño que puede ser considerable lo que puede llevar a que el sistema no sea controlable (recordemos que nuestro sistema es estable en su linealización, pero su condición de no lineal implica que perturbaciones muy grandes puedan hacer que el sistema diverja). Es por esto que es interesante notar una propiedad del MVE que tenemos: éste no depende explícitamente de la posición en el espacio, por lo que para cualquier valor de posición, si las otras 9 variables de estado se mantienen, el sistema es el mismo (y muy particularmente su linealización<sup>c</sup>).

Esta particularidad del sistema permite hacer el siguiente análisis: si el vector de estados se encuentra

---

<sup>c</sup>Esto se puede ver claramente en la versión linealizada notando que las primeras 3 columnas de la matriz A son nulas

en el punto de equilibrio para todas sus variables excepto la posición, el sistema linealizado se mantiene para este nuevo punto de operación sin importar la posición. Por lo tanto, buscar la convergencia primero en las 9 últimas variables del vector de estado y recién luego tratar de conciliar la posición real con la posición deseada puede funcionar adecuadamente como una estrategia de control si la diferencia de posición no es demasiado grande <sup>d</sup>

Procedimos entonces al agregado de cierta dinámica a la realimentación de estados, de modo de lograr simultáneamente dos objetivos: el primero es tratar de llegar con las últimas 9 variables del vector de estado al equilibrio antes que con la posición, y el segundo es que los errores en la posición aparezcan lentamente (y no como escalones) para evitar que esto cause un alejamiento del sistema de su punto de operación. Para esto una estrategia simple es el agregado de un bloque con un polo en el loop de realimentación de la posición. De esta manera se evita que al pasar de un punto de operación a otro, los errores en la posición (y su realimentación) dominen en el sistema y eviten que las otras variables de estado lleguen al equilibrio.

---

<sup>d</sup>Esto se debe a que una diferencia de posición demasiado grande podría necesitar de acciones de control demasiado grandes que sacaran a las otras 9 variables de la zona lineal alrededor del punto de operación.

# Software implementado

Del software implementado durante el transcurso de este proyecto se pueden identificar tres grandes categorías; el software de simulación, el *software* para el generador de trayectorias y la lógica de control propiamente dicha. La primera de ellas, el *software* de simulación, fue el primer programa que se implementó y cuyo primer objetivo fue tener el MVE funcionando en un ambiente simulado. El generador de trayectorias es el que tiene la misión de, dado los puntos por los cuales se deba pasar (*waypoints*), generar una trayectoria continua con derivada espacial continua que pase suficientemente cerca de los puntos especificados. Finalmente, el controlador, cuya primer capa está dedicada a lograr que el sistema siga una determinada trayectoria  $(x, y, z)$  dada, y la segunda capa es la encargada de interactuar con el generador de trayectorias para regenerar la trayectoria a seguir en caso de apartarse mucho de la original (cuando ya no tiene mucho sentido que la primer capa siga intentando seguir a la trayectoria original).

Es de mencionar en este punto que no todo el software disponible se engloba con estas tres categorías, por ejemplo, el software de linealización y trimado que se discutió en el capítulo ?? podría considerarse como dentro de la lógica de control pues es necesario utilizar dicho software para el cálculo de los controladores. Sin embargo, en realidad es un *software* completamente independiente y no corresponde concretamente a ninguna de las tres categorías. Otro ejemplo de esto podría ser el *software* de adquisición de señales y datos elaborado por el proyecto de fin de carrera anterior, los cuales se encuentra detallados en la referencia [7].

Todos los programas que se implementaron durante el transcurso de este proyecto fueron entregados en dos modalidades, que hemos dado a llamar por:

- ▷ **Paquete de aplicaciones**
- ▷ **Software integrado**

La modalidad de paquete de aplicación apunta hacia un usuario 'informado', es decir, un usuario que tenga conocimientos sólidos sobre las herramientas *matlab* y *simulink*. Se entregaron entonces todos los *scripts*, las funciones *.m* y los MDLs construidos. La documentación de los mismos se encuentra en el manual de usuario entregado. Estos programas se corresponden a la implementación sobre la plataforma *matlab* de todo lo referido a lo que se discutió en los capítulos anteriores, y representan el núcleo en si mismo del proyecto.

Por otro lado, todo este *software* implementado lo presentamos también en la modalidad *software* integrado, donde se encuentran disponibles todas las funcionalidades correspondientes a los programas construidos y disponibles en el paquete de aplicación, pero unificadas en un solo programa con interfaz gráfica que llamaremos de ahora en más UI (del inglés *User Interface*). Es claro que esta otra modalidad apunta hacia un usuario cuyo único objetivo es utilizar las funcionalidades sin necesidad de tener que acceder al código o a la línea de comandos de forma exhaustiva.

A continuación procederemos a describir brevemente el proceso por el cual se generó este *software*.

## 7.1. Software integrado

El objetivo de esta aplicación es brindar la posibilidad de utilizar las herramientas desarrolladas durante el transcurso del proyecto sin necesidad de tener que acceder al código. Se debieron integrar distintos

tipos de programas y datos. Básicamente este *software* debía integrar: las funciones *.m*, varios *scripts* de ejecución, datos a ser levantados desde archivos de texto *.txt*, datos que se almacenaban en el *workspace* de *Matlab* y la interfaz gráfica propiamente dicha, que debía ser diseñada e implementada.

### 7.1.1. Especificaciones

### 7.1.2. Diseño de la integración

Para lograr esta integración el primer paso fue erradicar la necesidad del *workspace* de *Matlab*, dado que se encontro dificultoso acceder a la modificaciones de los datos allí presentes desde la *UI* y las funciones *.m*.

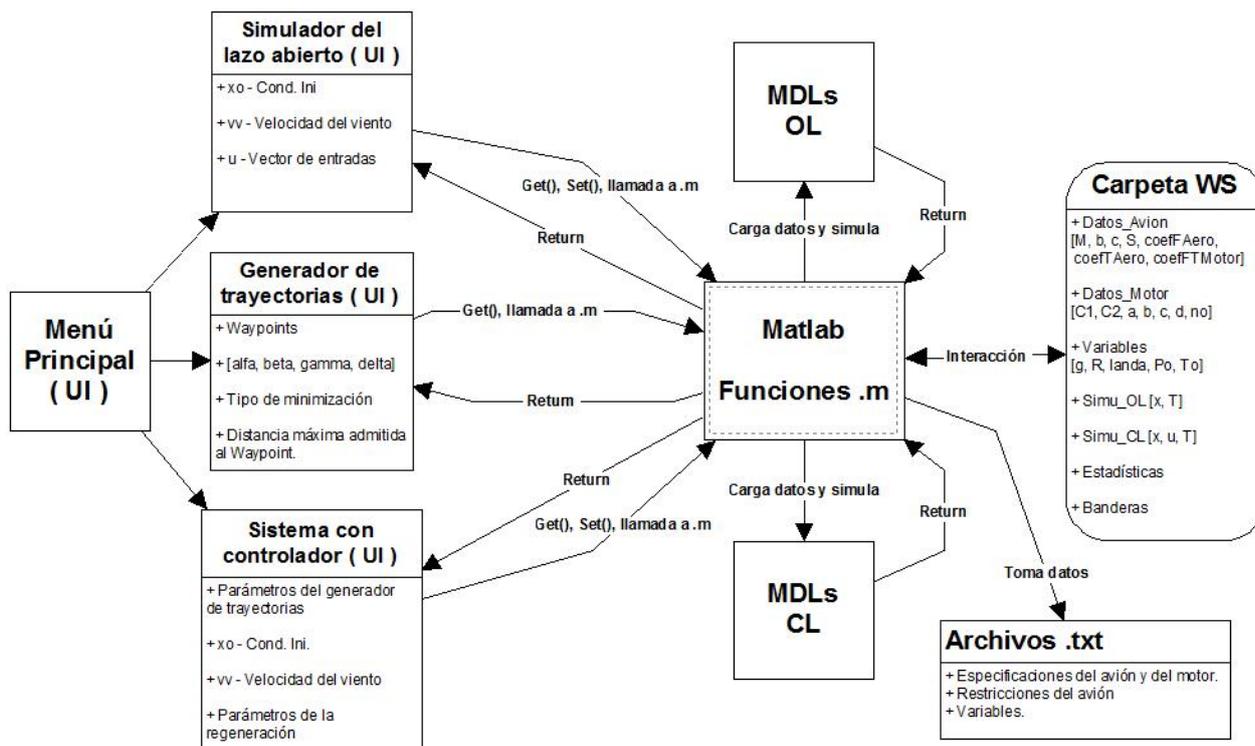


Figura 7.1: Diseño de la integración del *software*.

