

Proyecto Fin de Máster
Máster Universitario en Ingeniería Aeronáutica

Diseño de un Sistema de Actitud para Aviones No
Tripulados Basado en el Control Predictivo.

Autor: Carlos Díaz de Mayorga Ramos

Tutor: Dr. Francisco Gavilán Jiménez

Dep. de Ingeniería Aeroespacial y Mecánica de
Fluidos

Escuela Técnica Superior de Ingeniería

Sevilla, 2017



Proyecto Fin de Máster
Máster Universitario en Ingeniería Aeronáutica

Diseño de un Sistema de Actitud para Aviones No Tripulados Basado en el Control Predictivo.

Autor:

Carlos Díaz de Mayorga Ramos

Tutor:

Francisco Gavilán Jiménez

Profesor contratado Doctor interino

Dep. de Ingeniería Aeroespacial y Mecánica de Fluidos

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Máster: Diseño de un Sistema de Actitud para Aviones No Tripulados Basado en el
Control Predictivo.

Autor: Carlos Díaz de Mayorga Ramos
Tutor: Francisco Gavilán Jiménez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

*A mi yaya, siempre te tendré
presente.*

Agradecimientos

A mis padres, mi hermana y mis abuelas por confiar siempre en mí y apoyarme durante todos estos años. Por mostrarme el camino para seguir siempre hacia adelante, enseñándome que nunca hay que rendirse; y en especial a mi padre, por haberme ayudado además en la corrección de la redacción de este trabajo.

A mis amigos y mi novia, por estar cuando más los necesitaba, ofreciéndome su apoyo y animándome en los momentos más difíciles.

A mis compañeros de la universidad, por ayudarme siempre que lo he necesitado, por brindarme esos magníficos momentos en su compañía y por hacerme estos años mucho más agradables.

Al tutor de este proyecto, Francisco Gavilán Jiménez, por atenderme en todo momento que me haya sido necesario y ayudar en gran medida a la consecución del mismo.

Resumen

En este trabajo se trata el problema del diseño de un sistema de control de actitud predictivo para aviones no tripulados, centrándose en la formulación de un modelo de controlador para, posteriormente, realizar un conjunto de simulaciones en un entorno de Matlab/Simulink y analizar el comportamiento del sistema.

Para ello se parte de un modelo avión de avión de 6 grados de libertad, el cual es simplificado mediante la introducción de diversas hipótesis. Seguidamente, se genera un modelo de controlador predictivo (MPC) basado en la formulación simplificada de la aeronave, al cual se le introducen una serie de correcciones que tienen como objetivo una mejora de su respuesta y comportamiento. Tras esto se implementa el sistema en un modelo Simulink que simula el comportamiento del avión no tripulado Céfiro, diseñado por el grupo investigador del departamento de ingeniería aeroespacial de la Universidad de Sevilla.

Todo ello tendrá como objetivo analizar la respuesta que presenta el controlador ante diferentes situaciones para, de esta forma, valorar una futura integración dentro del sistema global de una aeronave real.

ÍNDICE

Agradecimientos	ix
Resumen	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvii
Notación	xix
1 Introducción	1
1.1 <i>Objetivos</i>	3
1.2 <i>Estructura del proyecto</i>	3
2 Modelo del Avión	5
2.1 <i>Desarrollo del modelo de avión</i>	8
2.1.1 Modelo no lineal de 6 grados de libertad	9
2.1.2 Modelo de 3 grados de libertad	16
2.2 <i>Estimación de parámetros</i>	17
3 Formulación del controlador predictivo	19
3.1 <i>Introducción al control predictivo</i>	19
3.1.1 Breve reseña histórica	19
3.1.2 Descripción	20
3.2 <i>Diseño del sistema de control de actitud basado en un controlador predictivo (MPC)</i>	21
3.2.1 MPC General	22
3.2.2 MPC Longitudinal	39
3.2.3 MPC Lateral-direccional	46
3.2.4 Descripción del modelo Simulink e implementación en el modelo original	50
4 Resultados	57

4.1	<i>Modelo original de MPC con control en u, γ, β y ϕ</i>	58
4.2	<i>Modelo original de MPC con control en $u, \gamma, q, \beta, \phi$ y p</i>	62
4.3	<i>Modelo de MPC con corrección en permanente y control en $u, \gamma, q, \beta, \phi$ y p</i>	66
4.4	<i>Modelo de MPC con saturaciones, corrección en permanente y control en $u, \gamma, q, \beta, \phi$ y p</i>	70
4.5	<i>Comparativa entre quadprog y corrección de saturaciones. Tiempos de computación.</i>	75
5	Conclusiones y líneas futuras	85
	Referencias	89
	Apéndices	91
	<i>Apéndice A. Montaje matrices F y G</i>	91
	<i>Apéndice B. Montaje matrices R y Q</i>	93
	<i>Apéndice C. Generador de trayectorias</i>	94
	<i>Apéndice D. Controladores</i>	100
	<i>Apéndice E. Saturaciones</i>	102

ÍNDICE DE TABLAS

Tabla 3-1 Valores de las derivadas de estabilidad del problema lateral-direccional.	27
Tabla 3-2 Valores de las derivadas de estabilidad del problema longitudinal.	27

ÍNDICE DE FIGURAS

Figura 2.1. Estructura general de un modelo de avión.	5
Figura 2.2. Notación	11
Figura 3.1 Esquema de funcionamiento del sistema de control de actitud.	21
Figura 3.2 Diagrama de flujo del proceso de resolución del problema de optimización con restricciones.	35
Figura 3.3 Error en régimen permanente de ϕ	37
Figura 3.4 Error en régimen permanente de u .	38
Figura 3.5 Error en régimen permanente de β .	38
Figura 3.6. Esquema de realimentación del MPC longitudinal.	44
Figura 3.7 Bloque para representar las nuevas variables de estado.	46
Figura 3.8 Modelo general del sistema.	50
Figura 3.9 Entorno de simulación original.	51
Figura 3.10 Modelo del sistema de control de actitud.	51
Figura 3.11 Estructura del controlador disgregada.	52
Figura 3.12 Bloque Longitudinal Controller.	54
Figura 3.13 Bloque Lateral Direccional Controller.	54
Figura 3.14 Controlador del modelo original.	56
Figura 4.1 Caso 1, velocidad aerodinámica en el eje x.	58
Figura 4.2 Caso 1, ángulo de trayectoria.	59
Figura 4.3 Caso 1, velocidad angular de cabeceo.	59
Figura 4.4 Caso 1, velocidad aerodinámica en el eje y.	60
Figura 4.5 Caso 1, ángulo de balance.	61
Figura 4.6 Caso 1, velocidad angular de alabeo.	61
Figura 4.7 Caso 2, velocidad aerodinámica en el eje x.	62
Figura 4.8 Caso 2, ángulo de trayectoria.	63

Figura 4.9 Caso 2, velocidad angular de cabeceo.	63
Figura 4.10 Caso 2, velocidad aerodinámica en el eje y.	64
Figura 4.11 Caso 2, ángulo de balance.	65
Figura 4.12 Caso 2, velocidad angular de alabeo.	65
Figura 4.13 Caso 3, velocidad aerodinámica en el eje x.	66
Figura 4.14 Caso 3, ángulo de trayectoria.	67
Figura 4.15 Caso 3, velocidad angular de cabeceo.	67
Figura 4.16 Caso 3, velocidad aerodinámica en el eje y.	68
Figura 4.17 Caso 3, ángulo de balance.	69
Figura 4.18 Caso 3, velocidad angular de alabeo.	69
Figura 4.19 Caso 4, velocidad aerodinámica en el eje x.	70
Figura 4.20 Caso 4, ángulo de trayectoria.	71
Figura 4.21 Caso 4, velocidad aerodinámica en el eje y.	71
Figura 4.22 Caso 4, ángulo de balance.	72
Figura 4.23 Simulación completa, velocidad aerodinámica en el eje x.	73
Figura 4.24 Simulación completa, ángulo de trayectoria.	74
Figura 4.25 Simulación completa, velocidad aerodinámica en el eje y.	74
Figura 4.26 Simulación completa, ángulo de balance.	75
Figura 4.27 Caso 5, velocidad aerodinámica en el eje x.	76
Figura 4.28 Caso 5, ángulo de trayectoria.	76
Figura 4.29 Caso 5, velocidad aerodinámica en el eje y.	77
Figura 4.30 Caso 5, ángulo de balance.	77
Figura 4.31 Zoom ángulo de trayectoria.	78
Figura 4.32 Zoom ángulo de balance.	79
Figura 4.33 Comparativa de métodos, velocidad aerodinámica en el eje x.	80
Figura 4.34 Comparativa de métodos, ángulo de trayectoria.	80
Figura 4.35 Comparativa de métodos, velocidad aerodinámica en el eje y.	81
Figura 4.36 Comparativa de métodos, ángulo de balance.	81
Figura 4.37 Tiempos de cálculo para el método diseñado, lateral-direccional.	82
Figura 4.38 Tiempos de cálculo para el método diseñado, longitudinal.	82
Figura 4.39 Tiempos de cálculo del quadprog, lateral-direccional.	83
Figura 4.40 Tiempos de cálculo del quadprog, longitudinal.	83

NOTACIÓN

\sin, s	Función seno
\cos, c	Función coseno
\tan	Función tangente
\sec	Función secante
$a \times b, a \wedge b$	Producto vectorial entre los vectores a y b.
$a \cdot b$	Producto vectorial entre los vectores a y b.
$\sum_{i=1}^n \xi_i$	Sumatorio desde 1 hasta n de las correspondientes ξ_i
\hat{g}	Vector adimensionalizado
v^T	Vector traspuesto
A^{-1}	Matriz inversa
$\frac{\partial f}{\partial y}$	Derivada de la función f respecto de y
$\sum_i \xi_i$	Sumatorio para todo i de ξ_i
R^{BH}	Matriz de rotación entre los sistemas de ejes B y H.
\dot{a}	Derivada de la variable a
$u(t)$	Dependencia de la variable u del tiempo.
∇	Operador gradiente

1 INTRODUCCIÓN

Desde los comienzos de la aviación se han llevado a cabo numerosas investigaciones y desarrollos encaminados a la consecución de una aeronave que sea capaz de realizar un vuelo sin necesidad de que exista un piloto en cabina. En este marco, aparece el concepto de los aviones no tripulados (UAVs), los cuales resultan de gran interés en numerosas aplicaciones. La principal razón de ello se debe a que este tipo de aeronaves presentan unas características especiales que las distinguen del resto y que pueden resultar de gran ventaja en muchas misiones, tanto en el ámbito civil como en el militar.

A lo largo de la historia, el desarrollo de los UAVs ha estado fuertemente condicionado por los avances implementados por la industria militar. Así, durante el transcurso de la primera Guerra Mundial se crearon los primeros prototipos de vehículos no tripulados, los cuales no tuvieron un gran éxito. Los años posteriores, tras el fracaso de los primeros UAVS y el fin de la primera Guerra Mundial, fueron bastantes oscuros en cuanto a innovaciones y nuevos desarrollos de estas aeronaves.

Años después, frente a la tesitura de una segunda Guerra Mundial, la industria aeronáutica vuelve a su auge produciéndose numerosos avances en la electrónica y los sistemas embarcados en el avión. En este sentido, las investigaciones y desarrollos en el campo de los vehículos no tripulados se multiplicaron. Tanto es así que en 1939 Reginald Denny creó la compañía “Radioplane Company” (Actual Northrop Grumman) y, junto a un equipo de ingenieros expertos en tecnología de radio, crearon una aeronave grande de control remoto. Dicha aeronave tendría un éxito considerable y fue producida en cadena bajo el nombre de OQ Targets.

Durante el transcurso de la segunda Guerra Mundial Adolf Hitler encargó la creación de una bomba voladora con el fin de usarse contra objetivos no militares (Vergeltungswaffe-1, V-1). Ante el éxito del V-1, la marina estadounidense comienza el desarrollo de sus propios UAVs para contrarrestar los ataques no tripulados alemanes surgiendo los modelos BQ-7 y PB4Y-1. Los años posteriores al fin de la segunda Guerra Mundial fueron poco alentadores en cuanto a avances en este tipo de vehículos, que

pasaron a usarse como blancos aéreos para el entrenamiento y pruebas de sistemas de defensa antiaérea.

Durante la guerra de Vietnam, los UAVs adquieren un protagonismo considerable, ya que se comienza a explotar el potencial de estos aviones no tripulados en escenarios bélicos. Así, en el año 1960 la fuerza aérea de EE. UU. inicia su primer programa para diseñar un avión invisible modificando los UAVs de combate para misiones de reconocimiento. El resultado de este programa fue el Ryan Firebee AQM-34, que se usó para misiones de reconocimiento en zonas hostiles y que permitió a la USAF aumentar su capacidad operacional sin poner en riesgo ninguna vida humana.

A pesar de todos los avances que se iban desarrollando a lo largo de los años en los UAVs, no sería hasta las décadas de los 80 y 90 cuando se produce el crecimiento exponencial de estas aeronaves. La principal razón de esta nueva tendencia de crecimiento reside en los avances en electrónica que se produjeron durante esos años. El aumento de la capacidad de cálculo y la minimización y reducción de peso de los computadores embarcados permitió extender el campo de aplicación de los UAVs y, por tanto, favoreció el avance y mejora de estas aeronaves. En la actualidad, los UAVs son una parte importante de cualquier ejército del mundo con alta tecnología y su uso se extiende desde la realización de misiones de seguimiento hasta operaciones de combate con UAVs armados.

A pesar del dominio militar en el campo de aplicaciones de los UAVs, en las últimas décadas se ha producido un incremento en el uso de estos aviones en ámbito civil. En particular, los UAVs medios y ligeros han sido implementados en numerosos sectores para la automatización de tareas repetitivas o peligrosas, tales como monitorización de cultivos, vigilancia, control de incendios, industria minera y pesquera, etc. Todo ello se debe al bajo coste de construcción, mantenimiento y operación que presentan este tipo de aeronaves.

En suma, se puede afirmar que los avances en el campo de los vehículos aéreos no tripulados están creciendo en los últimos años debido al aumento que ha surgido en su uso. Todo ello ha sido posible gracias a las nuevas propuestas y mejoras de los sistemas de guiado y control de actitud que embarcan estas aeronaves, los cuales han ampliado su precisión y eficiencia favoreciendo así su utilización en diversas aplicaciones.

En esta línea, hay que hacer referencia a la inserción de los modelos predictivos de control en el ámbito aeronáutico. Estos modelos permiten resolver el problema de control de actitud teniendo en cuenta las acciones futuras que deberá llevar a cabo la aeronave. Este carácter predictivo favorece el aumento de estabilidad del sistema general, lo cual, junto con la robustez y eficacia que suelen presentar estos modelos, hace de estos sistemas un atractivo de estudio. Este último punto, unido al

aumento de capacidad computacional de los procesadores embarcados en UAVs, que ha permitido el procesamiento de los algoritmos de estos modelos predictivos, han servido de motivación para el desarrollo de este proyecto.

1.1 Objetivos

Como punto de partida para este trabajo se dispone de un entorno de simulación compuesto por un controlador de actitud, un modelo de aeronave de seis grados de libertad (Céfiro UAV) y un modelo de Tierra Plana sin viento. En base a ello, el objetivo principal del proyecto se fija en el diseño de un sistema de control de actitud basado en el control predictivo y su posterior modelado en Simulink, para lo cual se deberá definir un modelo de predicción y un problema de optimización.

Una vez creado el modelo se pretende introducir una serie de correcciones que mejoren el comportamiento del sistema y la respuesta de la aeronave. Finalmente, este nuevo modelo deberá ser integrado dentro del entorno de simulación de partida.

En cuanto a las correcciones a introducir en el modelo original de MPC, surgen varios objetivos secundarios

- Inclusión de las saturaciones de las señales de control en la resolución del problema de control.
- Reducción de los tiempos de computación que requiere el controlador. Para ello se plantea un método alternativo al convencional para la resolución del problema de optimización cuadrática con restricciones.
- Eliminación de los errores que aparecen en el control en régimen permanente.

Por último, con el propósito de analizar y estudiar el modelo diseñado, se realizan un conjunto de simulaciones para cada una de las casuísticas presentadas.

1.2 Estructura del proyecto

En esta sección se va a realizar un resumen de los diferentes apartados de esta memoria para obtener una visión global del desarrollo del proyecto.

En base al objetivo final del trabajo, que es el diseño del mencionado controlador, se necesita definir previamente un modelo de avión que servirá de punto de partida para el desarrollo del MPC. Por ello se comienza, en el capítulo 2 de esta memoria, describiendo los diferentes modelos de avión que intervienen en el sistema diseñado.

Seguidamente, con el modelo base ya definido, en el capítulo 3 se desarrolla la formulación básica del controlador. Para ello, primero se realiza una linealización del modelo de avión original. Posteriormente, en base a un conjunto de hipótesis, se formula el modelo general del controlador, esto es, el conjunto de ecuaciones que definen el modelo de predicción y el problema de optimización cuadrática que debe resolver el controlador. Con las ecuaciones generales definidas, se le introducen una serie de correcciones, que permiten mejorar el comportamiento del sistema, se procede a su particularización para las dinámicas longitudinal y lateral-direccional y, finalmente, se lleva a cabo la integración del controlador en el entorno de partida.

De esta manera, en el capítulo 4, se realizan diversas simulaciones que permiten obtener un conjunto de resultados relativos a la respuesta del controlador.

Para finalizar, en el último capítulo se resumen los principales resultados, se obtienen las correspondientes conclusiones y se proponen, en base a ellas, posibles líneas de investigación y trabajo futuras.

2 MODELO DEL AVIÓN

En el presente capítulo se va a llevar a cabo la descripción del modelo de avión que será utilizado en las distintas simulaciones presentadas a lo largo de este trabajo. De la misma manera, dado que éste se encuentra dentro de un sistema general compuesto por diversos modelos, se procederá a dar una breve explicación de todos ellos con el objetivo de ofrecer una visión global del funcionamiento del sistema a pesar de que no sean el objeto de estudio de este trabajo.

Partiendo de una definición general, un modelo no es más que una representación simplificada de un sistema, elaborada para comprender, predecir y controlar el comportamiento del mismo. Es decir, hablando en términos ingenieriles, un modelo no es más que un conjunto de ecuaciones que permite simular el comportamiento de un sistema ante una situación determinada.

Para el caso concreto de un avión, el conjunto de ecuaciones o modelo global que describe el sistema debe ser capaz de determinar el comportamiento del mismo cuando éste se ve sometido a una serie de acciones externas que lo obligan a modificar su estado. Dicho modelo global está compuesto por cuatro elementos básicos, los cuales serán descritos a continuación, que son comunes a cualquier sistema que pretenda aproximar el comportamiento de una aeronave (Figura 2.1).