### 7.1.3. Interfaz gráfica



Figura 7.2: Menú principal del *software* integrado.

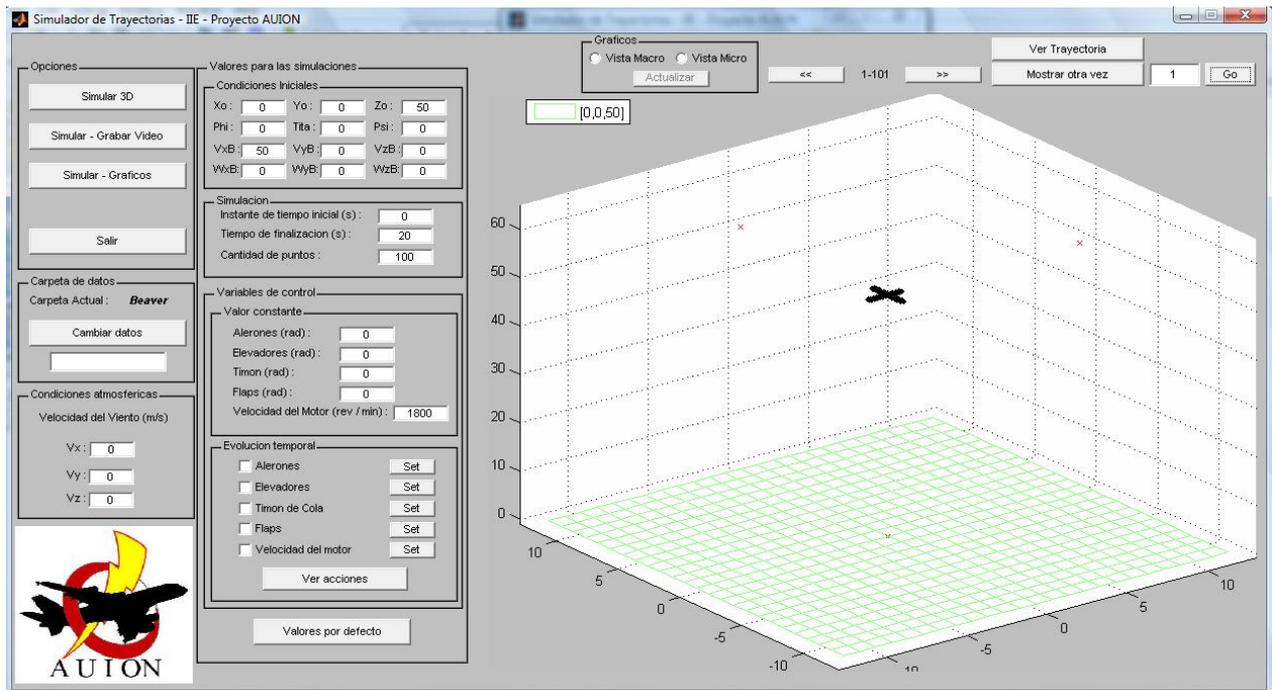


Figura 7.3: Menú para el simulador del lazo abierto del *software* integrado.

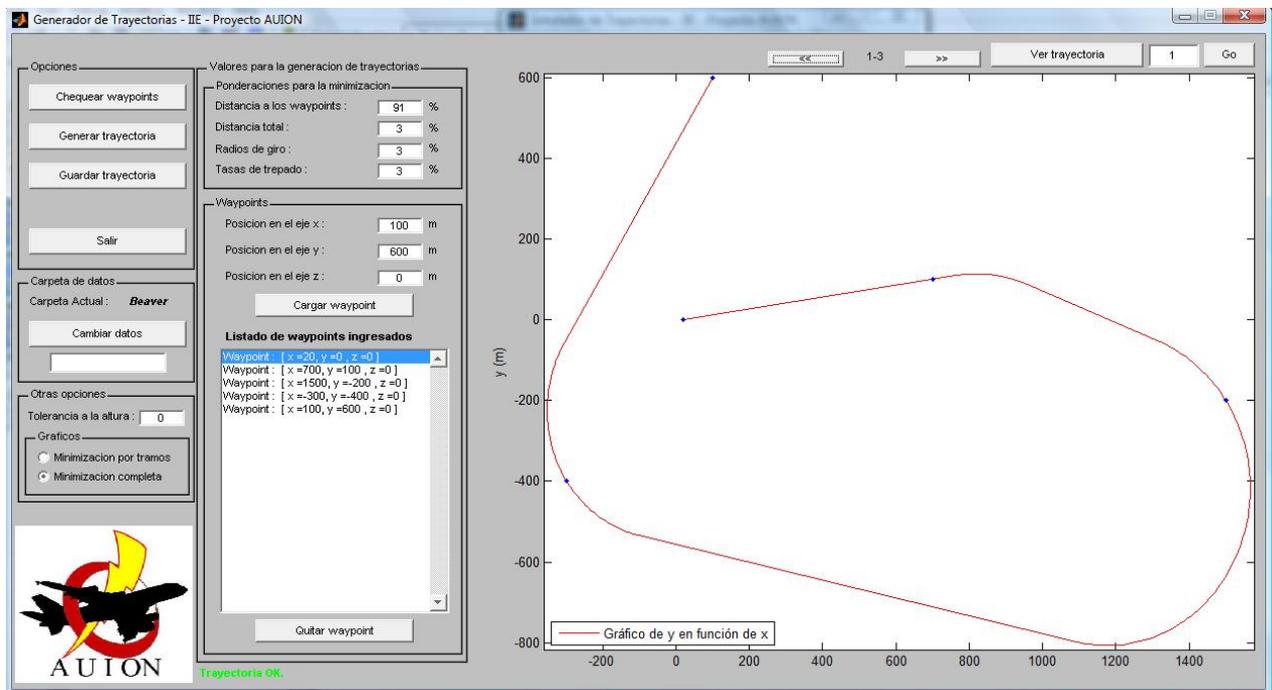


Figura 7.4: Menú para el generador de trayectorias del *software* integrado.

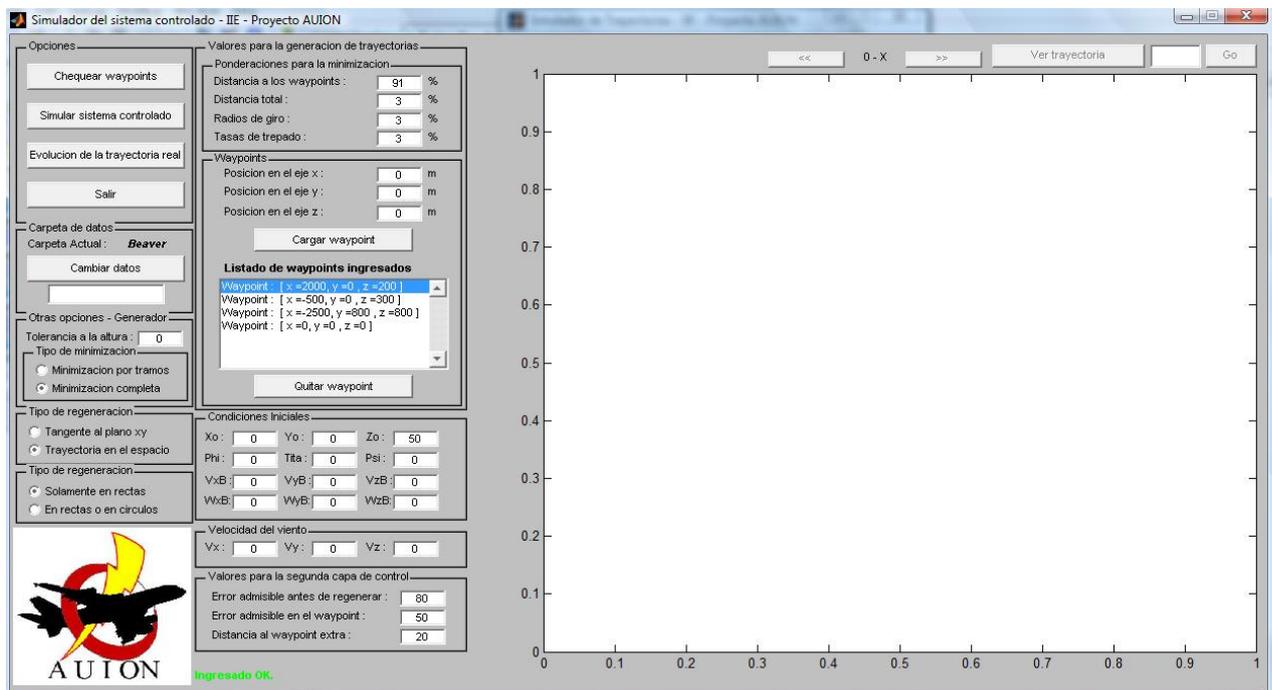


Figura 7.5: Menú para el simulador del lazo cerrado del *software* integrado.

# **Consideraciones generales para la implementación del sistema propuesto**

---

Como ya se dijo, originalmente el alcance de este proyecto incluía la implementación del sistema de control diseñado en el avión de la UdelaR. Dado que por diversos motivos el proyecto redujo su alcance a solamente el diseño del controlador, consideramos necesario dejar algunas pautas escritas sobre lo que puede ser una futura implementación del controlador, haciendo especial énfasis en una futura implementación en el avión de la UdelaR, que era el destinatario inicial de nuestros esfuerzos. La fundamentación para la existencia de este capítulo radica en que, por ser nosotros quienes desarrollamos el sistema, creemos tener un conocimiento del mismo, sus fortalezas y debilidades que ayudaría bastante a la hora de implementarlo y que no deseábamos que se perdiera por la falta de continuidad en el equipo de trabajo. Asimismo, dejar constancia de los pasos que creemos que habría que seguir para alcanzar los objetivos que inicialmente nos habíamos planteado para este proyecto.

## **8.1. El generador de trayectorias**

El generador de trayectorias tal como fue pensado no necesariamente deberá implementarse en el computador de abordo del avión, ya que una posibilidad sencilla sería la de correr el programa en una PC de escritorio estándar y luego pasar la trayectoria generada al computador del avión. De no optarse por esta alternativa, posiblemente sea necesario re-pensar el algoritmo desde el punto de vista de su eficiencia computacional, que probablemente sea el punto más débil del mismo.

## **8.2. Validación del modelo**

Quizás la primera y más importante tarea que queda pendiente para la etapa de implementación es la de validación del modelo. Nuestro modelo se basó en un análisis mecánico de la situación, así como en modelos desarrollados en otros trabajos para las fuerzas aerodinámicas y del motor. En particular estos últimos no recibieron por parte nuestra ninguna verificación más que un análisis cualitativo de las características del modelo.

Más allá de eso, es necesario antes de pasar a la implementación de todo el sistema una adecuación del modelo al avión que se desee controlar. Ha sido una constante a lo largo de este proyecto los intentos por diseñar el sistema de la forma más general posible, de manera que se pudiera portar al uso con otros aviones distintos al 'Beaver'. En primera instancia entonces sería necesario hacer un relevamiento de los parámetros del avión al que se le desea implementar el sistema (envergadura de alas, cuerda aerodinámica media, perfil alar, etc.), las fuerzas aerodinámicas y los coeficientes que las describen en términos de los polinomios que se presentaron en el capítulo de modelado (asumiendo que los polinomios con los mismos términos son aplicables, cosa que en principio estimamos razonable) y la fuerza y momento ejercidos por el motor, también expresado en términos de los coeficientes del polinomio que lo modela. Estos últimos relevamientos necesariamente requerirían de experimentos en un túnel de viento o incluso en vuelo, aunque esta última posibilidad parece más difícil de lograr (aunque esto último podría usarse para hacer un ajuste del

modelo una vez que esté disponible, mediante filtros de Kalman o similares para ajustar la salida esperada por el modelo con la salida real del sistema).

### 8.3. Evaluación de necesidades en aviónica

Otro elemento clave para la implementación del sistema de control es la aviónica, o electrónica de abordo. Este punto está fuertemente relacionado con las secciones siguientes, en particular debido a que el adecuado dimensionamiento de la aviónica depende de como se implemente la observación de estados, los algoritmos de control, y las necesidades de actuadores.

A modo de listado no exhaustivo del dimensionamiento de la aviónica necesario, presentamos aquí una primera versión: capacidad de procesador de abordo (velocidad, memoria, consumo energético), necesidad de conversores A/D y D/A (cantidad, rango dinámico, resolución), sensores de medida inercial (acelerómetros y giróscopos), otros sensores (GPS, pitot), generador de pulsos PWM para comandar los servomotores, baterías para la alimentación del conjunto, transmisores y receptores para la comunicación con tierra, etc.

En el caso del avión de la UdelaR muchas de estos elementos ya están presentes (GPS, acelerómetros, giróscopos, procesador, algunos conversores A/D y D/A, generador PWM) y otros que no. Sin embargo de los elementos presentes sería necesario una re-evaluación o re-dimensionamiento, por ejemplo en el caso de los conversores A/D y D/A ya había quedado claro para el grupo que desarrolló el avión que su resolución y rango dinámico era insuficiente. También es necesario re-evaluar la capacidad de procesamiento presente y necesario para la implementación del controlador propuesto. Muy particularmente también recomendamos el estudio de las necesidades de energía que tiene el sistema actualmente y como satisfacerlas, ya que el avión tal como se encuentra carece de cualquier tipo de alimentación. En nuestros primeros contactos con el mismo notamos que esto sería un tema crítico, pues para suministrar la energía necesaria con baterías de bajo costo la única solución disponible en el mercado consistía en baterías de gel que tienen un peso mayor a 0,5kg, lo que consideramos excesivo para un avión cuya payload es del entorno de los 2kg.

En cuanto a los sensores existen algunas carencias evidentes como lo es el pitot, pero el resto de los sensores -que ya están presentes- deberán ser evaluados en función de la capacidad que se tenga de hacer una correcta observación de estados a partir de los mismos. Algunas consideraciones adicionales sobre esto se hacen en la sección sobre observadores de estados.

El procesador que será necesario dependerá fuertemente de como se haga la implementación algorítmica del conjunto (complejidad computacional), los tiempos que efectivamente tome la ejecución de los distintos programas y los tiempos que efectivamente es razonable que tome su ejecución (por ejemplo, la regeneración de trayectorias no puede tomar 10s en ejecutarse pues volando a  $30m/s$  eso implica que se habrá diseñado una trayectoria para una situación de vuelo que ya dista  $300m$  de la situación real). En general no es posible evaluar las necesidades en términos del procesador hasta que se pase todo el software por un proceso serio de optimización en términos de eficiencia computacional, por lo que hacia esto deberían dirigirse los primeros esfuerzos en el tema.

Finalmente, el sistema de comunicación con tierra debe implementarse completamente (la única comunicación que existe actualmente es unidireccional desde el control remoto hacia el avión y no permite proveer más información que la posición deseada de los servo motores) y para esto primero debería hacerse un análisis de las características deseables para dicha comunicación. A modo de ejemplo, debería determinarse si se desea una comunicación permanente, la distancia de transmisión, el medio para la misma, qué se desea transmitir (estados, trayectorias, etc.), si la comunicación será unidireccional o bidireccional, etc.

### 8.4. Trimado y linealización

El trimado y la linealización del sistema alrededor de un punto de operación deberá poder hacerse abordo del avión si se desea implementar el controlador propuesto, debido a que el control mediante

regeneración de trayectorias exige la posibilidad de 'repensar' los puntos de operación y por lo tanto el control alrededor de los mismos.

Como ya se dijo, el trimado consiste esencialmente en resolver un sistema de ecuaciones no lineales mediante algún método numérico adecuado, pero debe destacarse que en su implementación actual la eficiencia del código no ha sido considerada exhaustivamente. Un elemento clave para todas las optimizaciones y resolución de ecuaciones no lineales es poseer un valor para comenzar a iterar suficientemente cercano a la solución. En el caso del trimado es fácil obtener puntos 'suficientemente cercanos' corriendo el software para un conjunto de casos estándar (p.ej.: una recta horizontal a velocidad de crucero sin viento) y luego asociando estos puntos conocidos como 'semilla' para el proceso de trimar una trayectoria similar.

Una situación absolutamente distinta se sostiene para la linealización, pues se han derivado las ecuaciones analíticamente y luego se evalúan las mismas para cada caso en función de los resultados del software de trimado, lo que es computacionalmente mucho más eficiente que buscar programar el MVE completo y linealizarlo numéricamente entorno al punto de operación deseado. De mantenerse el MVE no sería necesario hacer más que codificar el programa para el nuevo procesador, ya que fue programado teniendo la portabilidad como objetivo <sup>a</sup>.