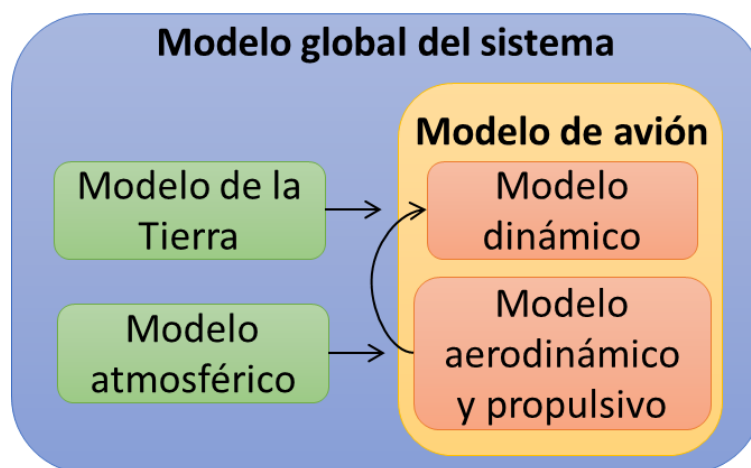


Figura 2.1. Estructura general de un modelo de avión.

- **Modelo dinámico**

Tal y como queda recogido en la referencia [1], este modelo está constituido por todo el conjunto de ecuaciones diferenciales que permiten determinar la evolución de las variables de estado del avión en función de las fuerzas y momentos exteriores. Se basa tanto en las acciones aerodinámicas y propulsivas como en las fuerzas gravitatorias para determinar las variables de estado, por lo que está intrínsecamente relacionado con dichos modelos (geofísico y aerodinámico y propulsivo).

- **Modelo aerodinámico y propulsivo**

Partiendo de las variables de estado de la aeronave, permite calcular las fuerzas y momentos aerodinámicos que actúan sobre la misma. Al conjunto compuesto por este modelo y el modelo dinámico se le denomina modelo de avión.

- **Modelo geofísico**

Basándose en un modelo terrestre y gravitatorio, permite determinar la fuerza gravitatoria sobre la aeronave y posicionarla respecto a una referencia terrestre. Para el caso de estudio del presente trabajo se ha usado un modelo de Tierra Plana con gravedad constante; la razón de ello reside en que no se esperan grandes cambios de actitud ni grandes alcances por las características de la aeronave estudiada (avión ligero no tripulado de pequeñas dimensiones). Un mayor desarrollo del modelo de Tierra puede encontrarse en la referencia [1].

- **Modelo de atmósfera**

Este modelo va a determinar las propiedades que presenta la atmósfera en función de la posición de la aeronave. Además, se encarga de generar una parte de los inputs que necesita el modelo aerodinámico para realizar sus cálculos. En el caso de estudio tratado se ha hecho uso de un modelo de atmósfera internacional (ISA) utilizado por la gran mayoría de aviones. Un mayor desarrollo del modelo atmosférico puede encontrarse en la referencia [1].

La unión de estos cuatro modelos, tal y como ha sido comentado al comienzo del capítulo, debe generar un modelo global que debe ser capaz de aproximar el comportamiento general de la aeronave. Este modelo se basa en una serie de ecuaciones generales obtenidas mediante la aplicación de diversas hipótesis que permiten simplificar el problema de estudio y, en consecuencia, facilitan el modelado del mismo.

Entrando más en profundidad en el modelo de avión aparecen una serie de dificultades a la hora de establecer un modelo adecuado para todos los elementos que componen la estructura del sistema de control y guiado, ya que la tesis en la que se basa el desarrollo de este trabajo [1] utiliza una estructura jerárquica en la que el sistema de guiado y el de control no están integrados, sino que son dos módulos diferentes. Debido a ello, el autor de la misma se decanta por la utilización de modelos diferentes para cada uno de los elementos que componen el sistema de control y guiado.

- *Controlador de trayectorias o sistema de guiado*: es el sistema encargado de generar señales de control de alto nivel que mantienen a la aeronave dentro de la trayectoria de referencia eliminando las desviaciones que puedan surgir debido a las perturbaciones exteriores. A la hora de diseñar este controlador es recomendable evitar introducir complejidades innecesarias en el problema. Consecuentemente, se hace conveniente despreocuparse del problema de actitud en el modelado del controlador. En base a estas premisas, el autor de la referencia [1] decidió utilizar un modelo puntual del avión de tres grados de libertad que permite posicionar la aeronave con respecto a la trayectoria de referencia deseada. En la sección 2.1.2 de este capítulo se lleva a cabo el desarrollo en profundidad de dicho modelo.

Es importante notar que, como se ha mencionado en el capítulo 1, el presente trabajo se ha centrado en el desarrollo de un controlador de actitud predictivo de bajo nivel. En consecuencia, a la hora de realizar las simulaciones correspondientes se ha ignorado el controlador de trayectorias, introduciendo manualmente en el controlador de bajo nivel las salidas que proporcionaría el sistema de guiado. No obstante, es conveniente tener una visión global del sistema para poder entender con claridad el funcionamiento del mismo; por ello, se ha decidido incluir en este capítulo el planteamiento seguido en la tesis [1] a la hora de modelar la parte correspondiente al sistema de guiado.

- *Controlador de la actitud*: partiendo de las consignas generadas por el controlador de trayectorias este sistema generará las señales de control que deberán implementar los mandos de vuelo para que, manteniendo la aeronave estable, ésta siga las referencias generadas por el sistema de guiado.

Para este controlador se va a utilizar un modelo de seis grados de libertad que va a permitir describir tanto la velocidad como la actitud de la aeronave, y en el que, favorecido por la arquitectura jerárquica del sistema de control y guiado, se ha despreocupado la posición espacial de la misma.

Por último, con el fin de disponer de un modelo de alta precisión para simular los controladores diseñados y tener cierta garantía de estabilidad en dichas simulaciones, el autor de la mencionada tesis desarrolla un modelo con todos los grados de libertad posibles, incluyendo modelos aerodinámicos y propulsivos al máximo nivel de detalle.

En base a todo esto, en las siguientes secciones del capítulo se va a profundizar en la obtención de las ecuaciones que permitan modelar el sistema avión de la manera más general posible. Entrando en la estructura del capítulo, en la sección 2.1 se van a deducir las ecuaciones generales que permiten describir el movimiento del avión (modelo dinámico, modelo aerodinámico y modelo propulsivo) mediante la imposición de unas hipótesis generales. Finalmente, estos modelos dependerán de una serie de parámetros, los cuales han sido obtenidos y calculados en la tesis [1], por lo que en la sección 2.2 se realizarán las referenciaciones correspondientes para conocer el origen de dichos parámetros.

2.1 Desarrollo del modelo de avión

Como ha sido mencionado con anterioridad, a lo largo de esta sección, partiendo de una serie de hipótesis generales, se van a obtener con la mayor generalidad posible tanto las ecuaciones del movimiento de la aeronave como los modelos aerodinámicos y propulsivo, es decir, se pretende desarrollar un modelo de avión general que permita su adaptación a cualquier situación específica con la introducción de consideraciones e hipótesis particulares.

En esta línea, se van a presentar dos modelos de avión. El primero de ellos, de mayor complejidad y realismo posible, es un modelo de 6 grados de libertad. Éste, que ha sido implementado por el autor [1] para ensayar al máximo nivel de realismo los controladores diseñados en la tesis, se utilizará en este proyecto para probar el funcionamiento del controlador de actitud predictivo desarrollado.

Por otro lado, se hace uso de un modelo puntual de tres grados de libertad, desarrollado en la tesis [1] para el diseño del sistema de guiado, y que permite, despreciando el problema de actitud, determinar la posición del centro de gravedad de la aeronave respecto de un sistema de referencia dado, es decir, modela el problema de traslación del avión. Como ya se ha mencionado con anterioridad, a efectos prácticos de este trabajo este modelo no ha sido utilizado, ya que el objetivo del mismo se centra únicamente en el diseño del controlador de actitud de bajo nivel y se ha optado por generar manualmente la referencias que provendrían del controlador de trayectorias. No obstante, como se puede comprobar en el capítulo 3, una vez concluido el diseño del controlador predictivo longitudinal se le ha introducido una modificación, que permite cambiar las variables de estado que controla el sistema, para permitir una integración futura con el sistema de guiado.

Por último, se hace necesario recalcar que en las secciones posteriores de este capítulo se va a realizar una síntesis de las correspondientes secciones de la referencia [1]. Ello se debe a que el modelo de avión utilizado para probar el desarrollo del controlador predictivo diseñado es el mismo que el usado por el autor de la correspondiente tesis.

2.1.1 Modelo no lineal de 6 grados de libertad

Este modelo tiene como objetivo su implementación en el entorno de Simulink en el cual se vayan a llevar a cabo las diferentes simulaciones del controlador predictivo diseñado. La razón de su uso reside principalmente en que es el modelo de mayor complejidad de los desarrollados en [1], ya que recoge los modelos aerodinámicos y propulsivos más precisos y las ecuaciones más generales del movimiento.

2.1.1.1 Modelo dinámico

Tal y como se recoge en la referencia [2], las ecuaciones del movimiento que caracterizan una aeronave se dividen en dos grupos principales.

- Ecuaciones cinemáticas: están compuestas por las ecuaciones cinemáticas lineales, las cuales establecen una relación entre la posición y la velocidad lineal, y las ecuaciones angulares, que establecen las relaciones existentes entre la actitud y la velocidad angular.
- Ecuaciones dinámicas: relacionan las velocidades lineales y angulares con las fuerzas y momentos aplicados.

A lo largo de este apartado, se va a proceder a la obtención de ambos conjuntos de ecuaciones, partiendo de unas ecuaciones generales a las que se aplicarán las siguientes hipótesis [1] [2] [3].

- El avión es un sólido rígido de 6 grados de libertad.
- La aeronave presenta un plano de simetría ($x_B y_B$).
- Se usa un modelo de tierra plana, lo cual implica que
 - el sistema topocéntrico es inercial,
 - el modelo de gravedad constante.
- Se usará un modelo de Tierra sin viento, lo cual implica que la velocidad respecto a tierra coincide con la velocidad aerodinámica de la aeronave. ($\vec{V}_c = \vec{V}$)
- Se desprecia el efecto de las partes móviles internas del avión (motor, hélice, etc.) en el momento cinético del mismo.

- Se considera que la masa y los momentos de inercia permanecen constantes, ya que la aeronave de trabajo es de propulsión eléctrica y no pierde combustible durante el vuelo.

Teniendo presentes todas estas hipótesis se comienza con la formulación de las ecuaciones dinámicas del movimiento.

Partiendo de referencias clásicas de la mecánica del vuelo [4] [5] [6], se puede afirmar que las expresiones más generales de las ecuaciones del movimiento de la aeronave vienen dadas por las ecuaciones de cantidad de movimiento y de momento cinético

$$\sum \vec{F}_{ext} = \vec{F}_{A,T} + \vec{F}_g = m \left. \frac{d\vec{V}_c}{dt} \right|_i \quad (2.1)$$

$$\sum \vec{M}_{ext} = \vec{M}_{A,T} = m \left. \frac{d\vec{h}_c}{dt} \right|_i, \quad h_c = \bar{I}_{CG} \vec{\omega} \quad (2.2)$$

donde \vec{V}_c es la velocidad del centro de gravedad del avión respecto a tierra, $\vec{\omega}$ es la velocidad angular del avión respecto de a tierra, $\vec{F}_{A,T} + \vec{M}_{A,T}$ son las fuerzas y momentos aerodinámicos propulsivos, \vec{F}_g es la fuerza gravitatoria, m es la masa y \bar{I}_{CG} es el tensor de inercia en el centro de gravedad.

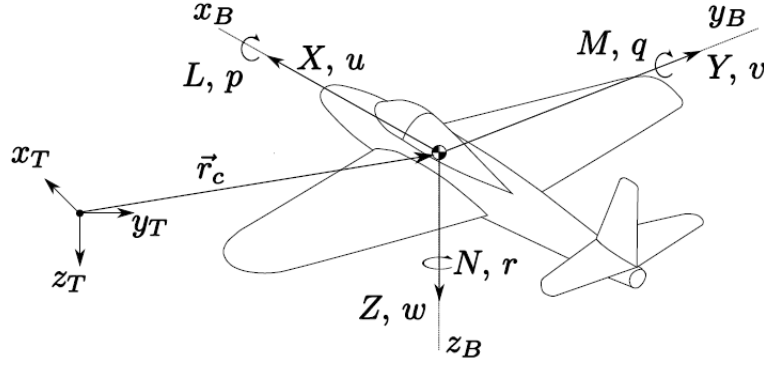
Con el fin de alcanzar una formulación más sencilla de las fuerzas y momentos se decide emplear un sistema de ejes ligado al avión, esto es, un sistema de ejes cuerpo que puede ser consultado en el apéndice A de la referencia [1]. De esta manera, proyectando las ecuaciones dinámicas en este sistema de ejes se consigue un tensor de inercia constante; sin embargo, dado que el sistema de ejes cuerpo no es inercial, se deberá aplicar la fórmula de Poisson para el cálculo de las derivadas. Con la aplicación de las hipótesis anteriormente comentadas y la proyección de las ecuaciones (2.1) y (2.2) en el sistema mencionado, la formulación queda de la siguiente manera

$$\vec{F}_{A,T} + \vec{F}_g = m \left[\left. \frac{d\vec{V}_c}{dt} \right|_B + \vec{\omega}_{BT} \wedge \vec{v}_{BT} \right] \quad (2.3)$$

$$\vec{M}_{A,T} = I_{CG} \left[\left. \frac{d\vec{\omega}_{BT}}{dt} \right|_B + \vec{\omega}_{BT} \wedge \vec{\omega}_{BT} \right] \quad (2.4)$$

donde el subíndice BT representa la velocidad, ya sea angular o lineal, del sistema de ejes cuerpo (B) respecto del sistema topocéntrico (T).

Para poder obtener un conjunto de ecuaciones escalares, se introduce una nueva notación como la que aparece en la Figura 2.2 obtenida de la referencia [2].



$$[\vec{V}]^B = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad [\vec{\omega}]^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad [\vec{F}_{A,T}]^B = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad [\vec{M}_{A,T}]^B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \quad [\vec{r}_c]^T = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

Figura 2.2. Notación.

Además, se deberá tener en cuenta la hipótesis de Tierra Plana, según la cual la gravedad se puede expresar en el sistema de ejes cuerpo como sigue

$$\vec{F}_g = R^{BH} \begin{bmatrix} 0 \\ 0 \\ g_0 \end{bmatrix} \quad (2.5)$$

siendo R^{BH} la matriz de transformación del sistema de ejes horizontales al sistema de ejes cuerpo mostrada en el apéndice A.

De esta manera, las ecuaciones (2.1) y (2.2) pueden escribirse de la siguiente manera.

$$X - mg_0 \sin \theta = m(\dot{u} - rv + qw) \quad (2.6)$$

$$Y + mg_0 \cos \theta \sin \phi = m(\dot{v} + ru - pw) \quad (2.7)$$

$$Z + mg_0 \cos \theta \cos \phi = m(\dot{w} - qu + pv) \quad (2.8)$$

$$L = I_x \dot{p} - I_{xz} \dot{r} + (I_z - I_y)qr - I_{xz}pq \quad (2.9)$$

$$M = I_y \dot{q} + I_{xz}(p^2 - r^2) + (I_x - I_z)pr \quad (2.10)$$

$$N = I_z \dot{r} - I_{xz} \dot{p} + (I_y - I_x)pq + I_{xz}qr \quad (2.11)$$

Una vez realizada la formulación de las ecuaciones dinámicas del movimiento se deberá completar el problema con las ecuaciones cinemáticas, las cuales, como se ha mencionado al comienzo de la sección, se dividen en dos grupos.

▪ Ecuaciones lineales

Las ecuaciones lineales nos permiten conocer la posición del vehículo a partir de su velocidad, esto es

$$\vec{V}_c = \left[\frac{d\vec{r}_c}{dt} \right]^T = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} \quad (2.12)$$

Tal y como se ha realizado en el caso de las ecuaciones dinámicas, la formulación del problema se va a realizar en ejes cuerpo, por lo que habrá que hacer la transformación a este sistema de referencia:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{bmatrix} = R^{TB} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} uc_\theta c_\psi + v(s_\theta s_\phi c_\psi - s_\psi c_\phi) + w(s_\theta c_\phi c_\psi + s_\psi s_\phi) \\ uc_\theta s_\psi + v(s_\theta s_\phi s_\psi + c_\psi c_\phi) + w(s_\theta c_\phi s_\psi - c_\psi s_\phi) \\ -us_\theta + vs_\phi c_\theta + wc_\phi c_\theta \end{bmatrix} \quad (2.13)$$

▪ Ecuaciones angulares

Estas ecuaciones nos permiten obtener una relación entre los parámetros de actitud (ángulos de Euler) y la velocidad angular. Para su obtención será necesario referirse a la secuencia de rotaciones que define la transformación $F_H \leftrightarrow F_B$ (Apéndice A de la referencia [1], ecuación A.8). De esta manera, se puede definir el vector velocidad angular de la siguiente forma:

$$\vec{\omega} = \dot{\psi} \vec{k}_H + \dot{\theta} \vec{j}_{X_1} + \dot{\phi} \vec{i}_B \quad (2.14)$$

Proyectando esta ecuación en ejes cuerpo

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \dot{\psi} R^{BH} [\vec{k}_H]^H + \dot{\theta} R^{BX_2} R^{X_2 X_1} [\vec{j}_{X_1}]^{X_1} + \dot{\phi} [\vec{i}_B]^B \quad (2.15)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} s_\theta \\ \dot{\theta} c_\phi + \dot{\psi} c_\theta s_\phi \\ -\dot{\theta} s_\phi + \dot{\psi} c_\theta c_\phi \end{bmatrix} \quad \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p + (qs_\phi + rc_\phi)t_\theta \\ qc_\phi - rs_\phi \\ (qs_\phi + rc_\phi)\sec\theta \end{bmatrix} \quad (2.16)$$

El grupo de ecuaciones (2.6), (2.7), (2.8), (2.9), (2.10), (2.11), (2.13) y (2.16) constituye un sistema de 12 ecuaciones diferenciales que permite calcular las 12 variables de estado

$$\vec{x} = [x_c \quad y_c \quad z_c \quad \phi \quad \theta \quad \psi \quad u \quad v \quad w \quad p \quad q \quad r] \quad (2.17)$$

Para cerrar el problema, es necesario conocer las expresiones de las fuerzas y momentos aerodinámicos y propulsivos. En general

$$F_{A,T}, M_{A,T} = f(\vec{x}, \vec{u}) \quad (2.18)$$

donde \vec{u} es el vector de control y \vec{x} es el vector de estados.

Por último, se hace hincapié en que el problema de trayectoria se encuentra desacoplado del resto, es decir, que pueden resolverse las ecuaciones de velocidad y actitud de forma independiente.

2.1.1.2 Modelo propulsivo

Al igual que se ha realizado en el modelo aerodinámico, para la obtención de las ecuaciones correspondientes al modelo propulsivo el autor de la referencia [1] aplica las siguientes hipótesis.

- El sistema propulsivo de la aeronave está compuesto por un sistema de baterías, un motor eléctrico (con su correspondiente controlador electrónico de velocidad) y una hélice de paso fijo.
- La fuerza propulsiva está alineada con el eje x del avión (en el sistema de ejes cuerpo).