## 8.5. Observación de estados

A la hora de desarrollar el controlador se asumió que se disponía en todo momento de una medida adecuada del vector de estados, hipótesis que nunca justificamos. Se desprende de esto que es necesario para implementar el sistema propuesto contar con dicha medida o estimar adecuadamente el vector de estados. En la realidad habrá que hacer una mezcla entre las cantidades que se pueden medir y las que habrá que 'observar', esto es, a partir de medidas que se pueden realizar deducir el valor de las magnitudes no mesurables. Entran en juego entonces el conjunto de los sensores y el procesamiento de las señales que los mismos brindan. Algunos sensores que serán muy útiles para esta etapa son los acelerómetros, giróscopos y GPS. Los giróscopos permiten -idealmente- conocer la velocidad angular del avión y eventualmente es posible, a partir de dichas medidas, conocer los ángulos de Euler o la inclinación del avión. De una manera análoga, los acelerómetros permiten conocer la aceleración instantánea del avión y es posible derivar a partir de ellos la velocidad y la posición del avión. En ambos casos para evitar errores integrales al obtener las nuevas magnitudes es necesario un procesamiento de las señales adquiridas.

De manera estándar, al tratar de derivar la posición desde un sistema de medida inercial (IMU) es necesario contar con una referencia adecuada que permita contrarrestar los errores integrales. Algunos pasos en este sentido fueron dados por el grupo que desarrolló el avión (REFERENCIA), al implementar un filtro de Kalman para establecer una medida adecuada de la posición a partir de los acelerómetros y el GPS. Posiblemente sea necesario realizar un procedimiento similar para obtener la inclinación (ángulos de Euler) a partir de los giróscopos.

Hay además que obtener alguna estimación de la velocidad relativa del viento, lo que puede hacerse mediante la inclusión de nuevos sensores (pitot, anemómetro) o mediante observación de estados.

Para todos los casos puede ser interesante implementar también un modelo de los sensores para incluir en el modelo total, de manera de poder hacer simulaciones más precisas de las situaciones de vuelo deseadas.

## 8.6. Entradas y actuadores

Otro elemento que no se consideró a la hora de diseñar el sistema de control es el hecho de que no es posible manejar directamente las 'entradas' del sistema sino que se actúa sobre actuadores (valga la

---

<sup>a</sup>Por 'mantenerse' entendemos que las ecuaciones que describen la situación, en particular las de las fuerzas y momentos asociados a la aerodinámica y al motor, mantengan su forma pero no necesariamente su cuantificación. Por ejemplo, si se pueden describir las nuevas fuerzas en términos de los mismos polinomios usados en este trabajo aunque los valores numéricos de los coeficientes varíen, entonces el software de linealización se puede mantener incambiado.

redundancia) que en su lugar afectan a las distintas entradas. Es necesario entonces considerar la dinámica de estas relaciones y modelarlas de algún modo si se desea hacer simulaciones realmente completas del sistema.

En el caso del avión de la UdelaR, existen 4 entradas que son controladas por servomotores en todos los casos. Para estos sería adecuado hacer dos estudios, el de la respuesta mecánica frente a las excitaciones eléctricas (pulsos PWM) y el de la dinámica del sistema, que determina la velocidad de respuesta de los actuadores sobre las entradas.

Estrictamente hablando, lo anterior alcanza para los actuadores de las superficies de vuelo (aleros, timón y elevadores) pero debe ser tratado con algo más de cuidado para el motor. En el caso del motor, el servomotor asociado controla el pasaje de nafta y no hay -a priori- una relación directa de esto con las RPM del mismo. Sería necesario entonces estudiar y hacer un modelado de la interacción entre el pasaje de nafta y las RPM o implementar quizás un mini-sistema regulador de las RPM que a su vez comande al servomotor.

## 8.7. Controlador lineal

Para la implementación del controlador lineal o de primera capa, existen esencialmente dos tipos de consideraciones para realizar. Las primeras están asociadas con el hecho de que el controlador debe implementarse en un sistema digital, lo que implica tener que discretizar el controlador y para esto es necesario primero determinar la frecuencia de muestreo necesaria. De alguna manera un proceso similar es el que se lleva a cabo para simular el controlador con el software diseñado, lo cual ya es prueba de la posibilidad de implementar el sistema discreto. La gran diferencia radica en que el simulador -tal como funciona actualmente- utiliza mecanismos para resolver las ecuaciones diferenciales que tienen tasas de muestreo variables y no necesariamente realistas.

Surgen entonces las consideraciones del segundo tipo, que son las que están relacionadas con la posibilidad de ejecutar el sistema en tiempo real. La ejecución del sistema en tiempo real es condición *sine qua non* para la implementación del mismo. Necesariamente esto está fuertemente relacionado con la tasa de muestreo del sistema, la complejidad computacional del algoritmo programado y las prestaciones del procesador y electrónica de abordo. Repitiendo lo expresado en la sección dedicada a la aviónica, quizás esta sea la parte más delicada y crítica a la hora de implementar el sistema propuesto.

## 8.8. Regeneración de trayectorias

Para la regeneración de trayectorias rigen criterios similares a los anteriores. Si bien no es necesario que la regeneración de trayectorias se haga en tiempos comparables con el ciclo de control de primera capa, si es necesario -por los motivos que se expresaron en la sección de aviónica- que sea suficientemente rápida. Dado que el proceso de generar (y regenerar) trayectorias consiste esencialmente en optimizar una función, una sugerencia que puede ayudar a agilizar el proceso es la de utilizar una 'semilla' adecuada, al igual que lo que se sugirió para el trimado. En el caso de la regeneración de trayectorias, una 'semilla' adecuada puede ser la trayectoria que se estaba siguiendo en el momento que se decide regenerar, ya que dicha trayectoria tiene la característica de pasar por todos los waypoints, y probablemente no diste mucho de la nueva trayectoria buscada.

Para concluir este capítulo nos gustaría reiterar que el énfasis de este proyecto estuvo puesto en la portabilidad, considerando siempre la mejor manera de dejar establecido el sistema de control de modo de que pudiera implementarse en un avión genérico. Esto implica entre otras cosas, que se ha intentado modularizar el sistema e independizarlo de las características del avión en cuestión, así como utilizar algoritmos cuya programación y ejecución fuera posible en un microprocesador genérico de manera sencilla. Un ejemplo de esto es que se usaron solo funciones de Matlab que implementan algoritmos conocidos y no propietarios que

fueran posibles de reproducir para cualquier persona que tuviera nociones básicas de métodos numéricos.

Si bien no nos ilusionamos con haber logrado una portabilidad e independencia absoluta del avión en el que se desee implementar el sistema, esperamos sí haber avanzado en estas líneas lo suficiente como para que en un futuro, si alguien desea tomar el sistema propuesto para su implementación, no sea necesario revisar y rehacer grandes porciones del mismo.

---

## Apéndice A

# Cálculos realizados

---

Apéndice destinado al detalle sobre los cálculos realizados para la obtención de algunos de los resultados que se consideran de importancia para este trabajo.

### A.1. Cálculos relativos a las linealizaciones

En esta sección se enumeran los cálculos relativos a la obtención de las linealizaciones jacobianas del MVE. Dado que dicho MVE es de tamaño considerable y que además las ecuaciones involucradas son bastante complicadas y no lineales, se debió realizar estas linealizaciones de manera que incluyeran la menor cantidad de cuentas posibles, o por lo menos, pudiera hacerse parte por parte.

En las líneas de lo anterior debimos buscar la manera más convenientes de obtener las derivadas necesarias de  $\dot{x}$  de la manera más conveniente. Observando un poco la forma del MVE podemos ver que si bien se cumple  $\dot{x} = f(x, u)$  existe una forma más conveniente de expresar esta dependencia como:

$$\dot{x} = f(x, u) = f^*(x, F, T) \quad \text{con} \quad \begin{cases} F = f_1(x, u) \\ T = f_2(x, u) \end{cases} \quad (\text{A.1})$$

Donde hemos hecho incapié en que la dependencia en  $u$  entra a la ecuación por las nuevas variables  $F$  y  $T$  que claramente representan a los vectores de la fuerza y el torque neto sobre el sistema, y también depende del vector de estados  $x$ . Como se desprende de la ecuación 5.3, para calcular las linealizaciones deberemos calcular los términos:

$$\left. \frac{\partial f_i(x, u)}{\partial x_j} \right|_{(x_0, u_0)} \quad \left. \frac{\partial f_i(x, u)}{\partial u_k} \right|_{(x_0, u_0)} \quad (\text{A.2})$$

Donde  $f_i$  es la  $i$ -ésima componente del vector  $f$ ,  $x_j$  es el  $j$ -ésimo elemento del vector de estado  $x$  y  $u_k$  es el  $k$ -ésimo elemento del vector de entradas  $u$ . Para después poder evaluarlos en las trayectorias de operación  $x_0$  y  $u_0$ . El cálculo de estas derivadas lo haremos de ahora en adelante utilizando las siguientes igualdades, las ecuaciones A.3 y A.4, la cual tiene su origen en la regla de la cadena.

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_i^*}{\partial x_j} + \left( \frac{\partial f_i^*}{\partial F_i} \right) \left( \frac{\partial f_{1i}}{\partial x_j} \right) + \left( \frac{\partial f_i^*}{\partial T_i} \right) \left( \frac{\partial f_{2i}}{\partial x_j} \right) \quad (\text{A.3})$$

$$\frac{\partial f_i}{\partial u_k} = \left( \frac{\partial f_i^*}{\partial F_i} \right) \left( \frac{\partial f_{1i}}{\partial u_k} \right) + \left( \frac{\partial f_i^*}{\partial T_i} \right) \left( \frac{\partial f_{2i}}{\partial u_k} \right) \quad (\text{A.4})$$

Derivadas que se deben hacer para todas las  $f_i$  y para cada  $x_j$  y  $u_k$ . Se debe notar en este punto que estas derivadas definen dos matrices, una por  $\left\{ \frac{\partial f_i(x, u)}{\partial x_j} \right\}$  con  $i, j = 1$  a  $12$  y la otra por  $\left\{ \frac{\partial f_i(x, u)}{\partial u_k} \right\}$  con  $i = 1$  a  $12$  y  $k = 1$  a  $5$ . Estas matrices son las matrices  $A$  y  $B$  de la ecuación 5.4.



▷ **Matriz**  $A_{22} \longrightarrow A_{22} \in \mathcal{M}^{3 \times 3}$ :

$$A_{22} = \begin{pmatrix} \cos \varphi \tan \theta w_{By} - \sin \varphi \tan \theta w_{Bz} & (\sin \varphi / \cos^2 \theta) w_{By} + (\cos \varphi / \cos^2 \theta) w_{Bz} & 0 \\ -\sin \varphi w_{By} - \cos \varphi w_{Bz} & 0 & 0 \\ (\cos \varphi / \cos \theta) w_{By} - (\sin \varphi / \cos \theta) w_{Bz} & (\sin \varphi \tan \theta / \cos \theta) w_{By} + (\cos \varphi \tan \theta / \cos \theta) w_{Bz} & 0 \end{pmatrix} \quad (\text{A.12})$$

▷ **Matriz**  $A_{24} \longrightarrow A_{24} \in \mathcal{M}^{3 \times 3}$ :

$$A_{24} = M = \begin{pmatrix} 1 & \sin \varphi \tan \theta & \cos \varphi \tan \theta \\ 0 & \cos \varphi & -\sin \varphi \\ 0 & \sin \varphi / \cos \theta & \cos \varphi / \cos \theta \end{pmatrix} \quad (\text{A.13})$$

### A.1.2. Linealización en torno a una trayectoria de giro a velocidad angular constante

Como se explicó en el capítulo de linealización, en la sección relativa a la linealización en torno a un trayectoria de gir a velocidad angular constante, se debió realizar un cambio de variables para lograr que el sistema fuera SLIT en torno a dicha trayectoria, y también para que sea mejor computable desde el punto de vista compuncional. El cambio de variable fue el que se describe en la ecuación A.14 que se muestra a continuación.

$$\begin{pmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{pmatrix} = \begin{pmatrix} v_{Bx} \\ v_{By} \\ v_{Bz} \end{pmatrix} - \begin{pmatrix} w_{Bx} \\ w_{By} \\ w_{Bz} \end{pmatrix} \times \begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} \quad (\text{A.14})$$

Al realizar este cambio de varianble, la matriz  $A$  que se obtiene de la linealización es levemente distinta a la de los casos de las dos subsecciones anteriores, y la misma tiene la forma que se muestra en la ecuación A.15.

$$A = \begin{pmatrix} A_{11} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & Id & A_{14} \\ \hline \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & A_{22} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & A_{24} \\ \hline \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & A_{32} & A_{33} & A_{34} \\ \hline \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} & A_{43} & A_{44} \end{pmatrix} \quad (\text{A.15})$$

Donde las submatrices  $A_{22}$ ,  $A_{24}$ ,  $A_{32}$ ,  $A_{33}$ ,  $A_{34}$ ,  $A_{42}$  y  $A_{44}$  mantienen la forma de las ecuaciones A.12, A.13, ??, ??, ??, ?? y ?? respectivamente. La matriz  $B$  permanece incambiada frente a dicho cambio de variable. Falta entonces definir la matrices  $A_{11}$  y  $A_{14}$ , las cuales son de la siguiente forma:

$$A_{11} = \begin{pmatrix} 0 & w_{Bz} & -w_{By} \\ -w_{Bz} & 0 & w_{Bx} \\ w_{By} & -w_{Bz} & 0 \end{pmatrix} \quad (\text{A.16})$$

$$A_{14} = \begin{pmatrix} 0 & -x_{Bz} & x_{By} \\ x_{Bz} & 0 & -x_{Bx} \\ -x_{By} & x_{Bz} & 0 \end{pmatrix} \quad (\text{A.17})$$

# Definición de la simbología utilizada

## Parámetros y variables - Alfabéto

$\alpha$	Ángulo de ataque.
$\alpha_S$	Valor del ángulo de ataque donde se da el fenómeno de <i>stall</i> .
$\beta$	Ángulo de <i>sideslip</i> .
$\delta$	Reservado para el ángulo de desplazamiento de las partes móviles de las distintas superficies de vuelo.
$\delta'_a$	Ángulo de movimiento de los alerones.
$\delta_a$	Variable que representa el movimiento de los alerones en el MVE. $\delta_a = 2\delta'_a$ .
$\delta_e$	Ángulo de movimiento de los elevadores.
$\delta_f$	Ángulo de movimiento de los <i>flaps</i> .
$\delta_r$	Ángulo de movimiento del timón de cola.
$\theta$	Tercera rotación para definir los ángulos de <i>Euler</i> .
$\kappa$	Parámetro importante en la determinación de las fuerzas y momentos del motor.
$\rho$	Densidad del aire a una altura genérica distinta de cero.
$\rho_0$	Densidad del aire a nivel del mar.
$\rho_{NS}$	Densidad de la nafta super a 15°C.
$\Upsilon$	Variable utilizada para definir la inversa de la matriz $I_{OB}$ . $\Upsilon =  I_{OB} /I_{yy}$ .
$\varphi$	Segunda rotación para definir los ángulos de <i>Euler</i> .
$\psi$	Primera rotación para definir los ángulos de <i>Euler</i> .

## Parámetros y variables - Abecedario

$a$	Parámetro de la variable $\kappa$ .
$b$	Según el contexto 1 - Largo del ala. <i>Wing span</i> . 2 - Parámetro de la variable $\kappa$ .
$C_1, C_2$	Parámetro de la variable $\kappa$ .
$c$	Según el contexto 1 - Cuerda aerodinámica. 2 - Parámetro de la variable $\kappa$ .
$\bar{c}$	Cuerda aerodinámica media. MAC.
$D$	Módulo del vector fuerza de <i>drag</i> sobre el avión.
$d$	Parámetro de la variable $\kappa$ .
$F$	Función que representa la fuerza neta aplicada sobre el sistema.
$f$	Función que relaciona la derivada temporal del vector de estados con el vector de estado y las entradas al sistema. $\dot{x} = f(x, u)$ .
$f_1$	Función utilizada en la linealización que vincula las fuerzas $F$ con el vector de estados y las entradas. $F = f_1(x, u)$