Teniendo presente estas hipótesis simplificadoras se procede a la formulación del modelo propulsivo. Para ello, se parte de la ecuación 2.37 de la referencia [1] en la que se modela el empuje que proporciona la hélice con la siguiente ecuación

$$\vec{F}_T = \frac{4}{\pi^2} \rho R^4 \Omega^2 C_T \vec{l}_B \quad (2.19)$$

donde R es la radio de la hélice, Ω la velocidad angular de giro y C_T es el coeficiente de tracción de la hélice que depende del parámetro de avance (J) de la misma, es decir

$$C_T = f(J), \quad J = \frac{\pi u}{\Omega R} \quad (2.20)$$

Admitiendo, tal y como se ha planteado en las hipótesis generales, que el eje de la hélice es paralelo al eje x_B , el momento propulsivo solo tendrá una componente según dicho eje de valor

$$\vec{M}_T = -\frac{4}{\pi^3} \rho R^5 \Omega^2 C_P \vec{l}_B \quad (2.21)$$

siendo $C_P = \frac{\pi^3 P}{4\rho\Omega^3 R^5}$ el coeficiente de potencia de la hélice.

Tanto C_P como C_T son coeficientes que dependen del parámetro de avance de la hélice y, por lo tanto, son características de cada hélice. Los modelos que definen ambos coeficientes deben ser calculados mediante ensayos, y su formulación y obtención queda fuera del ámbito de este trabajo. No obstante, en la sección 2.3 de la referencia [1] se puede consultar el modelado de estos parámetros.

En último lugar, cabe destacar que a lo largo de este apartado se ha desarrollado un modelo propulsivo en función de la velocidad angular de giro; sin embargo, como se podrá comprobar en el capítulo 3, la variable de control que proporciona el MPC longitudinal es la posición de palanca (δ_t). Este obstáculo es salvado con facilidad teniendo en cuenta que el motor eléctrico del avión en el que se basa este trabajo dispone de un regulador interno de revoluciones, de esta manera se puede establecer una relación lineal entre la posición de palanca y las revoluciones conseguidas.

2.1.1.3 Modelo aerodinámico

Tal y como queda recogido en la ecuación (2.18), las fuerzas y momentos aerodinámicos depende de las variables de estado y de control. Dicha dependencia resulta no lineal, lo que supone una dificultad a la hora de obtener un modelo aerodinámico preciso.

Con el fin de reducir la complejidad del problema y facilitar el modelado del mismo, se realiza una adimensionalización de las variables. Esto permite trabajar con coeficientes adimensionales, sobre los cuales existe más literatura y conocimiento.

Como puede comprobarse en el apartado 2.2.1.3 de la referencia [1], por conveniencia se decide usar los ejes estabilidad (apéndice A de la mencionada referencia) a la hora de formular el modelo aerodinámico.

En base a esto las fuerzas y momentos aerodinámicos quedan expresados como sigue

$$\begin{pmatrix} X_A \\ Y_A \\ Z_A \end{pmatrix} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{pmatrix} \frac{1}{2} \rho V_a^2 S C_{X_S} \\ \frac{1}{2} \rho V_a^2 S C_{Y_S} \\ \frac{1}{2} \rho V_a^2 S C_{Z_S} \end{pmatrix} \quad (2.22)$$

$$\begin{pmatrix} L_A \\ M_A \\ N_A \end{pmatrix} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \begin{pmatrix} \frac{1}{2} \rho V_a^2 S b C_{l_S} \\ \frac{1}{2} \rho V_a^2 S \bar{c} C_{m_S} \\ \frac{1}{2} \rho V_a^2 S b C_{n_S} \end{pmatrix} \quad (2.23)$$

donde C_{X_S} , C_{Z_S} , C_{Y_S} , C_{l_S} , C_{m_S} y C_{n_S} son los coeficientes adimensionales de las fuerzas y momentos en ejes estabilidad, cuya estimación puede ser consultada en la sección 2.3 de la tesis [1], y S , b y \bar{c} son la superficie, la envergadura y la cuerda media del ala, respectivamente.

Con la definición de los coeficientes adimensionalizados de las variables aerodinámicas se está en disposición de seleccionar un modelo aerodinámico para ellos. Esta elección no es nada trivial ya que no existe ningún modelo universal que proporcione una función para estos coeficientes. No obstante, se tienen a disposición numerosos modelos con diferentes grados de complejidad.

En lo que concierne a este trabajo, la elección de modelo viene fijada por la del autor de la referencia [1], el cual opta por basarse en el modelo aerodinámico que implementa el programa DATCOM (usado para estimar los parámetros aerodinámicos).

A continuación, se muestran las ecuaciones que modelan los coeficientes y que han sido tomadas de la tesis ya mencionada

$$C_{X_S} = C_{X_\alpha}(\alpha) + C_{X_{\delta_e}}(\alpha, \delta_e) + C_{X_{\delta_f}}(\alpha, \delta_f) \quad (2.24)$$

$$C_{Z_S} = C_{Z_\alpha}(\alpha) + C_{Z_{\dot{\alpha}}}(\alpha) \frac{\dot{\alpha} \bar{c}}{2u_s} + C_{Z_q}(\alpha) \frac{q \bar{c}}{2u_s} + C_{Z_{\delta_e}}(\delta_e) + C_{Z_{\delta_f}}(\delta_f) \quad (2.25)$$

$$C_{Y_S} = C_{Y_\beta}(\alpha) \beta + C_{Y_p}(\alpha) \frac{pb}{2u_s} + C_{Y_r}(\alpha) \frac{rb}{2u_s} + C_{Y_{\delta_r}}(\delta_r) \quad (2.26)$$

$$C_{l_S} = C_{l_\beta}(\alpha) \beta + C_{l_p}(\alpha) \frac{pb}{2u_s} + C_{l_r}(\alpha) \frac{rb}{2u_s} + C_{l_{\delta_a}}(\delta_a) + C_{l_{\delta_r}}(\delta_r) \quad (2.27)$$

$$C_{m_S} = C_{m_\alpha}(\alpha) + C_{m_{\dot{\alpha}}}(\alpha) \frac{\dot{\alpha} \bar{c}}{2u_s} + C_{m_q}(\alpha) \frac{q \bar{c}}{2u_s} + C_{m_{\delta_e}}(\delta_e) + C_{m_{\delta_f}}(\delta_f) \quad (2.28)$$

$$C_{n_S} = C_{n_\beta}(\alpha) \beta + C_{n_p}(\alpha) \frac{pb}{2u_s} + C_{n_r}(\alpha) \frac{rb}{2u_s} + C_{n_{\delta_a}}(\alpha, \delta_a) + C_{n_{\delta_r}}(\delta_r) \quad (2.29)$$

donde u_s es la velocidad de vuelo de referencia.

Es importante realizar una buena definición de las deflexiones de las superficies de control, en concreto de los criterios seguidos para los signos de las mismas. De esta manera

- δ_e : deflexión del timón de profundidad. $\delta_e > 0$ implica momento de picado.
- δ_a : deflexión de alerones. $\delta_a > 0$ implica momento de balance positivo.
- δ_r : deflexión del timón de dirección. $\delta_r > 0$ implica momento de guiñada positivo.
- δ_f : deflexión de flaps. $\delta_f > 0$ implica un incremento del coeficiente de sustentación.

2.1.2 Modelo de 3 grados de libertad

Como se introdujo al principio del capítulo, para el controlador de trayectorias diseñado en la tesis en la que se basa este trabajo, el autor hace uso de un modelo de 3 grados de libertad que, despreciando la dinámica de la actitud, describe el movimiento del centro de masas del avión. Este modelo es ampliamente usado para esta tarea en la referencia [7].

Al igual que en el caso del modelo de 6 grados de libertad, se realizan una serie de hipótesis simplificadoras generales.

- Avión rígido y simétrico.
- Empuje paralelo al eje x_B .
- Tierra plana.
- Viento estacionario, horizontal.
- Masa constante (motor eléctrico).

En base a estas hipótesis, mediante una serie de desarrollos que se salen del ámbito de estudio de este trabajo se pueden obtener las siguientes ecuaciones escalares para el modelo de 3 grados de libertad

$$\frac{dx}{dt} = V_a \cos \gamma \cos \chi + w_x \quad (2.30)$$

$$\frac{dy}{dt} = V_a \cos \gamma \sin \chi + w_y \quad (2.31)$$

$$\frac{dh}{dt} = V_a \sin \gamma \quad (2.32)$$

$$\frac{m dV_a}{dt} = T \cos \alpha \cos \beta - D - mg \sin \gamma \quad (2.33)$$

$$m V_a \cos \gamma \frac{d\chi}{dt} = (T \cos \alpha \sin \beta - C) \cos \mu + (L + T \sin \alpha) \sin \mu \quad (2.34)$$

$$m V_a \frac{d\gamma}{dt} = (L + T \sin \alpha) \cos \mu - mg \cos \gamma \quad (2.35)$$

siendo V_a, γ, χ el módulo de la velocidad aerodinámica, el ángulo de trayectoria y el ángulo de rumbo, que definen el vector velocidad aerodinámica; m la masa de la aeronave; x, y, h son las coordenadas del centro de masas del avión (norte, este y actitud); t el tiempo; T es la tracción de la hélice; L la sustentación, D la resistencia aerodinámica y μ el ángulo de balance de la velocidad aerodinámica (ángulo de alabeo); y w_x, w_y son las componentes de a velocidad del viento según los ejes x, y del sistema topocéntrico.

Admitiendo la condición de vuelo simétrico, lo que implica un ángulo de resbalamiento nulo ($\beta = 0$) y $C = 0$, las ecuaciones anteriores quedan de la siguiente manera.

$$\frac{dx}{dt} = V_a \cos \gamma \cos \chi + w_x \quad (2.36)$$

$$\frac{dy}{dt} = V_a \cos \gamma \sin \chi + w_y \quad (2.37)$$

$$\frac{dh}{dt} = V_a \sin \gamma \quad (2.38)$$

$$m \frac{dV_a}{dt} = T \cos \alpha - D - mg \sin \gamma \quad (2.39)$$

$$mV_a \cos \gamma \frac{d\chi}{dt} = (L + T \sin \alpha) \sin \mu \quad (2.40)$$

$$mV_a \frac{d\gamma}{dt} = (L + Y \sin \alpha) \cos \mu - mg \cos \gamma \quad (2.41)$$

Por último, otra consecuencia de la condición de vuelo simétrico es que el ángulo de alabeo y el ángulo de balance coincidan ($\mu = \phi$), puesto que la sustentación y la resistencia están contenidas en el plano de simetría de la aeronave.

2.2 Estimación de parámetros

Todos los parámetros utilizados para la formulación de los diferentes modelos expuestos a lo largo de este capítulo han sido obtenidos por el autor de la tesis [1] en la que se basa este trabajo. Por lo cual, su obtención y valor pueden ser consultados en la sección 2.3 de dicha disertación.

3 FORMULACIÓN DEL CONTROLADOR

PREDICTIVO

El objetivo principal del presente trabajo consiste en el diseño de un controlador predictivo (MPC) y su implementación dentro del sistema general expuesto con anterioridad en el capítulo 2 de esta memoria. Para ello se ha formulado un modelo de control predictivo, el cual ha sido añadido en el entorno de simulación generado en la tesis doctoral [1], sustituyendo al sistema de control de actitud diseñado en dicha disertación. En resumidas cuentas, el objeto de este trabajo ha consistido en el diseño, implementación y posterior análisis de un sistema de control de actitud basado en el control predictivo.

En base a este objetivo el presente capítulo va a tener dos propósitos principales. Por un lado, se pretende situar al lector en el tema principal del mismo (el control predictivo), para lo que se realiza una pequeña introducción a este tipo de modelos de control en la sección 3.1. Por otro lado, una vez se haya situado al lector dentro de la materia, en la sección 3.2 se desarrollará toda la formulación relativa al modelo de controlador predictivo. Se comenzará con una formulación general a la cual se le añadirán una serie de correcciones que tiene como objetivo mejorar el comportamiento de los controladores, cuyo análisis será expuesto en la sección de resultados de esta memoria (4), seguidamente se particulariza el modelo para los casos de los controladores longitudinal y lateral-direccional, finalmente se describe el modelo Simulink del sistema de control de actitud creado y se expone el proceso de integración del mismo en el entorno de simulación de partida.

3.1 Introducción al control predictivo

3.1.1 Breve reseña histórica

Las investigaciones relativas a los MPC (Model Predictive Control) comienzan a principios de la década de los 60 del siglo pasado con el trabajo de Propoi [8]. Sin embargo, no sería hasta las décadas de los 70 y 80 cuando se potencia el desarrollo de este tipo de modelo de control. La mayor parte de

estos desarrollos provienen de métodos implementados en la industria para la resolución de problemas reales, los cuales no estaban basados en ningún fundamento teórico; ejemplo de ello son los métodos DMC, GPC, IDCOM, QDMC y SMOC. Al no estar basados en ninguna formulación teórica, dichos métodos no garantizaban una estabilidad en la solución.

En la década de los noventa aparecieron un gran número de publicaciones que se centraron tanto en el desarrollo de la teoría de estabilidad del controlador como en la formulación del problema de seguimiento de una referencia. Sin embargo, pese al incremento del número de estas publicaciones, el gran salto en el desarrollo de estos modelos se produce a comienzos del nuevo milenio con la creación de un modelo predictivo de control explícito; este hito daría paso a una nueva área de investigación en el campo del control de la actitud. De hecho, durante las dos últimas décadas se han llevado a cabo numerosos estudios relativos a los modelos de control predictivo y se han propuesto diversas formulaciones tanto para los MPC lineales como para los no-lineales. Todo ello ha convertido a estos modelos en una de las técnicas de control más prometedoras para sistemas multivariables con restricciones. [9]

3.1.2 Descripción

Los modelos de control predictivo son métodos avanzados de control de procesos basados en la optimización numérica de un horizonte de tiempo finito; es decir, el MPC resuelve un problema de control a futuro en base a un modelo del sistema y un criterio de optimalidad. Esto permite obtener la solución del problema en el instante de tiempo actual, a la par que se están teniendo en cuenta las acciones futuras dentro del horizonte de predicción. Particularizando para el caso de una aeronave, el MPC determinará un conjunto de señales de control y las correspondientes respuestas a futuro de la misma, basándose en el modelo de avión y el criterio de optimización impuestos.

Actualmente, el control predictivo es una de las técnicas de control más extendida en la industria, ya que presenta una base teórica bastante sólida y sus propiedades de estabilidad, optimalidad y robustez son bien conocidas. Además, a pesar de ser controladores fáciles de diseñar e implementar, los MPC pueden controlar sistemas con un gran número de variables.

No obstante, pese a todo el conjunto de características ventajosas presentadas, la gran virtud que presenta el MPC frente al resto de modelos de control de actitud, es su capacidad para trabajar con problemas que presentan algún tipo de restricciones (las cuales se pueden encontrar en cualquier aplicación ingenieril). Es esta característica la que hace interesante la implementación de este tipo de controladores en el control de actitud de aeronaves que, como bien es sabido, está fuertemente condicionado por las diversas restricciones que presentan los sistemas que intervienen en el proceso

de control. [9] [10]

3.2 Diseño del sistema de control de actitud basado en un controlador predictivo (MPC)

Los sistemas de control de actitud de un avión deben ser capaces de llevar a cabo el seguimiento de una referencia dada, asegurando la estabilidad y el buen comportamiento de la aeronave. Para ello, el avión tiene la capacidad de actuar sobre una serie de señales de control de la misma, esto es, las deflexiones de las superficies de control (timón de profundidad δ_e , alerones δ_a , timón de dirección δ_r) y la palanca de gases del motor (δ_t).

El funcionamiento del sistema es bastante simple. Partiendo del estado actual de la aeronave y de la referencia a seguir (inputs del sistema), el controlador de actitud, basándose en los modelos del controlador y del avión, determina las señales de control que deberán ser impuestas para seguir la referencia (outputs del sistema). Con la implementación de dichas señales y la incertidumbre que rodea al sistema se obtendrá el estado del avión en el instante posterior, el cual realimentará al controlador para volver a reiniciar el proceso. Este esquema de funcionamiento del sistema queda recogido en la Figura 3.1 obtenida de la referencia [1].

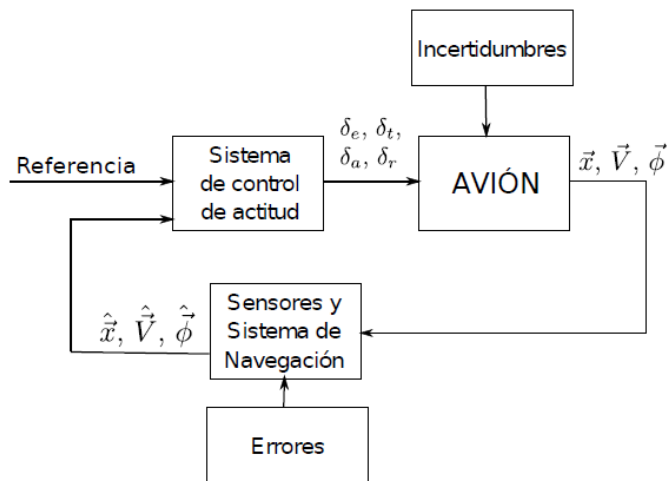


Figura 3.1 Esquema de funcionamiento del sistema de control de actitud.

Una vez expuesto de manera esquemática el proceso de control de la actitud se está en disposición de comenzar con la formulación del controlador predictivo implementado.

3.2.1 MPC General

A lo largo de este capítulo se va a desarrollar un controlador predictivo lineal discretizado en el tiempo. Sin embargo, el modelo de avión expuesto en el capítulo anterior (2) se trataba de un modelo no lineal de seis grados de libertad constituido por un conjunto de 12 ecuaciones (2.6), (2.7), (2.8), (2.9), (2.10), (2.11), (2.13) y (2.16), el cual permitía calcular las 12 variables de estado del problema de actitud (estando el problema de la trayectoria desacoplado del resto).

Este hecho, unido a que, a pesar de las simplificaciones que se aplicaron para la obtención del conjunto de ecuaciones anterior, no existe una solución analítica para el mismo (sistema no lineal acoplado), hace necesaria una linealización del modelo de avión previa a la formulación del controlador.

3.2.1.1 Linealización del modelo

Como puede recogerse en las referencias [2] y [6], para el estudio de los problemas de estabilidad y control, es necesario seleccionar un punto de operación del avión en torno al cual se linealizarán las ecuaciones del movimiento. De esta manera se obtiene un sistema de ecuaciones diferenciales lineales con el que se pueden abordar los mencionados problemas, teniendo en cuenta que

- esta aproximación sólo será válida si el movimiento de la aeronave consiste en pequeñas perturbaciones respecto del punto de operación,
- todas las conclusiones que se obtengan tendrán carácter local,
- en la práctica se obtienen buenos resultados.