$f_2$	Función utilizada en la linealización que vincula los momentos $T$ con el vector de estados y las entradas. $T = f_2(x, u)$
$g$	Aceleración a nivel del mar debido a la gravedad terrestre.
$h$	Vector de entradas que no son controlables, también se lo puede encontrar como $\vec{h}$ o $\vec{h}(t)$ .
$L$	Módulo del vector fuerza de <i>lift</i> sobre el avión.
$m$	Masa total del avión.
$m_c$	Masa total del combustible del avión.
$n$	Velocidad de giro de la hélice del avión medida en <i>rpm</i> .
$n_0$	Parámetro de la variable $\kappa$ .
$\overline{M}_a$	Masa molar del aire.
$O$	Centro de gravedad o centro de masa del avión.
$Ox_B$	Eje longitudinal del avión.
$Ox_Bz_B$	Plano definido por el eje longitudinal del avión y el eje vertical del avión.
$Ox_By_B$	Plano definido por el eje longitudinal del avión y el eje lateral del avión.
$Oy_B$	Eje lateral del avión.
$Oy_Bz_B$	Plano definido por el eje lateral del avión y el eje vertical del avión.
$Oz_B$	Eje vertical del avión.
$P_0$	Presión a nivel del mar.
$p$	Vector de parámetros numéricos importantes para el modelado, también se lo puede encontrar como $\vec{p}$ .
$q$	Usado como parámetro o variable genérica.
$\overline{R}$	Constante universal de los gases. $\overline{R} = 8,3145 kJ/kMolK$
$R_a$	Constante de gas para el aire.
$S$	Superficie alar.
$T$	Variable que representa el momento neto aplicada sobre el sistema.
$T_0$	Temperatura a nivel del mar.
$u$	Vector de entradas que podemos controlar desde la computadora de a bordo, también se lo puede encontrar como $\vec{u}$ o $\vec{u}(t)$ .
$v_{Bx}, v_{By}, v_{Bz}$	Componentes en la dirección $\hat{i}_B, \hat{j}_B$ y $\hat{k}_B$ respectivamente de la velocidad lineal del avión.
$v_{rBx}, v_{rBy}, v_{rBz}$	Componente en la dirección $\hat{i}_B, \hat{j}_B$ y $\hat{k}_B$ respectivamente de la velocidad relativa al viento del avión.
$v_{rBxz}$	Módulo de la proyección del vector $\vec{V}_{rB}$ sobre el plano $Ox_Bz_B$ .
$v_{vBx}, v_{vBy}, v_{vBz}$	Componente en la dirección $\hat{i}_B, \hat{j}_B$ y $\hat{k}_B$ respectivamente de la velocidad del viento.
$w_{Bx}, w_{By}, w_{Bz}$	Componente en la dirección $\hat{i}_B, \hat{j}_B$ y $\hat{k}_B$ respectivamente de la velocidad angular del avión.
$x$	Según el contexto: 1 - Coordenada $x$ utilizada para la posición absoluta del avión desde el SI. 2 - Vector de estados del MVE, también se lo puede encontrar como $\vec{x}$ .
$y$	Coordenada $y$ utilizada para la posición absoluta del avión desde el SI.
$z$	Coordenada $z$ utilizada para la posición absoluta del avión desde el SI.

---

### Coeficientes adimensionados

$C_{al}, C_{am}, C_{an}$	$C_{aj} = \frac{T_{ai}}{\rho S  \vec{v}_r ^2 / 2}$ con $i = x, y, z$ y $j = l, m, n$ respectivamente. Coeficientes para los momentos aerodinámicos $T_{ax}, T_{ay}$ y $T_{az}$ respectivamente, relaciones funcionales con las variables $x, u, h$ y $p$
$C_{ax}, C_{ay}, C_{az}$	$C_{ai} = \frac{F_{ai}}{\rho S  \vec{v}_r ^2 / 2}$ con $i = x, y, z$ . Coeficientes para las fuerzas aerodinámicas $F_{ax}, F_{ay}$ y $F_{az}$ respectivamente, relaciones funcionales con las variables $x, u, h$ y $p$ .
$C_D$	Coeficiente de <i>drag</i> .
$C_L$	Coeficiente de <i>lift</i> .
$C_{l\alpha\delta_a}$	Coeficiente para la variable $\alpha\delta_a$ del coeficiente del momento aerodinámico $C_{al}$ .
$C_{l\kappa\alpha^2}$	Coeficiente para la variable $\kappa\alpha^2$ del coeficiente del momento del motor $C_{pl}$ .
$C_{l\beta}, C_{n\beta}$	Coeficientes para la variable $\beta$ de los coeficientes de los momentos aerodinámicos $C_{al}$ y $C_{an}$ respectivamente.
$C_{l\delta_a}, C_{n\delta_a}$	Coeficientes para la variable $\delta_a$ de los coeficientes de los momentos aerodinámicos $C_{al}$ y $C_{an}$ respectivamente.
$C_{l\delta_r}, C_{n\delta_r}$	Coeficientes para la variable $\delta_r$ de los coeficientes de los momentos aerodinámicos $C_{al}$ y $C_{an}$ respectivamente.
$C_{l_0}, C_{m_0}, C_{n_0}$	Términos constantes de $C_{al}, C_{am}$ y $C_{an}$ respectivamente.
$C_{lw_{Bx}}, C_{nw_{Bx}}$	Coeficientes para la variable $\frac{bw_{Bx}}{2\ \vec{v}_r\ }$ de los coeficientes de los momentos aerodinámicos $C_{al}$ y $C_{an}$ respectivamente.
$C_{lw_{Bz}}, C_{mw_{Bz}}, C_{nw_{Bz}}$	Coeficientes para la variable $\frac{bw_{Bz}}{2\ \vec{v}_r\ }$ de los coeficientes de los momentos aerodinámicos $C_{al}, C_{am}$ y $C_{an}$ respectivamente.
$C_{m\alpha}$	Coeficiente para la variable $\alpha$ del coeficiente del momento aerodinámico $C_{am}$ .
$C_{m\alpha^2}$	Coeficiente para la variable $\alpha^2$ del coeficiente del momento aerodinámico $C_{am}$ .
$C_{m\beta^2}$	Coeficiente para la variable $\beta^2$ del coeficiente del momento aerodinámico $C_{am}$ .
$C_{m\delta_e}$	Coeficiente para la variable $\delta_e$ del coeficiente del momento aerodinámico $C_{am}$ .
$C_{m\delta_f}$	Coeficiente para la variable $\delta_f$ del coeficiente del momento aerodinámico $C_{am}$ .
$C_{m\kappa}$	Coeficiente para la variable $\kappa$ del coeficiente del momento del motor $C_{pm}$ .
$C_{mw_{By}}, C_{nw_{By}}$	Coeficientes para la variable $\frac{cw_{By}}{\ \vec{v}_r\ }$ de los coeficientes de los momentos aerodinámicos $C_{am}$ y $C_{an}$ respectivamente.
$C_{n\beta^3}$	Coeficiente para la variable $\beta^3$ del coeficiente del momento aerodinámico $C_{an}$ .
$C_{n\kappa^3}$	Coeficiente para la variable $\kappa^3$ del coeficiente del momento del motor $C_{pn}$ .
$C_{pl}, C_{pm}, C_{pn}$	$C_{pj} = \frac{T_{pi}}{\rho S  \vec{v}_r ^2 / 2}$ con $i = x, y, z$ y $j = l, m, n$ respectivamente. Coeficientes para los momentos generados por el motor $T_{px}, T_{py}$ y $T_z$ respectivamente, relaciones funcionales con las variables $x, u, h$ y $p$ .
$C_{px}, C_{pz}$	$C_{pi} = \frac{F_{pi}}{\rho S  \vec{v}_r ^2 / 2}$ con $i = x, z$ . Coeficientes para las fuerzas debidas al motor $F_{px}$ y $F_{pz}$ respectivamente, relaciones funcionales con las variables $x, u, h$ y $p$ .
$C_{x\alpha}, C_{z\alpha}$	Coeficientes para la variable $\alpha$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{az}$ respectivamente.
$C_{x\alpha\delta_f}, C_{z\alpha\delta_f}$	Coeficientes para la variable $\alpha\delta_f$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{az}$ respectivamente.
$C_{x\alpha^2}$	Coeficiente para la variable $\alpha^2$ del coeficiente de la fuerza aerodinámica $C_{ax}$ .
$C_{x\alpha^3}, C_{z\alpha^3}$	Coeficientes para la variable $\alpha^3$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{az}$ respectivamente.
$C_{x\alpha\kappa^2}$	Coeficiente para la variable $\alpha\kappa^2$ del coeficiente de la fuerza del motor $C_{px}$ .
$C_{x\delta_f}, C_{z\delta_f}$	Coeficientes para la variable $\delta_f$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{az}$ respectivamente.
$C_{x\delta_r}, C_{y\delta_r}$	Coeficientes para la variable $\delta_r$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{ay}$ respectivamente.
$C_{x\kappa}, C_{z\kappa}$	Coeficientes para la variable $\kappa$ de los coeficientes de las fuerzas del motor $C_{px}$ y $C_{pz}$ respectivamente.
$C_{xw_{By}}, C_{zw_{By}}$	Coeficientes para la variable $\frac{cw_{By}}{\ \vec{v}_r\ }$ de los coeficientes de las fuerzas aerodinámicas $C_{ax}$ y $C_{az}$ respectivamente.

$C_{x_0}, C_{y_0}, C_{z_0}$	Términos constantes de $C_{ax}, C_{ay}$ y $C_{az}$ respectivamente.
$C_{y\alpha\delta_r}$	Coefficiente para la variable $\alpha\delta_r$ del coeficiente de la fuerza aerodinámica $C_{ay}$ .
$C_{y\beta}$	Coefficiente para la variable $\beta$ del coeficiente de la fuerza aerodinámica $C_{ay}$ .
$C_{y\delta_a}$	Coefficiente para la variable $\delta_a$ del coeficiente de la fuerza aerodinámica $C_{ay}$ .
$C_{yw_{Bx}}$	Coefficiente para la variable $\frac{bw_{By}}{2\ v_r\ }$ del coeficiente de la fuerza aerodinámica $C_{ay}$ .
$C_{yw_{Bz}}$	Coefficiente para la variable $\frac{bw_{Bz}}{2\ v_r\ }$ del coeficiente de la fuerza aerodinámica $C_{ay}$ .
$C_{z\delta_e}$	Coefficiente para la variable $\delta_e$ del coeficiente de la fuerza aerodinámica $C_{az}$ .
$C_{z\delta_e\beta^2}$	Coefficiente para la variable $\delta_e\beta^2$ del coeficiente de la fuerza aerodinámica $C_{az}$ .

### Unidades utilizadas

$^{\circ}C$	Grados Celsius. Escala centígrada para la medida de temperatura cuyo cero está ubicado a la temperatura en Kelvin a la cual el agua entra en cambio de fase líquido - sólido, a presión $P_0$ .
$g$	Gramo. Medida de cantidad masa.
$Hz$	Hertz. Medida de frecuencia.
$K$	Kelvin. Escala centígrada para la medida de temperatura cuyo cero es el cero absoluto.
$l$	Litro. Medida de volumen.
$m$	Metro. Medida de longitud.
$m^2$	Metro cuadrado. Medida de área.
$m^3$	Metro cúbico. Medida de volumen.
$N$	Newton. Medida de fuerza.
$s$	Segundo. Medida de tiempo.
$T$	Período. Medida de tiempo.
$W$	Watt. Medida de potencia.

### Versores

$\hat{i}_B, \hat{j}_B, \hat{k}_B$	Versores con punto de aplicación en $O$ según el eje longitudinal, lateral y vertical del avión respectivamente. Base del sistema de referencia SB.
$\hat{i}_I, \hat{j}_I, \hat{k}_I$	Versores con punto de aplicación en el origen que define la dirección de la coordenada $x, y$ y $z$ . Base del sistema de referencia SI.
$\hat{i}_R, \hat{j}_R, \hat{k}_R$	Versores con punto de aplicación en $O$ . Base del sistema de referencia SR que se define a partir del vector $v_{rB}$ .
$\hat{i}_1$	Versor que se obtiene después de efectuar la primera rotación de $-\beta$ correspondiente al cambio de base desde el SR al SB.
$\hat{j}_1$	Según el contexto: 1 - Versor que se obtiene después de efectuar la primera rotación de $\psi$ correspondiente a los ángulos de <i>Euler</i> . 2 - Versor que se obtiene después de efectuar la primera rotación de $-\beta$ correspondiente al cambio de base desde el SR al SB.
$\hat{k}_2$	Versor que se obtiene después de efectuar la segunda rotación de $\theta$ correspondiente a los ángulos de <i>Euler</i> .

### Vectores

$\vec{D}$	Vector fuerza de <i>drag</i> sobre el avión.
$\vec{F}_a$	Fuerza aerodinámica.
$F_{Neta}$	Fuerza neta ejercida sobre el avión.
$\vec{F}_p$	Fuerza que ejerce el motor o la hélice.
$\vec{g}$	Vector aceleración a nivel del mar debido a la gravedad terrestre.
$\vec{h}$	Vector de entradas que no son controlables, también se lo puede encontrar como $\vec{h}(t)$ o $h$ .
$\vec{L}$	Vector fuerza de <i>lift</i> sobre el avión.
$\vec{L}_O$	Momento angular del avión respecto al punto $O$ .
$\vec{P}$	Vector peso del avión.
$\vec{p}$	Vector de parámetros numéricos importantes para el modelado, también se lo puede encontrar como $p$ .
$\vec{q}$	Utilizado como vector genérico.
$\vec{r}$	Vector posición absoluta del avión.
$\vec{T}_a$	Momento debido a las fuerzas aerodinámicas.
$T_{Neto}$	Momento neto ejercido sobre el avión.
$\vec{T}_p$	Momento que ejerce el motor o la hélice.
$\vec{u}$	Vector de entradas que podemos controlar desde la computadora de a bordo, también se lo puede encontrar como $\vec{u}(t)$ o $u$ .
$\vec{v}$	Vector velocidad lineal del avión.
$\vec{v}_B$	Velocidad lineal del avión expresada en la base del SB.
$\vec{v}_I$	Velocidad lineal del avión expresada en la base del SI.
$\vec{v}_r$	Vector velocidad relativa al viento del avión. $\vec{v}_r = \vec{v} - \vec{v}_v$ .
$v_{rB}$	Velocidad relativa al viento del avión expresada en la base del SB.
$\vec{v}_v$	Vector velocidad del viento.
$v_{vB}$	Velocidad del viento expresada en la base del SB.
$\vec{w}$	Vector velocidad angular del avión.
$\vec{w}_B$	Velocidad angular del avión expresada en la base del SB.
$\vec{x}$	Vector de estados del MVE, también se lo puede encontrar como $x$ según el contexto.

---

### Matrices

$A$	Hace referencia a las matrices de transición de estados de un sistema lineal. Matriz de transición de estado que se obtiene del proceso de linealización jacobiana. $A \in \mathcal{M}^{12 \times 12}$ . $\dot{x} = Ax + Bu$ .
$A_{11}, A_{12}, A_{13}, A_{14}$	Submatrices $\in \mathcal{M}^{3 \times 3}$ de la matriz $A$ , correspondientes a la primer hilera de la matriz $A \in \mathcal{M}^{12 \times 12}$ en un agrupamiento de submatrices $3 \times 3$ .
$A_{22}, A_{24}$	Submatrices $\in \mathcal{M}^{3 \times 3}$ de la matriz $A$ , correspondientes a la segunda hilera de la matriz $A \in \mathcal{M}^{12 \times 12}$ en un agrupamiento de submatrices $3 \times 3$ .
$A_{32}, A_{33}, A_{34}$	Submatrices $\in \mathcal{M}^{3 \times 3}$ de la matriz $A$ , correspondientes a la tercer hilera de la matriz $A \in \mathcal{M}^{12 \times 12}$ en un agrupamiento de submatrices $3 \times 3$ .
$A_{33}, A_{34}$	Submatrices $\in \mathcal{M}^{3 \times 3}$ de la matriz $A$ , correspondientes a la cuarta hilera de la matriz $A \in \mathcal{M}^{12 \times 12}$ en un agrupamiento de submatrices $3 \times 3$ .
$B$	Hace referencia a las matrices de transición de entradas de un sistema lineal. Matriz de transición de entradas que se obtiene del proceso de linealización jacobiana. $B \in \mathcal{M}^{12 \times 5}$ . $\dot{x} = Ax + Bu$ .
$B_{31}, B_{41}$	Submatrices $\in \mathcal{M}^{3 \times 5}$ de la matriz $B$ , correspondientes a la tercer y cuarta submatriz de la matriz $B \in \mathcal{M}^{12 \times 5}$ en un agrupamiento de submatrices $3 \times 5$ .