A continuación, con el fin de adaptar la notación del problema al concepto de punto de operación definido, se expresan las variables de estado como suma del valor en el punto de operación de la misma más su desviación respecto de dicho punto (advértase el abuso de notación realizado).

$$\begin{aligned}
 u &= u_s + u(t) & p &= p_s + p(t) & \phi &= \phi_s + \phi(t) \\
 v &= v_s + v(t) & q &= q_s + q(t) & \theta &= \theta_s + \theta(t) \\
 w &= w_s + w(t) & r &= r_s + r(t) & \psi &= \psi_s + \psi(t) \\
 X &= X_s + \Delta X & Y &= Y_s + \Delta Y & Z &= Z_s + \Delta Z \\
 L &= L_s + \Delta L & M &= M_s + \Delta M & N &= N_s + \Delta N
 \end{aligned}$$

Con esta nueva formulación y teniendo en cuenta las siguientes hipótesis,

- El punto de operación satisface las ecuaciones.
- Se desprecian los términos de segundo orden en las perturbaciones respecto del punto de operación.

- Los términos no lineales que puedan aparecer en las ecuaciones se aproximan por series de Taylor entorno al punto de operación, quedándose sólo con la parte lineal, esto es

$$\sin(\phi_s + \phi(t)) \approx \sin \phi_s + \phi(t) \cos \phi_s$$

$$\cos(\phi_s + \phi(t)) \approx \cos \phi_s - \phi(t) \sin \phi_s$$

las ecuaciones del movimiento linealizadas quedan como sigue

$$\Delta X - mg\theta \cos \theta_s = m(\dot{u} - rv_s - vr_s + qw_s + wq_s) \quad (3.1)$$

$$\Delta Y - mg(\phi \cos \theta_s \cos \phi_s - \theta \sin \theta_s \sin \phi_s) = m(\dot{v} + ru_s + ur_s - pw_s - wp_s) \quad (3.2)$$

$$\Delta Z - mg(\phi \cos \theta_s \sin \phi_s - \theta \sin \theta_s \cos \phi_s) = m(\dot{w} - qu_s - uq_s + pv_s + vp_s) \quad (3.3)$$

$$\Delta L = I_x \dot{p} - I_{xz} \dot{z} + (I_z - I_y)(qp_s + pq_s) + I_{xz}(qr_s + rq_s) \quad (3.4)$$

$$\Delta M = I_y \dot{q} + (I_x - I_z)(pr_s + rp_s) - 2I_{xz}(qr_s + rq_s) \quad (3.5)$$

$$\Delta N = I_z \dot{r} - I_{xz} \dot{p} - (I_z - I_y)(qr_s + rq_s) - I_{xz}(qp_s + pq_s) \quad (3.6)$$

$$\dot{\phi} = p + (q \sin \phi_s + r \cos \phi_s + \phi \dot{\theta}_s) \tan \theta_s + \theta \dot{\psi}_s \sec \theta_s \quad (3.7)$$

$$\dot{\theta} = q \cos \phi_s - r \sin \phi_s - \phi \cos \theta_s \dot{\psi}_s \quad (3.8)$$

$$\dot{\psi} = (q \sin \theta_s + r \cos \theta_s + \phi \dot{\theta}_s + \theta \sin \theta_s \dot{\psi}_s) \sec \theta_s \quad (3.9)$$

Considerando ahora un vuelo de referencia con las siguientes características,

- Simétrico: $v_s = 0$
- Rectilíneo: $\phi_s = 0$
- Estacionario: $u_s, v_s, w_s, p_s, q_s, r_s$ constantes
- Velocidad angular nula $p_s, q_s, r_s = 0 \Rightarrow \dot{\phi}_s, \dot{\theta}_s, \dot{\psi}_s = 0$

las ecuaciones anteriores se simplifican, obteniéndose las siguientes expresiones

$$\Delta X - mg\theta \cos \theta_s = m(\dot{u} + qw_s) \quad (3.10)$$

$$\Delta Y - mg\phi \cos \theta_s = m(\dot{v} + ru_s - pw_s) \quad (3.11)$$

$$\Delta Z - mg\theta \sin \theta_s = m(\dot{w} - qu_s) \quad (3.12)$$

$$\Delta L = I_x \dot{p} - I_{xz} \dot{z} \quad (3.13)$$

$$\Delta M = I_y \dot{q} \quad (3.14)$$

$$\Delta N = I_z \dot{r} - I_{xz} \dot{p} \quad (3.15)$$

$$\dot{\phi} = p + r \tan \theta_s \quad (3.16)$$

$$\dot{\theta} = q \quad (3.17)$$

$$\psi = r \sec \theta_s \quad (3.18)$$

Formulando el problema en ejes estabilidad (apéndice A de la tesis [1]), las ecuaciones (3.10), (3.11) y (3.12) correspondientes a las fuerzas se reducen a

$$\Delta X - mg\theta \cos \theta_s = m\dot{u} \quad (3.19)$$

$$\Delta Y + mg\phi \cos \theta_s = m(\dot{v} + ru_s) \quad (3.20)$$

$$\Delta Z - mg \sin \theta_s = m(\dot{w} - qu_s) \quad (3.21)$$

Como se puede apreciar en el conjunto de ecuaciones anteriores (3.19), (3.20), (3.21), (3.13), (3.14), (3.15), (3.16), (3.17) y (3.18), la formulación está realizada en términos de incrementos. Sin embargo, no se ha mencionado el valor que deben tener las fuerzas y momentos de referencia. Dichos valores pueden ser consultados en la referencia [2] en el apartado de “Ecuaciones del estado de referencia”.

Finalmente, para cerrar el problema se necesita un modelo lineal de fuerzas y momentos aerodinámicos y propulsivos. Estos modelos son funcionales de las variables de estados, esto es, que no sólo dependen de ellas en el instante actual, sino también en su historia pasada. No obstante, debido a la complejidad que induce este planteamiento, generalmente, se propone un modelo linealizado considerando únicamente los términos relevantes hasta la primera derivada; este modelo se puede simplificar aún más con la introducción de una serie de hipótesis generales y las condiciones de vuelo de referencia. No obstante, el desarrollo del mismo se escapa al alcance de este trabajo y puede ser consultado en la referencia [2].

Una vez formulado e introducido el modelo mencionado en las ecuaciones del movimiento se produce un desacople entre la dinámica longitudinal y la dinámica lateral-direccional, consecuencia de las hipótesis formuladas para el modelo aerodinámico y propulsivo. Así las ecuaciones resultantes escritas de manera matricial son

▪ Ecuaciones longitudinales

$$\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m - Z_{\dot{w}} & 0 & 0 \\ 0 & -M_{\dot{w}} & I_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -mg \cos \theta_s \\ Z_u & Z_w & Z_q + mu_s & -mg \sin \theta_s \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (3.22)$$

▪ **Ecuaciones laterales-direccionales**

$$\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_x & -I_{xz} & 0 \\ 0 & -I_{xz} & I_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_v & Y_p & Y_r - mu_s & mg \cos \theta_s \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & \tan \theta_s & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & Y_{\delta_r} \\ 0 & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3.23)$$

$$\dot{\psi} = r \sec \theta_s$$

Por último, al igual que se llevó a cabo en la sección 2.1.1.3, se va a proceder a adimensionalizar las ecuaciones linealizadas obtenidas con el objetivo de simplificar el problema, facilitar el modelado del mismo y trabajar con coeficientes adimensionales (en este caso se denominan derivadas de estabilidad de la aeronave), de los cuales se tiene un mayor conocimiento.

Para llevar a cabo la adimensionalización se requiere una diferenciación entre el problema longitudinal y el lateral-direccional puesto que las magnitudes características son distintas en cada uno de ellos.

▪ **Problema longitudinal**

- *Magnitudes características:*

$$\text{Longitud: } \frac{c}{2} \quad \text{Superficie: } S \quad \text{Velocidad: } u_s \quad \text{Masa: } \frac{1}{2} \rho S c \quad \text{Tiempo: } \frac{c}{2u_s}$$

$$\text{Presión: } \frac{1}{2} \rho V^2 \quad \text{Fuerza: } \frac{1}{2} \rho V^2 S \quad \text{Momento: } \frac{1}{2} \rho V^2 S c \quad \text{M. Inercia: } \rho S \left(\frac{c}{2}\right)^3$$

- *Variables adimensionales:*

$$C_x = \frac{X}{\left(\frac{1}{2}\right) \rho V^2 S} \quad C_z = \frac{Z}{\frac{1}{2} \rho V^2 S} \quad C_m = \frac{M}{\frac{1}{2} \rho V^2 S c}$$

$$\hat{u} = \frac{u}{u_s} \quad w = \frac{w}{u_s} \equiv \alpha \quad \hat{q} = \frac{qc}{2u_s}$$

$$\hat{t} = \frac{t2u_s}{c} \quad \mu = \frac{2m}{\rho S c} \quad \hat{I}_y = \frac{I_y}{\rho S \left(\frac{c}{2}\right)^3}$$

- *Ecuaciones adimensionalizadas:*

$$\begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & 2\mu - C_{z\hat{\alpha}} & 0 & 0 \\ 0 & -C_{m\hat{\alpha}} & \hat{I}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{\alpha} \\ \hat{q} \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} 2C_{X_s} + C_{X\hat{u}} & C_{X\alpha} & 0 & C_{Z_s} \\ 2C_{Z_s} + C_{Z\hat{u}} & C_{Z\alpha} & 2\mu + C_{Z\hat{q}} & -C_{X_s} \\ C_{m\hat{u}} & C_{m\alpha} & C_{m\hat{q}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & C_{X\delta_t} \\ C_{Z\delta_e} & 0 \\ C_{M\delta_e} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (3.24)$$

▪ **Problema lateral-direccional**

- *Magnitudes características:*

$$\begin{aligned} \text{Longitud: } \frac{b}{2} \quad \text{Superficie: } S \quad \text{Velocidad: } u_s \quad \text{Masa: } \frac{1}{2} \rho S b \quad \text{Tiempo: } \frac{b}{2u_s} \\ \text{Presión: } \frac{1}{2} \rho V^2 \quad \text{Fuerza: } \frac{1}{2} \rho V^2 S \quad \text{Momento: } \frac{1}{2} \rho V^2 S b \quad \text{M. Inercia: } \rho S \left(\frac{b}{2}\right)^3 \end{aligned}$$

- *Variables adimensionales:*

$$\begin{aligned} C_Y &= \frac{Y}{\left(\frac{1}{2}\right) \rho V^2 S} & C_l &= \frac{L}{\frac{1}{2} \rho V^2 S b} & C_n &= \frac{N}{\frac{1}{2} \rho V^2 S b} \\ \hat{v} &= \frac{v}{u_s} \equiv \beta & \hat{p} &= \frac{pb}{2u_s} \equiv \alpha & \hat{r} &= \frac{rb}{2u_s} \\ \hat{t} &= \frac{t2u_s}{b} & \mu &= \frac{2m}{\rho S b} & \hat{I}_y &= \frac{I_y}{\rho S \left(\frac{b}{2}\right)^3} \end{aligned}$$

- *Ecuaciones adimensionalizadas*

$$\begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & \hat{I}_x & -\hat{I}_{xz} & 0 \\ 0 & -\hat{I}_{xz} & \hat{I}_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{p} \\ \hat{r} \\ \hat{\phi} \end{bmatrix} = \begin{bmatrix} C_{Y\beta} & C_{Yp} & C_{Yr} - 2\mu & C_{Zs} \\ C_{l\beta} & C_{lp} & C_{lr} & 0 \\ C_{n\beta} & C_{np} & C_{nr} & 0 \\ 0 & 1 & \tan \theta_s & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & C_{Y\delta_r} \\ C_{l\delta_a} & C_{l\delta_r} \\ C_{n\delta_a} & C_{n\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3.25)$$

$$\dot{\psi} = \hat{r} \sec \theta_s \quad (3.26)$$

Nótese que en la formulación adimensional hablar de \hat{w} (velocidad adimensional en el eje z) o de α (ángulo de ataque) es exactamente lo mismo, por lo que, de ahora en adelante, se utilizarán cualquiera de las dos formulaciones. Lo mismo ocurre para el caso de \hat{v} y de β .

Finalmente, una vez realizado todo el desarrollo anterior, las ecuaciones (3.24), (3.25) y (3.26) representan un modelo de avión lineal y adimensionalizado de 6 grados de libertad, que quedaría totalmente definido una vez determinadas todas las derivadas de estabilidad que intervienen en el sistema. Para su cálculo, se ha hecho uso del modelo de simulación original facilitado por el autor de la tesis en la que se apoya este trabajo, obteniéndose los siguientes valores para las mismas.

- *Derivadas de estabilidad del problema lateral-direccional.*

Tabla 3-1 Valores de las derivadas de estabilidad del problema lateral-direccional.

-	C_Y	C_l	C_n
β	-0.5829	-0.1035	0.2021
\hat{p}	-0.1136	-0.4638	-0.0326
\hat{r}	0.4431	0.0716	-0.2194
δ_a	-	0.1731	-0.0073
δ_r	-0.3134	-0.0361	0.1576

- o *Estado de referencia adimensional.*

$$C_{Z_{sLD}} = -\frac{gb}{u_s^2} \mu \cos \theta_s = -0.6211$$

- *Derivadas de estabilidad del problema longitudinal.*

Tabla 3-2 Valores de las derivadas de estabilidad del problema longitudinal.

-	C_X	C_Z	C_m
\hat{u}	-0,0888	-0.050	0.05
α	-0.172	-6.44	-2.58
\hat{q}	-	-10.22	-51.9
δ_e	-	-0.207	-1.069
δ_t	0.3382	-	-
$\dot{\alpha}$	-	-1.4567	-9.922

- o *Estado de referencia adimensional*

$$C_{X_{sLong}} = \frac{gc}{u_s^2} \mu \sin \theta_s = 0$$

$$C_{Z_{sLong}} = -\frac{gc}{u_s^2} \mu \cos \theta_s = -0.6211$$

3.2.1.2 Formulación del MPC general

Una ley de control predictivo está formada por tres componentes básicos que son

- el modelo de predicción,
- el modelo de optimización y,
- el horizonte de predicción e intervalo de muestreo.

En base a esto, se comenzará el apartado con el planteamiento del modelo de predicción utilizado por el controlador; para lo cual hay que tener en cuenta que, como ya se indicó al comienzo de la sección, el modelo del MPC implementado en este trabajo es lineal y discretizado en el tiempo.

Teniendo presente este apunte se asumirá que la dinámica de la aeronave, la cual viene dada por la ecuación diferencial $M\dot{x} = Ax + bu$, puede ser descrita por una ecuación diferencial discretizada en el tiempo como sigue

$$Mx_{k+1} = Ax_k + Bu_k \quad (3.27)$$

donde el subíndice k hace referencia al instante de tiempo actual $x_k = x(kT_s)$, siendo T_s el intervalo de muestreo del sistema; x_k es el vector de variables de estados en el instante k ; y u_k es el vector de señales de control en el instante actual.

En lo que sigue, para simplificar la notación del problema se va a despejar la variable a determinar en la ecuación anterior y se va a suponer que la ecuación que describe el sistema es la siguiente

$$x_{k+1} = A'x_k + B'u_k \quad (3.28)$$

siendo $A' = M^{-1}A$ y $B' = M^{-1}B$, que, por simplicidad de los desarrollos, van a pasar a denotarse como A y B .

La ecuación (3.28) permite, conociendo las variables de estado de la aeronave y las señales de control en el instante de tiempo actual, determinar la respuesta futura de la misma. En este sentido, particularizando la ecuación para el instante de tiempo $k + 2$ y sustituyendo donde corresponda la expresión ya obtenida para x_{k+1} , seríamos capaces de predecir la respuesta que presentará la aeronave dentro de dos períodos en función del estado actual del sistema y de las señales de control en los instantes actual y el futuro inmediato, esto es

$$x_{k+1} = Ax_k + Bu_k \quad (3.29)$$

$$x_{k+2} = Ax_{k+1} + Bu_{k+1} = A^2x_k + ABu_k + Bu_{k+1} \quad (3.30)$$

De este modo, dada una secuencia de entradas predichas las correspondientes predicciones del estado de la aeronave son generadas mediante la simulación del modelo discretizado (3.28) a lo largo del horizonte de predicción. Definiendo un horizonte de N intervalos de muestreo se obtendrá el siguiente sistema matricial.

$$x_s = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ x_{k+3} \\ \vdots \\ x_{k+N} \end{bmatrix}_{N \times n} = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}_{N \times n, n} x_k + \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & \vdots \\ A^2B & AB & B & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ A^{N-1}B & \dots & A^2B & AB & B \end{bmatrix}_{N \times m, N \times n} \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N-1} \end{bmatrix} \quad (3.31)$$

siendo n el número de variables de estado del problema y m el número de señales de control del mismo.

Introduciendo una nueva notación con las matrices F y G , la ecuación anterior queda de la siguiente manera

$$x_s = Fx_k + Gu_s \quad (3.32)$$

donde la variable x_k representa el estado actual de la aeronave, el cual viene determinado por las mediciones de los sensores de la misma y supone la condición inicial del problema de optimización, u_s es el vector de señales de control a determinar mediante el problema de optimización, y F y G son las siguientes matrices.

$$F = \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^N \end{bmatrix}_{N \times n, n}, \quad G = \begin{bmatrix} B & 0 & 0 & \dots & 0 \\ AB & B & 0 & \dots & \vdots \\ A^2B & AB & B & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ A^{N-1}B & \dots & A^2B & AB & B \end{bmatrix}_{N \times m, N \times n} \quad (3.33)$$

El sistema obtenido (ecuación (3.32)) permite predecir el conjunto de estados futuros de la aeronave en base a un modelo de la misma (matrices A y B) y, por lo tanto, constituye el modelo de predicción del MPC para la aeronave en cuestión.

Con la definición del modelo y el horizonte de predicción, la formulación del MPC queda cerrada, a expensas de definir una ley de retroalimentación del control predictivo. Dicha ley es computada mediante la minimización de una función de coste J definida en función de las secuencias de predicción, esto es $J = J(x(k), u(k))$

Para este trabajo, se ha definido una función de coste cuadrática, para la cual el coste de predicción a minimizar queda determinado por la siguiente expresión

$$J(k) = \sum_{k=1}^N \left[(x_k - x_{ref})^T Q (x_k - x_{ref}) \right] + \sum_{i=0}^{N-1} u_k^T R u_k \quad (3.34)$$

donde las matrices Q y R son definidas positivas y su ponderación depende de las variables de estado a las que se refieran. A consecuencia de ello, el desarrollo de las mismas será llevado a cabo en cada uno de los apartados correspondientes (3.2.1.2.1 y 3.2.2.2).

En base a esta función de coste obtenida, el problema de optimización sin restricciones a resolver vendría dado por la siguiente formulación.

$$\text{Minimizar: } J(x(k), u(k)) = \sum_{k=1}^N \left[(x_k - x_{ref})^T Q (x_k - x_{ref}) \right] + \sum_{i=0}^{N-1} u_k^T R u_k \quad (3.35)$$

Sujeto a:

$$\begin{aligned} x_{k+1} &= A x_k + B u_k \\ x_k &\in \mathbb{R}^n \\ u_k &\in \mathbb{R}^m \end{aligned} \quad (3.36)$$

Esta formulación puede ser expresada en términos generales mediante la ecuación (3.32) como sigue

$$\text{Minimizar: } J(x, u) = (x_s - x_{ref})^T Q (x_s - x_{ref}) + u_s^T R u_s^T \quad (3.37)$$

Sujeto a:

$$\begin{aligned} x_s &= F x_k + G u_s \\ x_s &\in \mathbb{R}^{n \times N} \\ u_s &\in \mathbb{R}^{m \times N} \\ x_k &\in \mathbb{R}^n \end{aligned} \quad (3.38)$$

siendo ahora x_{ref} un vector de referencias futuras de tamaño igual al número de variables del problema por el número de puntos del horizonte de predicción ($n \times N$)

En lo que resta de capítulo se va a proceder al desarrollo y resolución del problema de optimización mediante el cual se obtendrá la secuencia de señales de control óptima (u_s^*) que minimiza la función de coste cuadrático.

Introduciendo la ecuación (3.32) en la expresión general de la función de coste a minimizar se obtiene la siguiente expresión

$$J(u) = (F x_k + G u_s - x_{ref})^T Q (F x_k + G u_s - x_{ref}) + u_s^T R u_s \quad (3.39)$$

Realizando las operaciones correspondientes, la expresión anterior pasa a ser

$$\begin{aligned} J(x, u) &= x_k^T F^T Q F x_k + x_k^T F^T Q G u_s - x_k^T F^T Q x_{ref} + u_s^T G^T Q F x_k + u_s^T G^T Q G u_s \\ &\quad - u_s^T G^T Q x_{ref} - x_{ref}^T Q F x_k - x_{ref}^T Q G u_s + x_{ref}^T Q x_{ref} + u_s^T R u_s \end{aligned} \quad (3.40)$$

Por último, cancelando términos y agrupando se consigue la siguiente ecuación

$$J(u) = u_s^T (G^T Q G + R) u_s + 2(F x_k - x_{ref})^T Q G u_s + cte \quad (3.41)$$

donde todos los términos que no dependen de u_s han sido agrupados en una constante, ya que, a efectos prácticos del problema de optimización, no aportan ningún valor. Dicho problema queda formulado con la nueva expresión de la siguiente manera

$$\text{Minimizar: } J(u) = u_s^T (G^T QG + R) u_s + 2(Fx_k - x_{ref})^T QG u_s + cte \quad (3.42)$$

Sujeto a:

$$\begin{aligned} u_s &\in \mathbb{R}^{m \times N} \\ x_k &\in \mathbb{R}^n \end{aligned} \quad (3.43)$$

Derivando la función de coste respecto de la variable a optimizar.

$$\nabla J(u) = 2(G^T QG + R) u_s + 2(Fx_k - x_{ref})^T QG \quad (3.44)$$

Introduciendo la definición de las matrices H y f , expuestas a continuación, e igualando a cero la expresión

$$H = 2(G^T QG + R), \quad f = 2(Fx_k - x_{ref})^T QG \quad (3.45)$$

$$\nabla J(u) = H u_s + f = 0 \quad (3.46)$$

la solución al problema de optimización (u_s^*) se obtendría despejando dicha variable en la ecuación anterior

$$u_s^* = -H^{-1} f \quad (3.47)$$

Esta expresión permite el cálculo del vector de señales de control de la aeronave que minimiza la función de coste cuadrática definida teniendo en cuenta el modelo de avión utilizado (las matrices H y f dependen de las matrices F y G , las cuales se conforman con las matrices A y B que provienen del modelo lineal adimensionalizado de seis grados de libertad del avión).

Por último, para dar por cerrada la formulación se hace necesario realizar un apunte. Inicialmente, en base a la formulación lineal del modelo de avión, se asumió que la dinámica de la aeronave vendría dada por una ecuación diferencial discretizada en el tiempo, ecuación (3.27). En dicha ecuación se produjo un abuso de notación a la hora de definir la variable x_{k+1} , ya que el resultado de discretizar la ecuación diferencial original sería el siguiente,

$$M\Delta x_{k+1} = Ax_k + Bu_k \quad (3.48)$$

por lo que, realmente, cuando se ha hablado de x_{k+1} en realidad se estaba hablando del incremento que sufrían las variables de estado en el instante $k + 1$, esto es $\Delta x_{k+1} = x_{k+1} - x_k$. Este abuso de notación se ha mantenido a la hora de definir los vectores x_s y u_s , los cuales en realidad quedarían como sigue

$$x_s = \begin{bmatrix} x_{k+1}' \\ x_{k+2}' \\ x_{k+3}' \\ \vdots \\ x_{k+N}' \end{bmatrix} = \begin{bmatrix} x_{k+1} - x_k \\ x_{k+2} - x_{k+1} \\ x_{k+3} - x_{k+2} \\ \vdots \\ x_{k+N} - x_{k+N-1} \end{bmatrix}, \quad u_s = \begin{bmatrix} u_{k_0} \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N-1} \end{bmatrix} = \begin{bmatrix} u_{k_0} - u_{k-1} \\ u_{k+1} - u_{k_0} \\ u_{k+2} - u_{k+1} \\ \vdots \\ u_{k+N-1} - u_{k+N-2} \end{bmatrix} \quad (3.49)$$

En base a esto, se puede afirmar que se ha realizado una formulación del problema en términos incrementales. No obstante, a la hora de generar el modelo Simulink es mucho más sencillo tratar el problema de optimización en términos absolutos de las variables, por lo que se hace necesaria una redefinición del modelo propuesto.

De esta forma, se va a proceder a continuación a realizar la mencionada modificación. Para ello, hay que tener presente lo expuesto en las ecuaciones (3.48) y (3.49) y que, a partir de ahora, x_k y u_k van a designar los valores absolutos de las variables, mientras que se utilizará el término Δ para designar los incrementos de las variables.

Se comenzará definiendo la matriz A_u que va a permitir modificar la formulación del problema sin la necesidad de introducir muchos cambios en el desarrollo ya realizado.

$$A_u = \begin{bmatrix} I_{d_{mxm}} & 0 & \dots & \dots & 0 \\ -I_{d_{mxm}} & I_{d_{mxm}} & 0 & \dots & \vdots \\ 0 & -I_{d_{mxm}} & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -I_{d_{mxm}} & I_{d_{mxm}} \end{bmatrix}_{Nxm, Nxm} \quad (3.50)$$

Esta matriz es cuadrada de tamaño el número de variables de control del problema por el número de puntos del horizonte de predicción y definida por bloques. Además, es propia de cada problema (longitudinal y lateral-direccional); sin embargo, dado que el número de variables de control de ambos problemas es el mismo, no existe una distinción entre $A_{u_{Long}}$ y $A_{u_{LD}}$, razón por la que se define en este apartado.

Definiendo ahora el vector u_{s-1} como sigue,

$$u_{s-1} = \begin{bmatrix} u_{k-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{N \times m} \quad (3.51)$$

la matriz A_u implicaría lo siguiente

$$\Delta u_s = A_u u_s - u_{s-1} \quad (3.52)$$

Una vez queda completamente definida la matriz A_u , su introducción en la formulación del problema se hace mediante una redefinición de la matriz H y la adición de un término en la ecuación que define la matriz f , esto es,

$$H^* = 2(G^T QG + A_u^T R A_u), \quad f^* = 2(Fx_k - x_{ref})^T QG - 2u_{s-1} R A_u \quad (3.53)$$

siendo u_{s-1} el vector ya definido con anterioridad y x_k el valor absoluto de las variables de estado en el instante k .

De esta forma la solución final del problema de optimización sin restricciones es

$$u_s^{opt} = -(H^*)^{-1} f^* \quad (3.54)$$

siendo $u_s^{opt} = u_s^*$, pero al que se le ha cambiado la notación para distinguirlo del superíndice asignado a las matrices H^* y f^* modificadas tras la corrección aplicada.

De esta forma, con este simple cambio, se permite pasar de una formulación en términos incrementales a un modelo definido en términos absolutos de las variables de estado y de las señales de control de la aeronave.

Por último, es necesario destacar dos aspectos importantes. Primeramente, hay que tener en cuenta que toda la resolución del problema de optimización se ha llevado a cabo sin imponer ninguna restricción extra a la del cumplimiento del modelo. En ese sentido, el modelo del MPC no recoge las saturaciones que puedan aparecer en las variables de control de la aeronave, lo cual, debido a las razones que serán explicadas posteriormente, puede derivar en un comportamiento no deseado. Es por ello que se decide introducir una corrección en la formulación del controlador de manera que tenga en cuenta dichas saturaciones.

Seguidamente, tras las primeras simulaciones se detecta que la solución del controlador presenta un error constante en régimen permanente que provoca que el estado de la aeronave nunca llegue a coincidir con la referencia deseada. En consecuencia, se decide introducir una corrección que solucione esta discrepancia.

De esta manera, en los dos sub-apartados siguientes se exponen todos los desarrollos y explicaciones relativas a la implementación de ambas modificaciones.

3.2.1.2.1 Restricciones en el problema de optimización (saturaciones)

Como ya se ha mencionado previamente, en este capítulo se ha definido un problema de optimización sin restricciones que permite obtener una solución óptima del vector de señales de control, es decir, proporciona un resultado que minimiza la función de coste cuadrática J con la única restricción de cumplir el modelo de predicción. Sin embargo, el hecho de no presentar ningún tipo de

restricción puede incidir en la aparición de comportamientos no deseados en el controlador. La razón de ello reside en que el controlador calcula una solución y cree que se está poniendo en práctica; sin embargo, puede darse el caso de que una de estas señales no pueda ser implementada por el avión porque alcance el valor de saturación. A consecuencia de ello se produce un desfase entre lo que ocurre en la realidad y lo que el controlador piensa que está ocurriendo, pudiéndose originar sobre-oscilaciones, fallos puntuales en el controlador e incluso entrada en inestabilidad del sistema. Por todo ello, una de las modificaciones incluidas al modelo de MPC diseñado, la cual va a ser expuesta a continuación, consiste en la formulación de un problema aproximado de optimización con restricciones, que permite la inclusión de las saturaciones de las señales de control en el modelo y que, como se podrá observar en el apartado de conclusiones, mejora el comportamiento del mismo ante situaciones límites sin incurrir en tiempos de cálculo muy grandes, hecho éste que representa un requisito indispensable en un avión no tripulado ya que los procesadores embarcados tienen limitaciones en las capacidades de cálculo y, dado que el tiempo de muestreo es de 20 milisegundos, los procesos ejecutados en el controlador no deben exceder ese tiempo.

En ese sentido, se decide reformular el modelo de controlador de manera que se incluyan estas saturaciones en la resolución del problema de optimización. Así, el nuevo problema a resolver sería el siguiente

$$\text{Minimizar: } J(u) = u_s^T (G^T Q G + R) u_s + 2(F x_k - x_{ref})^T Q G u_s + cte \quad (3.55)$$

Sujeto a:

$$\begin{aligned} u_s &\in \mathbb{R}^{m \times N} \\ x_k &\in \mathbb{R}^n \\ u_s^{opt} &\in [LB^{m \times N}, UB^{m \times N}] \end{aligned} \quad (3.56)$$

siendo LB y UB dos vectores de tamaño igual al número de variables de control del problema (m) por el horizonte de predicción (N), y que representan los valores mínimos (LB) y máximos (UB) permitidos de las señales de control.

La resolución de este problema se puede llevar a cabo mediante la función de Matlab “quadprog”; sin embargo, la ejecución de la misma lleva asociado un gasto computacional en tiempo y memoria que no se puede permitir en el sistema embarcado en el UAV.

A consecuencia de ello se decide implementar el proceso seguido por la función quadprog pero con dos diferencias fundamentales. Por un lado, el procedimiento de saturar y recalcular el problema de optimización no se realiza para todas las señales de control, sino que solo se lleva a cabo para un número limitado de saturaciones encontradas. Por otro lado, la formulación propuesta no permite, una vez saturada la variable correspondiente, encontrar el óptimo global del problema con restricciones,

sino que lo que se obtiene es un pseudo-óptimo del problema que, como se comprobará en el capítulo de resultados, ofrece una buena solución con un coste de tiempo computacional bastante inferior. Para su implementación se genera una función Matlab que, basándose en el modelo matemático del problema que se va a exponer a continuación, aplica las correcciones pertinentes al vector de señales de control. Esta función se encuentra recogida en el apéndice de este trabajo.

A continuación se muestra el diagrama de flujo que representa el proceso de aplicación de las restricciones en el sistema. Posteriormente, en base a este diagrama se desarrolla la matemática del problema.

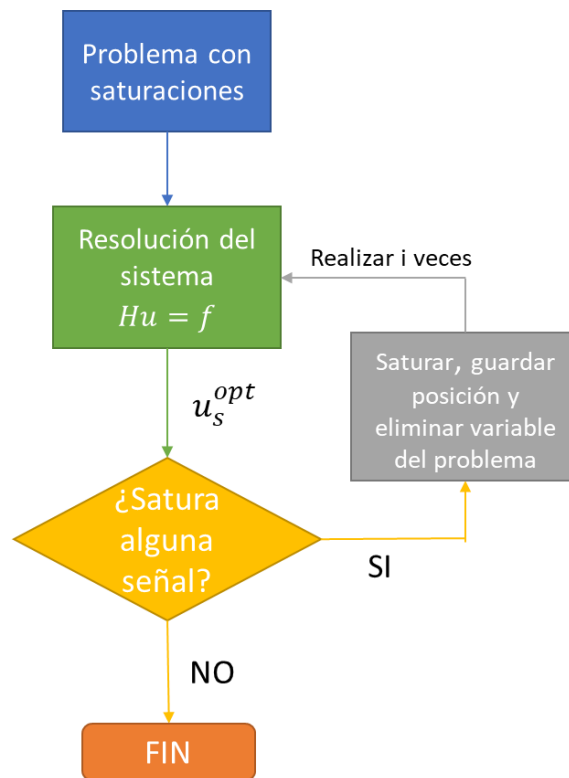


Figura 3.2 Diagrama de flujo del proceso de resolución del problema de optimización con restricciones.

Para el desarrollo matemático se va a partir de la ecuación (3.54) que determina el resultado del problema de optimización sin restricciones. Posteriormente se deberá realizar un barrido en el vector obtenido en busca de las saturaciones que hayan podido aparecer. Una vez se haya encontrado la primera saturación se deberá guardar la posición de la componente dentro del vector de señales de control y el tipo de saturación dada (límite inferior o superior), saturar la señal, eliminarla del problema, redefinir el problema de optimización y recalcular el vector de señales de control como se expone a continuación. Este proceso, tal y como se ha comentado anteriormente, podrá repetirse hasta un número máximo de veces, parámetro que vendrá impuesto por el usuario.

Supongamos que se tiene el siguiente vector de señales de control del problema longitudinal en el que la señal δ_{e_1} ha saturado por superar a $\delta_{e_{maximo}}$.

$$u_s^{opt} = \begin{bmatrix} \delta_{e_0} \\ \delta_{t_0} \\ \delta_{e_1} \\ \delta_1 \\ \vdots \\ \delta_{e_{N-1}} \\ \delta_{t_{N-1}} \end{bmatrix} \quad (3.57)$$

Sabiendo que la tercera componente del vector ha saturado, deberemos guardar su posición y tipo de saturación en la siguiente matriz

$$M_{saturacion} = [3 \quad \delta_{e_{maximo}}] \quad (3.58)$$

donde la primera columna indica la componente que ha saturado y la segunda el tipo de saturación producida.

A continuación, se debe eliminar dicha componente del problema y, en consecuencia, redefinir las matrices H y f .

$$H' = \begin{bmatrix} H_{11} & H_{12} & H_{14} & \dots & H_{1n} \\ H_{21} & H_{22} & H_{24} & \dots & \vdots \\ H_{41} & H_{42} & H_{44} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{n1} & H_{n2} & H_{n4} & \dots & H_{nn} \end{bmatrix} \quad (3.59)$$

$$f' = \begin{bmatrix} f_1 \\ f_2 \\ f_4 \\ \vdots \\ f_n \end{bmatrix} \quad (3.60)$$

donde, como puede observarse, se han eliminado la fila y la columna correspondientes a la componente saturada que, para el caso de estudio, se trata de la tercera.

Finalmente, se resuelve un nuevo problema de optimización que viene dado por la siguiente ecuación.

$$H' \begin{bmatrix} \delta_{e_0} \\ \delta_{t_0} \\ \delta_{t_1} \\ \vdots \\ \delta_{e_{n-1}} \\ \delta_{t_{n-1}} \end{bmatrix} = -f' - \begin{bmatrix} H_{13} \delta_{e_{maximo}} \\ H_{23} \delta_{e_{maximo}} \\ H_{43} \delta_{e_{maximo}} \\ \vdots \\ H_{n3} \delta_{e_{maximo}} \end{bmatrix} \quad (3.61)$$

El proceso, que ha sido desarrollado para el controlador longitudinal, es análogo al que habría que aplicar para el MPC lateral-direccional, por lo que se ha obviado su explicación.

Los resultados obtenidos en las simulaciones del modelo con las saturaciones implementadas se encuentran expuestos en el siguiente capítulo, junto a la comparativa realizada con el controlador original sin saturaciones.

3.2.1.2.2 Eliminación del error en régimen permanente

Como se comentó con anterioridad, al simular el controlador se detecta que la solución del mismo presenta un error constante en régimen permanente que provoca que el estado de la aeronave nunca llegue a coincidir con la referencia deseada. Esto puede ser comprobado en las figuras expuestas a continuación. (Figura 3.3, Figura 3.4 y Figura 3.5).