$H_1^2$	Matriz que efectua la segunda rotación de <i>Euler</i> de ángulo $\theta$ .
$H_1^I$	Matriz que deshace la primera rotación de <i>Euler</i> de ángulo $\psi$ .
$H_2^1$	Matriz que deshace la segunda rotación de <i>Euler</i> de ángulo $\theta$ .
$H_2^B$	Matriz que efectua la tercera rotación de <i>Euler</i> de ángulo $\varphi$ .
$H_B^2$	Matriz que deshace la tercera rotación de <i>Euler</i> de ángulo $\varphi$ .
$H_B^I$	Matriz que efectua el cambio de base desde la base del SB a la base del SI.
$H_B^R$	Matriz que efectua el cambio de base desde la base del SB a la base del SR.
$H_I^1$	Matriz que efectua la primera rotación de <i>Euler</i> de ángulo $\psi$ .
$H_I^B$	Matriz que efectua el cambio de base desde la base del SI a la base del SB.
$H_R^B$	Matriz que efectua el cambio de base desde la base del SR a la base del SB.
$I_{OB}$	Matriz de inercia del avión respecto al punto $O$ expresada en la base del SB.
$I_{OB}^{-1}$	Inversa de la matriz de inercia del avión respecto al punto $O$ expresada en la base del SB.
$L$	Matriz que efectua el cambio de coordenadas desde el vector $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ hacia el vector $(w_{Bx}, w_{By}, w_{Bz})$ .
$M$	Matriz que efectua el cambio de coordenadas desde el vector $(w_{Bx}, w_{By}, w_{Bz})$ hacia el vector $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ .

---

## Apéndice C

# Acrónimos utilizados

---

- ▷ CL - *Close Loop*.
- ▷ EDO - Ecuación Diferencial Ordinaria.
- ▷ IMU - Inercial Measurement Unit - Referencia a los elementos de medición a bordo del avión.
- ▷ DHC - *De Havilland of Canada Ltd.*. Empresa que manufacturó el avión *BEAVER* que fue utilizado como modelo de avión durante este proyecto.
- ▷ MAC - *Mean Aerodynamic Chord*. Referencia a la cuerda aerodinámica media.
- ▷ MVE - Modelo en Variables de Estado.
- ▷ LQR - *Linear Quadratic Regulator*.
- ▷ OL - *Open Loop*.
- ▷ RPM - Revoluciones por minuto.
- ▷ SB - Sistema Body - Sistema de referencia con base  $(\hat{i}_B, \hat{j}_B, \hat{k}_B)$ .
- ▷ SI - Sistema Inercial - Sistema de referencia con base  $(\hat{i}_I, \hat{j}_I, \hat{k}_I)$ .
- ▷ SLIT - Sistema Lineal Invariante en el Tiempo.
- ▷ SR - Sistema Relativo al viento - Sistema de referencia definido por el vector  $v_{rB}$  cuya base es  $(\hat{i}_R, \hat{j}_R, \hat{k}_R)$ .
- ▷ UAV - Unmanned Aerial Vehicle.
- ▷ UI - *User Interface*.

# Índice de figuras

1.1.	Distintos elementos que componen la configuración estandar en un avión. [9] <i>Imagen modificada.</i> . . .	4
1.2.	Descriptivo de los ejes longitudinal, lateral y vertical. Se puede notar el sentido definido positivo para las rotaciones de <i>roll</i> , <i>pitch</i> y <i>yaw</i> . [9] <i>Imagen modificada.</i> . . . . .	5
1.3.	Descriptivo de las distintas superficies de vuelo. [9] <i>Imagen modificada.</i> . . . . .	5
1.4.	Forma de medir los ángulos para la deflexión de las superficies de vuelo. [2] <i>Imagen modificada.</i> . . .	6
1.5.	Descriptivo de los ángulos utilizados para medir los desplazamientos de los alerones, elevadores y timón. Se muestra en caso para valores de los parámetros tales que $\delta'_\alpha, \delta_e, \delta_r > 0$ . [9] <i>Imagen modificada.</i>	6
1.6.	Ángulo de ataque $\alpha$ y ángulo <i>sideslip</i> $\beta$ . [9] <i>Imagen modificada.</i> . . . . .	7
1.7.	Triángulos involucrados en la determinación de $\alpha$ y $\beta$ . . . . .	8
1.8.	Cuerda aerodinámica para un perfil infinito y homogéneo. [2] <i>Imagen modificada.</i> . . . . .	8
1.9.	<i>Wing Span</i> y ángulo de <i>sweepback</i> . [2] <i>Imagen modificada.</i> . . . . .	9
1.10.	Vista lateral del <i>BEAVER</i> . [5]. . . . .	10
2.1.	Ilustrativa de los distintos sistemas de referencia a utilizar y las coordenadas $(x, y, z)$ . . . . .	13
2.2.	Esquema de las distintas rotaciones en base a los ángulos de <i>Euler</i> . Por convención desde SI a SB. . . .	13
2.3.	Modelado de la atmósfera. . . . .	17
2.4.	Relación funcional entre $C_L$ y $\alpha$ para el modelo del <i>BEAVER</i> y el perfil NACA 64 A 410. . . . .	26
2.5.	Relación entre $C_D$ y $C_L$ para el modelo del <i>BEAVER</i> y el perfil NACA 64 A 410. . . . .	26
3.1.	Modelo MDL elaborado en base a las ecuaciones del MVE. . . . .	31
3.2.	Modelo MDL elaborado para el bloque funcionF. . . . .	32
3.3.	Modelos MDLs que implementan la fuerza neta y el momento neto sobre el sistema. . . . .	33
3.4.	Modelo MDL elaborado para el bloque <b>Peso B</b> . . . . .	33
3.5.	Modelo MDL elaborado para el bloque <b>FAero B</b> . . . . .	33
3.6.	Modelo MDL elaborado para el bloque <b>TAero B</b> . . . . .	34
3.7.	Modelo MDL elaborado para el bloque <b>FT Motor</b> . . . . .	34
3.8.	Trayectorias predichas por el simulador para la prueba 1. . . . .	37
3.9.	Trayectorias para la prueba 1. <b>Azul:</b> simulador implementado. <b>Rojo:</b> utilizando el sistema 6DoF de <i>Matlab</i> . . . . .	37
3.10.	Trayectorias para la prueba 1, ampliación de los gráficos $xz$ y $\varphi(t)$ . <b>Azul:</b> simulador implementado. <b>Rojo:</b> utilizando el sistema 6DoF de <i>Matlab</i> . . . . .	38
7.1.	Diseño de la integración del <i>software</i> . . . . .	67
7.2.	Menú principal del <i>software</i> integrado. . . . .	67
7.3.	Menú para el simulador del lazo abierto del <i>software</i> integrado. . . . .	68
7.4.	Menú para el generador de trayectorias del <i>software</i> integrado. . . . .	68
7.5.	Menú para el simulador del lazo cerrado del <i>software</i> integrado. . . . .	69

# Bibliografía

---

- [1] Ingeniería de Control Moderna - Katsuhiko Ogata. Tercera edición 1998.
- [2] Fluid Mechanics - Pijush K. Kundu, Ira M. Cohen. Segunda edición 2002.
- [3] Airplane Flight Dynamics and Automatic Flight Control, Parte I - Jan Roskam. Tercera edición 2001.
- [4] Airplane Flight Dynamics and Automatic Flight Control, Parte II - Jan Roskam. Tercera edición 2001.
- [5] FDC 1.2 - A Simulink Toolbox for Flight Dynamics and Control Analysis - Marc Rauw. Segunda edición 2001.
- [6] Flight Dynamics Principles, A linear systems approach to aircraft stability and control - Michael V. Cook. Segunda edición 2007.
- [7] Diseño de un vehículo aéreo no tripulado: AMFO-1/Bobby, Documentación del Proyecto de Fin de Carrera - Andrés Chappe, Rodrigo Sosa, Andrés Suttner. 2007.
- [8] Applied Nonlinear Control - Jean-Jacques E. Slotine, Weiping Li. 1991.
- [9] <http://www.aviacion.cl>
- [10] Stability and control derivatives of the De Havilland 'BEAVER' aircraft - J. A. Mulder, R. T. H. Tjee. Report LR-556. 1988.
- [11] Fundamentos de termodinámica - J. Van Wylen, R. E. Sonntag. Segunda edición.
- [12] Introduction to aerodynamics of flight - T. A. Talay. NASA SP-367 1975.
- [13] [www.wikipedia.com](http://www.wikipedia.com)
- [14] Theory of wing sections - Ira H. Abbot, Albert E. Von Doenhoff. Segunda edición 1959.