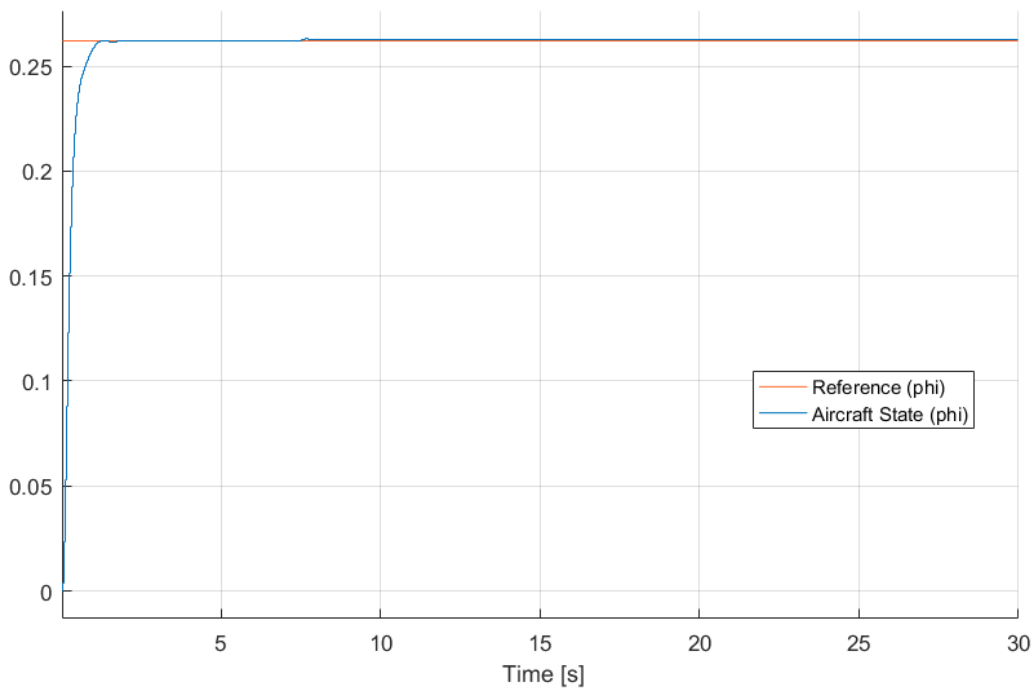
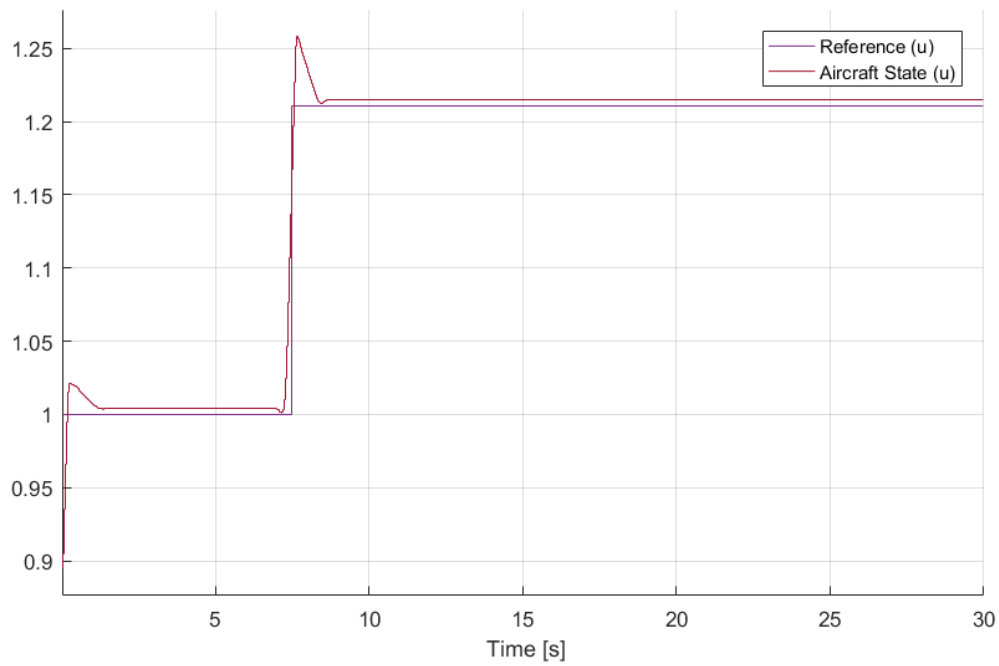
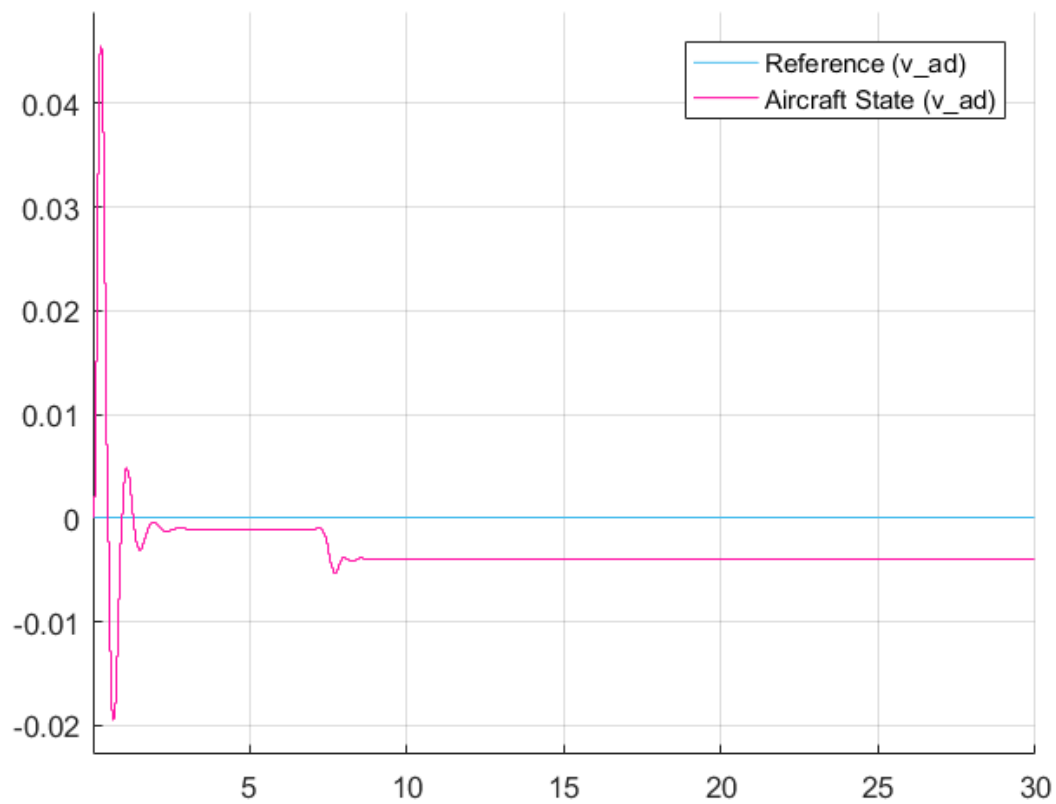


Figura 3.3 Error en régimen permanente de ϕ .

Figura 3.4 Error en régimen permanente de u .Figura 3.5 Error en régimen permanente de β .

Para corregir esta desviación se plantea una modificación que consiste en aplicar una corrección a la referencia de entrada del controlador, de manera que el MPC no recibe la referencia real, sino que recibe una señal corregida igual a la referencia original más la diferencia existente entre el estado y la referencia deseada en permanente, es decir, más el error existente en régimen permanente.

Esta corrección aplicada se corresponde matemáticamente con la integración de un PI en el controlador, el cual será introducido mediante una formulación bastante simple que solo afecta a la referencia generada, y que sigue la siguiente ecuación

$$x_{ref} = x_{ref_{real}} + K_e(x_{ref_{real}} - x_k) + K_i \int_{t_0}^{t_k} (x_{ref_{real}} - x_k) dt \quad (3.62)$$

siendo x_k las variables de estado de la aeronave en el instante actual de simulación, $x_{ref_{real}}$ la referencia deseada, x_{ref} la referencia modificada que recibe el controlador, K_e y K_i las ganancias del PI, y el término integral representa el llamado error “acumulado” y no es más que el sumatorio de la historia de los errores por el intervalo de tiempo de muestreo y por la ganancia del integrador (K_i).

Como ya ha sido comentado, la aparición del término integral implica una realimentación en el controlador de la variable error “acumulado”, la cual puede verse en la Figura 3.10.

Para su integración dentro del sistema que modela al controlador se ha optado por la generación de un código al final de las funciones `ref_generator_LD` y `ref_generator_Long`, que permite añadir la corrección a las referencias justo antes de que éstas sean enviadas al controlador. Este código puede ser consultado junto al resto de scripts en el apéndice de esta memoria.

Para finalizar, hay que mencionar que los resultados obtenidos, los cuales serán expuestos en el capítulo 4, demuestran una mejora en el comportamiento del sistema al integrar la corrección debida al error permanente. En el mencionado capítulo se procederá, en base a las gráficas obtenidas de las diversas simulaciones, a realizar un análisis más profundo sobre esta comparativa.

3.2.2 MPC Longitudinal

Recapitulando lo expuesto al comienzo de la sección, para cerrar el modelo linealizado de avión de seis grados de libertad se hizo necesario elaborar un modelo lineal de las fuerzas y momentos aerodinámicos y propulsivos. Esta elaboración se hizo en base a una serie de condiciones de referencia e hipótesis generales que pueden ser consultadas en [2]. Con la introducción de dicho modelo de fuerzas y momentos en el modelo de avión se obtuvieron unas ecuaciones que presentaban un desacople entre la dinámica lateral-direccional y la longitudinal. Este resultado supone una gran ventaja a la hora de desarrollar el modelo del MPC, ya que permite tratar ambos problemas

independientemente. En consecuencia, el sistema de control de actitud desarrollado se subdividirá en dos controladores que resolverán ambos problemas de actitud de manera individual. Ello explica la diferenciación realizada en esta memoria a la hora de presentar la formulación de los modelos de MPC lateral-direccional y longitudinal.

Por consiguiente, este apartado se centrará en la particularización de las ecuaciones anteriores para el caso de la dinámica longitudinal y en la definición de las matrices dependientes de las variables del problema tratado.

Para finalizar, con el fin de plantear una futura integración del sistema de control de actitud con el sistema de guiado desarrollado en la tesis doctoral [1], se introduce un apartado en el que se procede a modificar las variables de estado que definen el modelo.

3.2.2.1 Formulación

Para comenzar con la formulación del modelo es necesario saber cuáles son las variables de estado y de control que intervienen en la dinámica longitudinal. Estas variables han sido agrupadas en dos vectores de la siguiente forma

$$x_{Long} = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix}, \quad u_{Long} = \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix}$$

donde u y w son las velocidades aerodinámicas de la aeronave en los ejes x y z en el sistema de referencia de ejes cuerpo, q representa la velocidad angular de cabeceo y θ el valor del ángulo de cabeceo. Por su parte, como se ha mencionado anteriormente, δ_e y δ_t son los valores de las señales de control correspondientes a la deflexión del timón de profundidad y a la posición de la palanca de gases de la aeronave, respectivamente. Nótese el abuso de notación realizado a la hora de eliminar el símbolo “^” que indica que las variables se encuentran adimensionalizadas.

Una vez conocidas las variables que intervienen en el sistema se está en disposición de concretar todas las matrices que intervienen en el problema de simulación y que dependen directa o indirectamente de dichas variables.

Partiendo de la ecuación diferencial adimensional que describe al sistema

$$\begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & 2\mu - C_{z\hat{\alpha}} & 0 & 0 \\ 0 & -C_{m\hat{\alpha}} & \hat{I}_{\hat{y}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{\alpha} \\ \hat{q} \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} 2C_{x_s} + C_{x_{\hat{u}}} & C_{x_{\alpha}} & 0 & C_{z_s} \\ 2C_{z_s} + C_{z_{\hat{u}}} & C_{z_{\alpha}} & 2\mu + C_{z_{\hat{q}}} & -C_{x_s} \\ C_{m_{\hat{u}}} & C_{m_{\alpha}} & C_{m_{\hat{q}}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0 & C_{x_{\delta_t}} \\ C_{z_{\delta_e}} & 0 \\ C_{M_{\delta_e}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \quad (3.63)$$

y haciendo un símil con el modelo dinámico propuesto al comienzo de la sección 3.2.1.2 se pueden identificar claramente las matrices M , A y B , de manera que

$$M_{Long} = \begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & 2\mu - C_{z\hat{\alpha}} & 0 & 0 \\ 0 & -C_{m\hat{\alpha}} & \hat{I}_{\hat{y}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.64)$$

$$A_{Long} = \begin{bmatrix} 2C_{X_s} + C_{X_{\hat{u}}} & C_{X_{\alpha}} & 0 & C_{Z_s} \\ 2C_{Z_s} + C_{Z_{\hat{u}}} & C_{Z_{\alpha}} & 2\mu + C_{Z_{\hat{q}}} & -C_{X_s} \\ C_{m_{\hat{u}}} & C_{m_{\alpha}} & C_{m_{\hat{q}}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (3.65)$$

$$B_{Long} = \begin{bmatrix} 0 & C_{X_{\delta_t}} \\ C_{Z_{\delta_e}} & 0 \\ C_{M_{\delta_e}} & 0 \\ 0 & 0 \end{bmatrix} \quad (3.66)$$

donde se le ha añadido el subíndice “Long” para distinguir las matrices M , A y B de ambos problemas (longitudinal y lateral-direccional). Si se aplica el mismo cambio de notación que el realizado en la ecuación (3.28), se redefinen las matrices obteniéndose el siguiente resultado

$$A_{Long} = M_{Long}^{-1}A_{Long}, \quad B_{Long} = M_{Long}^{-1}B_{Long} \quad (3.67)$$

Con esta nueva formulación, se ha conseguido llegar a la misma notación utilizada en la formulación del modelo general ya explicado. El siguiente paso sería el montaje de las matrices F_{Long} y G_{Long} que definen el modelo de predicción del controlador $x_s = F_{Long}x_k + G_{Long}u_s$. Este proceso ha sido implementado en una función de Matlab que permite obtener conjuntamente las matrices F y G de ambos problemas en base a las matrices A y B de cada uno de ellos y el horizonte de predicción fijado. El código generado puede ser consultado en el apéndice correspondiente de esta memoria (Apéndice A. Montaje matrices **F** y **G**).

Las siguientes matrices particulares de este problema son Q y R que, como ya se introdujo, son matrices definidas positivas (Q puede ser semi-definida positiva), y representan el peso que tiene cada una de las variables de estados (Q) y de control (R) en el problema de optimización.

Ambas matrices son diagonales y, al igual que ocurría con la matriz de corrección A_u , están definidas por bloques como sigue

$$Q = \begin{bmatrix} Q_{Long} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Q_{Long} \end{bmatrix}_{N \times n, N \times n}, \quad R = \begin{bmatrix} R_{Long} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{Long} \end{bmatrix}_{N \times m, N \times m} \quad (3.68)$$

es decir, ambas son matrices cuadradas de tamaños el horizonte de predicción por el número de variables de estado del problema longitudinal, para el caso de la Q , y el horizonte de predicción por el número de variables de control del problema longitudinal, para el caso de la R .

Por su parte, Q_{Long} y R_{Long} se definen de la siguiente manera

$$Q = \begin{bmatrix} \frac{p_u}{(\Delta u)^2} & 0 & 0 & 0 \\ 0 & \frac{p_w}{(\Delta w)^2} & 0 & 0 \\ 0 & 0 & \frac{p_q}{(\Delta q)^2} & 0 \\ 0 & 0 & 0 & \frac{p_\theta}{(\Delta \theta)^2} \end{bmatrix}_{4 \times 4}, \quad R = \begin{bmatrix} p_{\delta_e} & 0 \\ 0 & p_{\delta_t} \end{bmatrix}_{2 \times 2} \quad (3.69)$$

donde $p_u, p_w, p_q, p_\theta, p_{\delta_e}$ y p_{δ_t} son los pesos asignados a cada una de las variables que intervienen en el problema, y $\Delta_u, \Delta_w, \Delta_q$ y Δ_θ representan variaciones típicas de la variables u, w, q y θ , respectivamente. La introducción de estas variaciones en la matriz Q se explica por la formulación de la función de coste cuadrática definida, ecuación (3.37). En ella la matriz Q se encuentra multiplicada de manera cuadrática por las diferencias entre el vector de variables de estado y la referencia que se desea para dichas variables, es decir, las variaciones que presentan las variables; consecuentemente, para que los pesos $p_u, p_w, p_q, p_\theta, p_{\delta_e}$ y p_{δ_t} adquieran el sentido de su definición, habría que adimensionalizar las variaciones dividiendo por el cuadrado de las variaciones típicas ya que se está trabajando con variables adimensionales en la formulación del controlador.

El ajuste de los pesos anteriores puede llegar a suponer grandes diferencias en la respuesta del controlador; en consecuencia, se ha realizado un ajuste particular de ellos para cada una de las simulaciones desarrolladas en este trabajo, intentando evitar en la medida de los posible que aparezcan sobre-oscilaciones en las mismas.

Llegados a este punto se hace necesario realizar una anotación importante; el sistema con el que se está trabajando en la presente memoria es un sistema linealizado en torno a una condición de vuelo de referencia. En base a ello, se definieron las variables del problema como la suma del valor de la misma en el punto de operación (vuelo de referencia en equilibrio) más su desviación, por lo que las soluciones obtenidas al resolver el sistema no son los valores de las variables de estado, sino sus desviaciones con respecto al vuelo de equilibrio. Todo esto implica que hay que tener un especial cuidado a la hora de trabajar con las variables, ya que el modelo de avión no trabaja con los incrementos y, en consecuencia, el estado con el que se alimenta el controlador está definido en términos globales. Así, definiendo el

vuelo de referencia como sigue,

$$\vec{u}_{sref} = [\phi \quad \theta \quad \psi \quad u \quad v \quad w \quad p \quad q \quad r] = [0 \quad 0 \quad 0 \quad 19 \text{ m/s} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \quad (3.70)$$

$$\vec{u}_{srefad} = [\phi \quad \theta \quad \psi \quad u \quad v \quad w \quad p \quad q \quad r] = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0] \quad (3.71)$$

la disconformidad quedaría resuelta restando el equilibrio longitudinal, previamente a la resolución del problema de optimización, a las variables de estado de la aeronave, las referencias generadas y las variables de control que realimentan el sistema. Una vez obtenido el vector de señales solución del problema se deberá volver a sumar el equilibrio antes de mandar la señal al modelo de avión. Este proceso también deberá ser llevado a cabo en el controlador lateral-direccional, restando en este caso el correspondiente equilibrio; sin embargo, al ser cero todas las variables del problema lateral-direccional en el equilibrio no se ha tomado ninguna acción en el MPC correspondiente.

Con la definición de las matrices anteriores y el apunte realizado, lo único que quedaría para poder formular el problema de optimización para la dinámica longitudinal es definir las matrices H_{Long}^* y f_{Long}^* , para lo cual se hace uso de la ecuación (3.53).

$$H_{Long}^* = 2(G_{Long}^T Q_{Long} G_{Long} + A_u^T R_{Long} A_u), \quad (3.72)$$

$$f^* = 2(F_{Long} x_{k_{Long}} - x_{ref_{Long}})^T Q_{Long} G_{Long} - 2u_{s-1_{Long}} R_{Long} A_u \quad (3.73)$$

Así, de la misma manera que se resolvió el problema de optimización para el caso general, la solución del problema de optimización sin restricciones longitudinal es la siguiente

$$u_{s_{Long}}^{opt} = -H_{Long}^{*-1} f_{Long}^* \quad (3.74)$$

En este caso, la solución $u_{s_{Long}}^{opt}$ representa un vector de predicciones de señales de control de $N \times n$ componentes (horizonte de predicción por número de señales de control del problema longitudinal),

$$u_{s_{Long}}^{opt} = \begin{bmatrix} \delta_{e_{k_0}} \\ \delta_{t_{k_0}} \\ \delta_{e_{k+1}} \\ \delta_{t_{k+1}} \\ \vdots \\ \delta_{t_{k+N-1}} \\ \delta_{t_{k+N-1}} \end{bmatrix} \quad (3.75)$$

que optimiza la función de coste cuadrática definida. Las primeras dos componentes de este vector (δ_{e_k} y δ_{t_k}) serán las que reciba y utilice el modelo de avión para determinar el estado futuro de la aeronave. Este nuevo estado calculado (en el caso de un sistema real el estado no sería determinado

por el modelo de avión, sino que vendría dado por la actuación de la aeronave y medido por los sensores de la misma) realimenta al controlador, junto con la referencia longitudinal y las dos componentes mencionadas, para que éste pueda volver a resolver el sistema en el instante siguiente. Es el esquema de realimentación del controlador longitudinal queda recogido en la Figura 3.6, en la que por simplicidad se han omitido los superíndices que indican solución óptima.

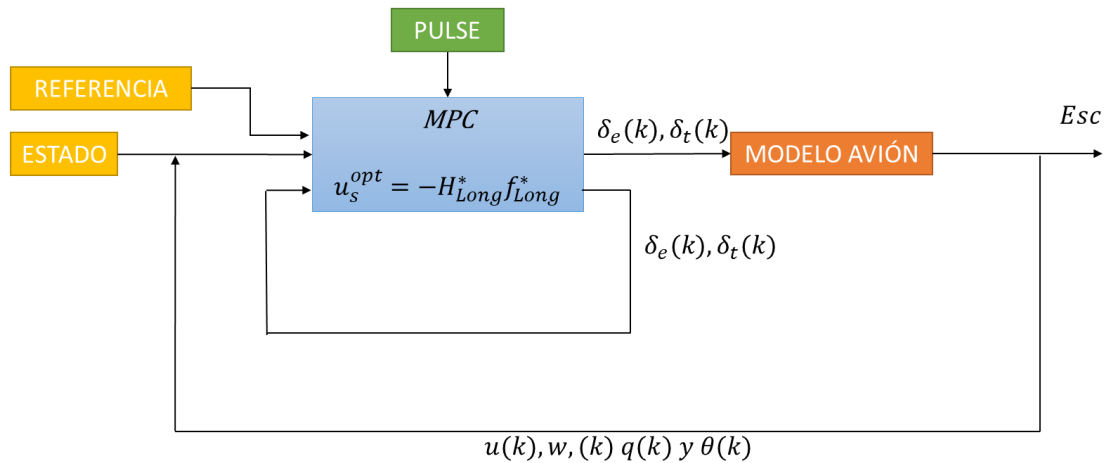


Figura 3.6. Esquema de realimentación del MPC longitudinal.

Esta solución obtenida, ecuación (3.75), se trata de la solución del problema de optimización sin restricciones. Para determinar la solución del problema con restricciones habría que aplicarle la formulación propuesta en el apartado 3.2.1.2.1.

Por último, es de necesaria mención que todo el código generado para el cálculo de matrices, montaje del problema de optimización y resolución del mismo puede ser consultado en el capítulo Apéndices de esta memoria.

3.2.2.2 Adaptación de las variables de estado a las desarrolladas en la tesis de referencia

Tal y como ha sido expuesto en los capítulos anteriores de esta disertación, las variables de estado del problema longitudinal utilizadas para definir el modelo del controlador son diferentes a las usadas por la referencia [1]. En base a ello, con el objetivo tanto de poder realizar futuras comparaciones y análisis entre ambos sistemas como de integrar el controlador en el sistema general diseñado en la mencionada tesis, se introduce una modificación en la formulación del problema que permite pasar del conjunto actual de variables de estado longitudinales (u, w, q y θ) al grupo de señales utilizado en la referencia mencionada anteriormente (u y γ). No obstante, por cuestiones de mejora en la estabilidad del avión, se decide mantener el control en la velocidad angular de cabeceo (q).

Para poder llevar a cabo la transformación explicada con anterioridad se define una matriz de cambio que permita pasar de un conjunto de variables a otro en base a las siguientes premisas

- las variables u y q se mantiene de una formulación a otra,
- el ángulo de trayectoria se define como la diferencia entre el ángulo de cabeceo θ y el de ataque α ,
- con la introducción del ángulo de trayectoria en la formulación, se elimina la capacidad de control sobre las variables que la conforman (α y θ).

De esta forma, la matriz queda como sigue

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.76)$$

$$\begin{bmatrix} u \\ \gamma \\ q \end{bmatrix} = C \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \quad (3.77)$$

donde la primera fila se corresponde con la variable u , la segunda con la nueva variable definida γ y la tercera con la aceleración angular de cabeceo q .

Una vez definida la matriz C , se procede a reformular el problema completo. Ello implica realizar una serie de modificaciones tanto en las matrices definidas en la formulación del MPC longitudinal como en el modelo de Simulink

Se comenzará con las rectificaciones del modelo de predicción. De esta manera, las matrices F , G y Q_{Long} pasan a ser las siguientes.

$$Q_{Long}' = \begin{bmatrix} \frac{p_u}{(\Delta u)^2} & 0 & 0 \\ 0 & \frac{p_\gamma}{(\Delta \gamma)^2} & 0 \\ 0 & 0 & \frac{p_q}{(\Delta q)^2} \end{bmatrix} \quad (3.78)$$

$$F' = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^N \end{bmatrix}, \quad G' = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & \vdots \\ CA^2B & CAB & CB & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ CA^{N-1}B & \dots & CA^2B & CAB & CB \end{bmatrix} \quad (3.79)$$

Por su parte, para que exista una coherencia en el controlador también deberá ser modificada la referencia que recibe éste. De esta forma, una vez aplicada la corrección del error en permanente a la señal de referencia, ésta se multiplica por bloques por la matriz C como sigue

$$x'_{ref} = \begin{bmatrix} Cx_{ref_1} \\ Cx_{ref_2} \\ \vdots \\ Cx_{ref_N} \end{bmatrix} \quad (3.80)$$

siendo x_{ref_i} las referencias de las cuatro variables de estado originales para el instante i de la predicción.

Todas estas variaciones expuestas han sido introducidas en los scripts de Matlab originales y pueden ser consultadas en los apéndices de esta memoria.

Por último, en cuanto a los cambios introducidos en el modelo Simulink, solamente ha sido necesario incluir un bloque dentro del sistema del MPC longitudinal que permita reorganizar las variables a representar (estado de la aeronave) para poder comparar la evolución de dichas variables con respecto a sus referencias. Una captura del interior del bloque se muestra en la siguiente imagen (Figura 3.7).

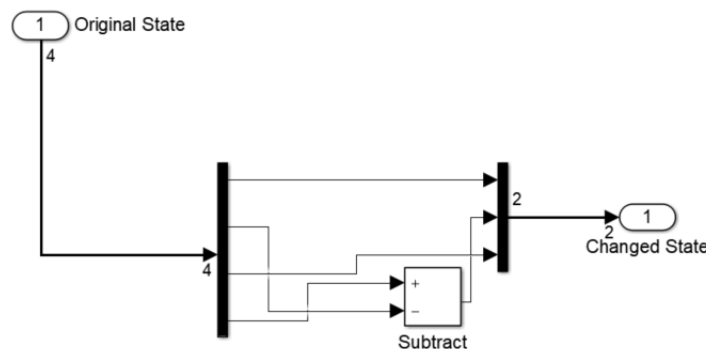


Figura 3.7 Bloque para representar las nuevas variables de estado.

3.2.3 MPC Lateral-direccional

Siguiendo la línea del proceso desarrollado en el apartado anterior, se va a proceder a la particularización del modelo general de controlador para la resolución del problema de dinámica Lateral-direccional. En ese sentido se definirán las variables del problema, las matrices que dependen de ellas, se propondrá una formulación propia del problema de optimización para la mencionada dinámica y se obtendrá la solución del mismo. Finalmente se describe el modelo Simulink utilizado para implementar el controlador.

3.2.3.1 Formulación

Inicialmente, previo al desarrollo de la formulación, es necesario conocer las variables de estado y de control que van a intervenir en el problema. Estas han sido agrupadas en el siguiente vector

$$x_{LD} = \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix}, \quad u = \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

siendo v la velocidad aerodinámica de la aeronave en el eje y_B (sistemas de ejes cuerpo), p y r son las velocidades angulares en los ejes x_B y z_B , respectivamente, y ϕ y ψ son los ángulos de balance y guiñada, respectivamente. Por su parte, como ya ha sido expuesto en otros apartados, δ_a y δ_r son los valores de las señales de control correspondientes a la deflexión de los alerones y del timón de dirección. Al igual que en caso anterior se ha realizado un abuso de notación al eliminar el símbolo “^” que indica que las variables se encontraban adimensionalizadas.

Nótese que, a pesar de encontrarse desacoplada la ecuación correspondiente a la variable de estado ψ , se ha optado por incluirla con el resto de variables. Así, las ecuaciones (3.25) y (3.26) quedan agrupadas en la siguiente expresión.

$$\begin{bmatrix} 2\mu & 0 & 0 & 0 & 0 \\ 0 & \widehat{I}_x & -\widehat{I}_{xz} & 0 & 0 \\ 0 & -\widehat{I}_{xz} & \widehat{I}_z & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\beta} \\ \widehat{p} \\ \widehat{r} \\ \widehat{\phi} \\ \widehat{\psi} \end{bmatrix} = \begin{bmatrix} C_{Y\beta} & C_{Yp} & C_{Yr} - 2\mu & C_{Z_s} & 0 \\ C_{l\beta} & C_{lp} & C_{lr} & 0 & 0 \\ C_{n\beta} & C_{np} & C_{nr} & 0 & 0 \\ 0 & 1 & \tan \theta_s & 0 & 0 \\ 0 & 0 & \sec \theta_s & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & C_{Y\delta_r} \\ C_{l\delta_a} & C_{l\delta_r} \\ C_{n\delta_a} & C_{n\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (3.81)$$

Haciendo un símil con el modelo dinámico propuesto al comienzo de la sección 3.2.1.2 se pueden identificar claramente las matrices M , A y B , de manera que

$$M_{LD} = \begin{bmatrix} 2\mu & 0 & 0 & 0 & 0 \\ 0 & \widehat{I}_x & -\widehat{I}_{xz} & 0 & 0 \\ 0 & -\widehat{I}_{xz} & \widehat{I}_z & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.82)$$

$$A_{LD} = \begin{bmatrix} C_{Y\beta} & C_{Yp} & C_{Yr} - 2\mu & C_{Z_s} & 0 \\ C_{l\beta} & C_{lp} & C_{lr} & 0 & 0 \\ C_{n\beta} & C_{np} & C_{nr} & 0 & 0 \\ 0 & 1 & \tan \theta_s & 0 & 0 \\ 0 & 0 & \sec \theta_s & 0 & 0 \end{bmatrix} \quad (3.83)$$

$$B_{LD} = \begin{bmatrix} 0 & C_{Y\delta_r} \\ C_{l\delta_a} & C_{l\delta_r} \\ C_{n\delta_a} & C_{n\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.84)$$

donde el subíndice “LD” hace referencia al problema Lateral-direccional.

Una vez más aplicando el mismo cambio de variable que el de la ecuación (3.28) se iguala la notación a la desarrollada en la formulación general del problema.

$$A_{LD} = M_{LD}^{-1}A_{LD}, \quad B_{LD} = M_{LD}^{-1}B_{LD} \quad (3.85)$$

El siguiente paso sería el montaje de las matrices F_{LD} y G_{LD} que definen el modelo de predicción del controlador $x_{s_{LD}} = F_{LD}x_{k_{LD}} + G_{LD}u_s$. Como ya se comentó en el apartado 3.2.1.2.1 este proceso ha sido implementado en una función de Matlab, cuyo código puede ser consultado en el apéndice correspondiente de esta memoria. (Apéndices)

A continuación se definen las matrices Q y R de la misma manera que se realizó en el problema longitudinal con la salvedad de que, en este caso, la matriz Q tendrá dimensiones diferentes a la del problema longitudinal ya que se ha formulado un modelo con cinco variables de estado frente a las cuatro que se presentaban en el modelo longitudinal.

$$Q = \begin{bmatrix} Q_{LD} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Q_{LD} \end{bmatrix}_{N \times n, N \times n}, \quad R = \begin{bmatrix} R_{LD} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{LD} \end{bmatrix}_{N \times m, N \times m} \quad (3.86)$$

El significado de estas matrices es el mismo que en el caso anterior, es decir, son matrices definidas positivas que ponderan el peso de las variables de control (R) y de las variables de estado (Q) en el problema de optimización. Además, como se puede ver en la ecuación anterior (3.86) son matrices definidas por bloques de tamaños iguales al número de variables de estado del sistema por el número de puntos del horizonte de predicción (Q), y al número de variables de control del sistema por el número de puntos del horizonte de predicción (R).

Por su parte, Q_{LD} y R_{LD} se definen como sigue

$$Q = \begin{bmatrix} \frac{p_v}{(\Delta v)^2} & 0 & 0 & 0 & 0 \\ 0 & \frac{p_p}{(\Delta p)^2} & 0 & 0 & 0 \\ 0 & 0 & \frac{p_r}{(\Delta r)^2} & 0 & 0 \\ 0 & 0 & 0 & \frac{p_\phi}{(\Delta \phi)^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{p_\psi}{(\Delta \psi)^2} \end{bmatrix}_{5 \times 5}, \quad R = \begin{bmatrix} p_{\delta_a} & 0 \\ 0 & p_{\delta_r} \end{bmatrix}_{2 \times 2} \quad (3.87)$$

donde $p_v, p_p, p_r, p_\phi, p_\psi, p_{\delta_a}$ y p_{δ_r} son los pesos asignados a cada una de las variables que intervienen en el problema, y $\Delta_v, \Delta_p, \Delta_r, \Delta_\phi$ y Δ_ψ representan variaciones típicas de la variables v, p, r, ϕ y ψ respectivamente, las cuales adquieren el mismo significado que las expuestas en el MPC longitudinal.

Al igual que en el caso anterior, es importante realizar un buen ajuste de los pesos para evitar errores en la respuesta de la aeronave al seguimiento de la trayectoria.

Los únicos elementos que faltarían para poder definir el problema de optimización son las matrices H y f . Para lo que se ha hecho uso de la ecuación (3.53).

$$H_{LD}^* = 2(G_{LD}^T Q_{LD} G_{LD} + A_u^T R_{LD} A_u), \quad (3.88)$$

$$f^* = 2(F_{LD} x_{k_{LD}} - x_{ref_{LD}})^T Q_{LD} G_{LD} - 2u_{s-1_{LD}} R_{LD} A_u \quad (3.89)$$

Con todos los elementos que intervienen en el problema de optimización definidos, éste se puede resolver aplicando la ecuación (3.54)

$$u_{s_{LD}}^{opt} = -H_{LD}^{*-1} f_{LD}^* \quad (3.90)$$

Al igual que para el problema longitudinal, la solución $u_{s_{LD}}^{opt}$ representa un vector de predicciones de señales de control de $N \times n$ componentes (horizonte de predicción por número de señales de control del problema Lateral-direccional),

$$u_{LD}^{opt} = \begin{bmatrix} \delta_{a_{k_0}} \\ \delta_{r_{k_0}} \\ \delta_{a_{k+1}} \\ \delta_{r_{k+1}} \\ \vdots \\ \delta_{a_{k+N-1}} \\ \delta_{r_{k+N-1}} \end{bmatrix} \quad (3.91)$$

que optimiza la función de coste cuadrática definida. Las primeras dos componentes de este vector

(δ_{a_k} y δ_{r_k}) serán las que reciba y utilice el modelo de avión para determinar el estado futuro de la aeronave. Este nuevo estado calculado realimenta al controlador junto con la referencia longitudinal y las dos componentes mencionadas, para que éste pueda volver a resolver el sistema en el instante siguiente.

Esta solución, ecuación (3.91), se trata de la solución del problema de optimización sin restricciones. Para determinar la solución del problema con restricciones habría que aplicarle la formulación propuesta en el apartado 3.2.1.2.1.

Del mismo modo que en el apartado 3.2.2.1, todo el código generado para poder montar las matrices que intervienen en el problema y resolverlo puede ser consultados en los apéndices correspondientes de esta memoria (Apéndices).

3.2.4 Descripción del modelo Simulink e implementación en el modelo original

A lo largo de este apartado se va a proceder a explicar el modelo Simulink que ha permitido implementar el sistema de control de actitud diseñado en el presente trabajo. En ese sentido, hay que destacar que sólo serán descritos los bloques que componen el sistema sin entrar en el detalle del código de programación que esconden. En caso de que el lector desee consultar dichos códigos, se pueden localizar en los apéndices de la presente memoria. Finalmente, se describe brevemente el proceso de implementación en el entorno de simulación original

Para comenzar con la descripción de los diferentes bloques diseñados, se parte del modelo general del sistema (Figura 3.8), el cual se ha basado en el modelo original desarrollado en la tesis [1] (Figura 3.9).

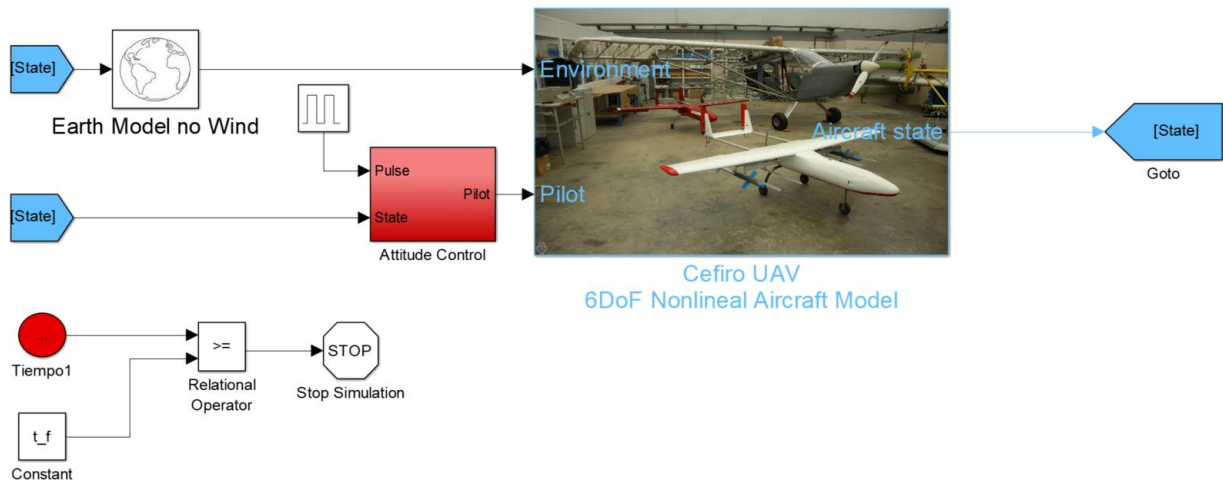


Figura 3.8 Modelo general del sistema.

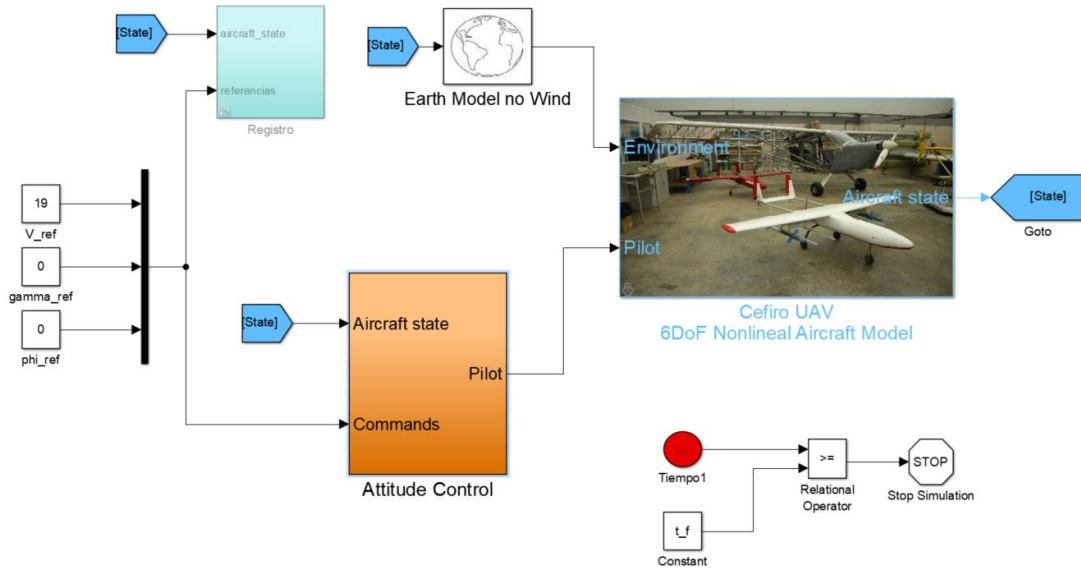


Figura 3.9 Entorno de simulación original.

Con respecto a la estructura del modelo creado para este proyecto, se presenta un modelo de Tierra sin viento, el modelo de 6 G.D.L no lineal del UAV Céfiro y el controlador. Entrando en el sistema de control de actitud (Figura 3.10), se puede comprobar que el sistema presenta cuatro variables realimentadas que se corresponden con las señales de control óptimas utilizadas por el controlador para volver a resolver el problema de optimización y con las salidas del error acumulado en régimen permanente, que deben ser realimentados para poder implementar la corrección, desarrollada en la sección 3.2.1.2.2 de este capítulo, de dicho error.

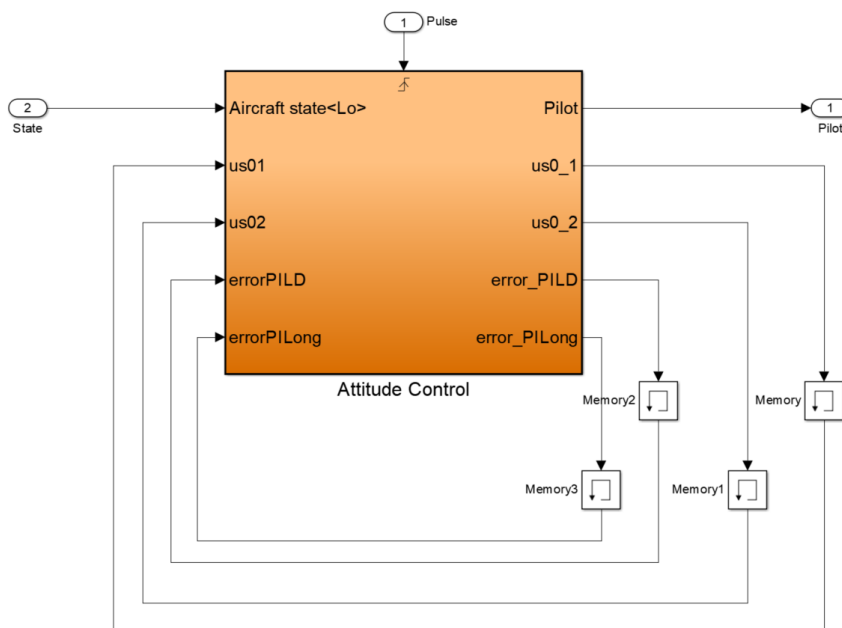


Figura 3.10 Modelo del sistema de control de actitud.

Es importante destacar el input 1 del sistema “Pulse”. Este input genera un pulso de valor unidad cada 20 milisegundos, que coincide con el intervalo de muestreo. La razón por la que se genera este pulso reside en que el modelo de avión es un modelo definido en el tiempo continuo; sin embargo, el controlador es un modelo discreto que sólo puede ejecutarse cada T_s segundos.

Para lograr que no se produzcan errores en la simulación se coloca un Trigger en el modelo de MPC completo, que provocará que éste se ejecute únicamente cuando reciba el pulso de entrada. De esta manera, se consigue que el modelo de controlador realice los cálculos pertinentes cada intervalo de muestreo mientras que se evitan errores de compilación del sistema completo.

Accediendo al bloque de control de actitud (Figura 3.11) se pueden distinguir claramente tres sistemas de Simulink. El primero de ellos, que tiene como input el estado de la aeronave, agrupa las variables de estado según pertenezcan a la dinámica longitudinal o a la lateral-direccional y, posteriormente, las adimensionaliza para que puedan ser tratadas por el modelo de dinámica adimensional propuesto. Los otros dos se corresponden con sendos controladores, lateral-direccional y Longitudinal. Además, se observa el bloque trigger comentado con anterioridad.

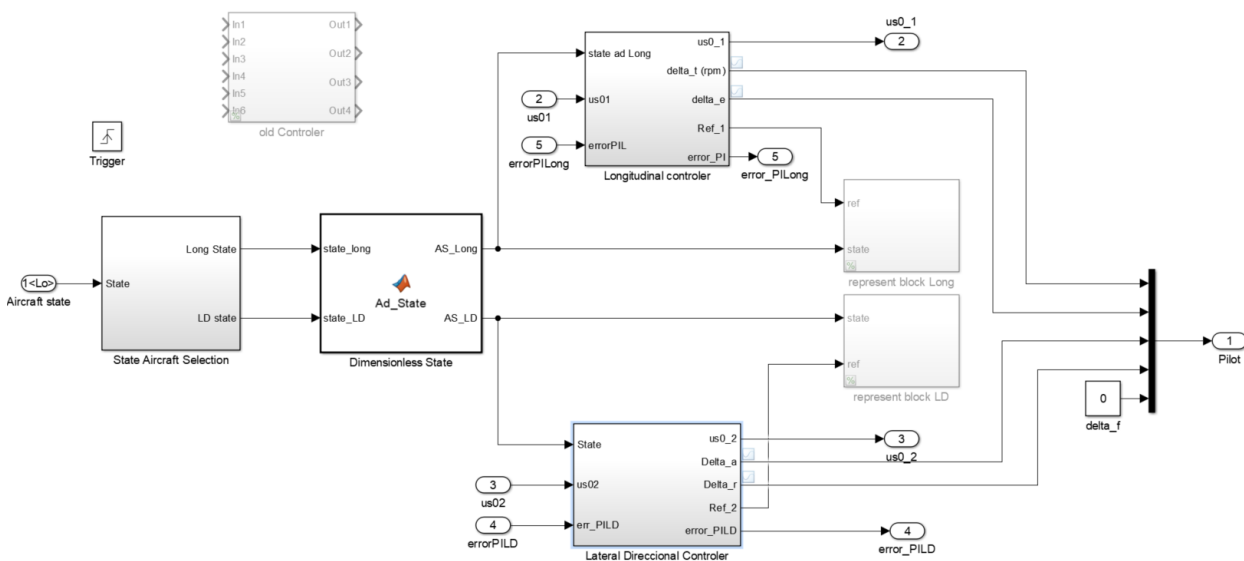


Figura 3.11 Estructura del controlador desglosado.

A continuación, se pasan a describir ambos controladores individualmente, realizando una serie de comentarios con respecto a ellos.

- **MPC longitudinal**

En la Figura 3.12 queda recogida la estructura del bloque correspondiente al controlador longitudinal. Ésta se encuentra compuesta por el MPC, que es donde se resuelve el problema de optimización y se obtiene la secuencia de señales de control, dos saturaciones, que evitan que la señal de control que le llega al avión supere los límites permitidos (hay que recordar que este modelo no está formulado con restricciones), un bloque que transforma la posición de palanca en revoluciones, ya que el modelo de avión trabaja con ellas, y un bloque que permite representar la referencia en tiempo real y, consecuentemente, comprobar las diferencias existentes entre el estado del avión y la referencia impuesta.

Con respecto a las salidas del sistema hay que recordar que las dos únicas entradas necesarias para el modelo de avión (se corresponden con las salidas del controlador) son las dos primeras componentes del vector de señales de control solución del problema de optimización, esto es, la deflexión del timón de profundidad y la posición de la palanca de gases. No obstante, se han generado otras salidas del sistema para su representación gráfica.

Para finalizar este apartado se va a centrar la atención en el bloque del MPC longitudinal para dar un último apunte al mismo. El párrafo anterior se ha centrado en describir los subsistemas que componen el modelo sin realizar ninguna mención de las entradas. Éstas, tal y como puede ser corroborado en la imagen anterior (Figura 3.12), se corresponden con las ya comentadas durante toda la formulación del MPC (estado de la aeronave adimensional, referencias y vector de señales de control del instante de simulación anterior); sin embargo, consecuencia de la corrección del error en régimen permanente, aparece una nueva entrada (y salida, debido a la realimentación de la misma) que se corresponde con el error en permanente “acumulado” y cuya definición se ha llevado a cabo en el apartado 3.2.1.2.2. Por último, habría que añadir como entradas del sistema todas las matrices desarrolladas a lo largo del anterior sub-apartado.

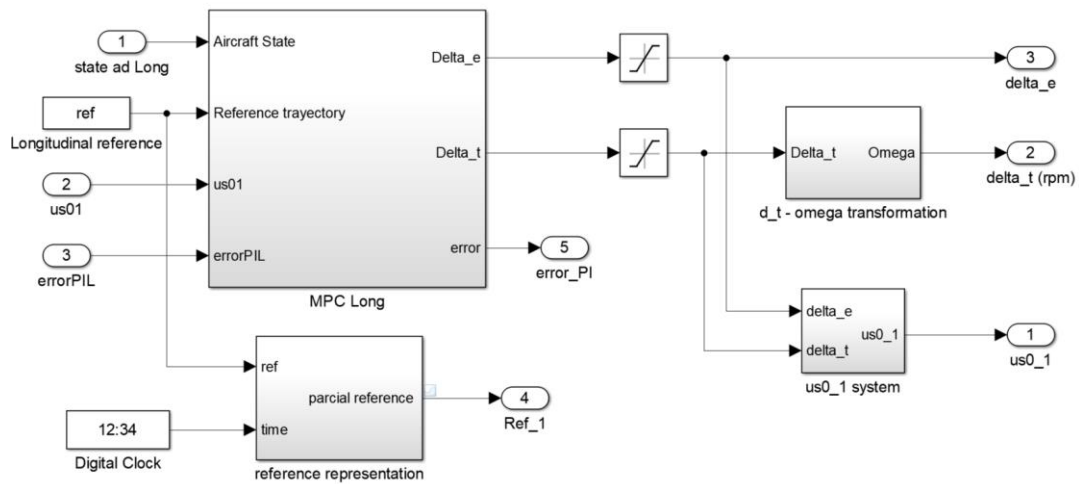


Figura 3.12 Bloque Longitudinal Controller.

• **MPC lateral-direccional**

Este bloque del MPC lateral-direccional (Figura 3.13) presenta una estructura similar al correspondiente al controlador longitudinal (Figura 3.12) y está compuesto por los mismos bloques ya comentados anteriormente.

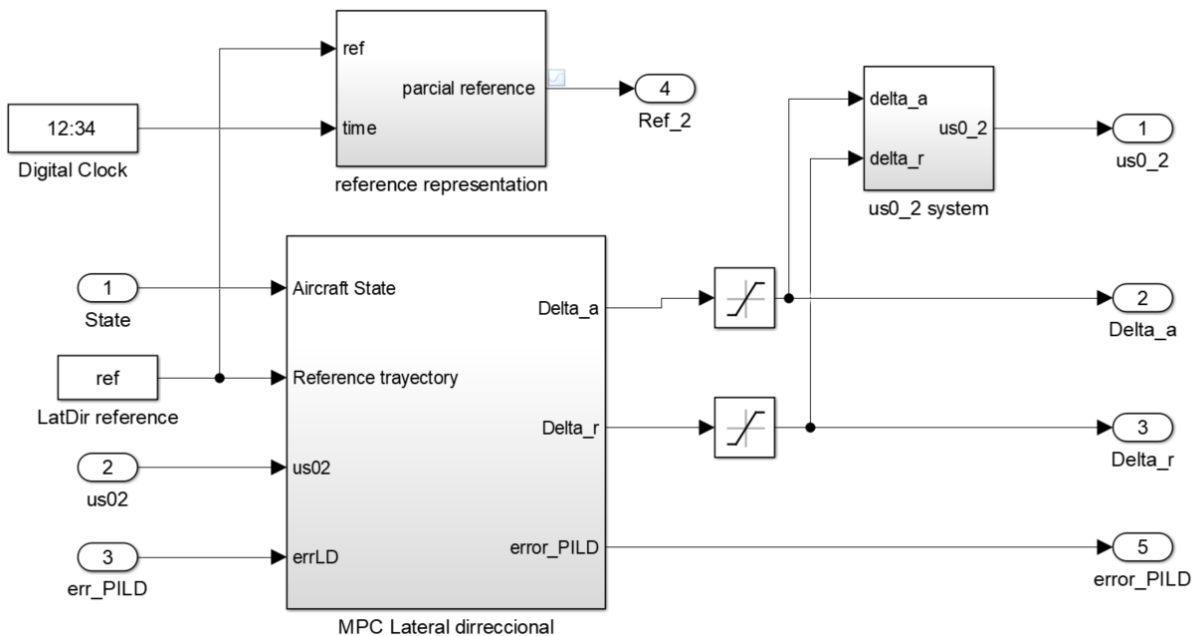


Figura 3.13 Bloque Lateral Direccional Controller.

Con respecto a las entradas, tal y como aparece en la formulación del controlador, el sistema necesita una serie de entradas básicas para poder resolver el problema de optimización. Estas entradas básicas, las cuales pueden verse en la Figura 3.13, son

- Estado: solo se computan las variables de estado correspondientes a la dinámica Lateral-direccional.
- u_{s-1} ($us02$): como ya se ha explicado el controlador necesitar ser realimentado con las señales de control del instante anterior.
- Error acumulado en permanente (err_PILD): Esta variable será utilizada para corregir el error en régimen permanente que presenta el controlador.

además, se necesitarán todas las matrices definidas a lo largo del sub-apartado anterior.

Con respecto a las salidas del sistema, al igual que ocurría para el MPC longitudinal, las únicas entradas que requiere el modelo de avión (se corresponden con las salidas del controlador) son las dos primeras componentes del vector de señales de control solución del problema de optimización, es decir, la deflexión del timón de dirección y la de los alerones. No obstante, también se han creado una serie de salidas para su representación gráfica y, además, se ha asignado el error “acumulado” en régimen permanente a otra de las salidas para su posterior realimentación.

Entrando en la estructura del bloque, éste se encuentra compuesto por dos saturaciones, que tienen la misma función que las existentes en el controlador longitudinal, un bloque generador de referencias parciales, que ha sido usado para poder representar dichas referencias frente al estado de la aeronave, un bloque que genera el vector u_{s-1} ($us0_2$ system), que realimenta al controlador y el propio controlador, que realiza una llamada a los scripts de Matlab correspondientes.

3.2.4.1 Implementación en el entorno original de simulación

Se va a proceder ahora a exponer el proceso por el cual el Sistema de control de actitud original ha sido sustituido por el generado en este trabajo. Para ello se parte del modelo de Simulink original (Figura 3.9). En éste se presentan un modelo de Tierra sin viento, el modelo de 6 G.D.L no lineal del UAV Céfito y el sistema de control de actitud. La organización de este último queda recogida en la Figura 3.14 y, como puede verse, está compuesto por dos sub-controladores (controlador de ángulo de trayectoria y controlador de velocidad), los cuales conforman el sistema de control de actitud longitudinal, y un controlador lateral-direccional.

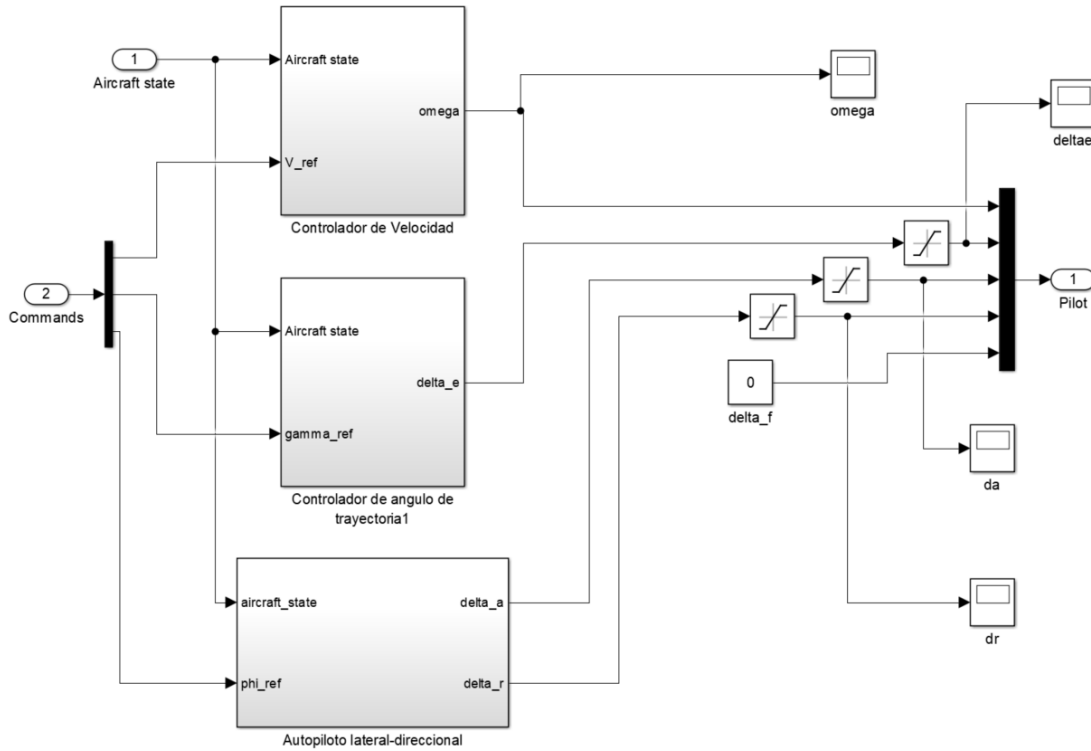


Figura 3.14 Controlador del modelo original.

Es de especial relevancia el hecho de que, como ya se ha comentado con anterioridad, las variables de estado longitudinales controladas por el sistema de actitud original son distintas a las que se han utilizado inicialmente a la hora de formular el MPC longitudinal. En esta línea, tal y como se recoge en la sección 3.2.2.2, se introduce una modificación al controlador para cambiar su formulación a las variables u , q y γ . No obstante, este cambio se realiza únicamente a nivel de código por lo que desde el punto de vista del modelo Simulink el controlador es alimentado con las variables iniciales del MPC. Esto supone un problema a la hora de implementar el nuevo controlador ya que, como es evidente, las variables de control no son las mismas. Ante esta situación se decide eliminar los controladores antiguos y diseñar un bloque nuevo desde cero, pero conservando las saturaciones y el montaje de la salida realizado.

Además, esta decisión supuso una reorganización del bloque principal del modelo Simulink y un cambio en la manera de generar las referencias. De esta forma, se pueden ver ciertas diferencias entre los layout principales de cada modelo (Figura 3.8 y Figura 3.9).

En base a todo lo anterior, la integración del nuevo sistema de control de actitud se reduce a la eliminación del contenido del bloque “Attitude Control” de la Figura 3.14, introducción de las realimentaciones, que se encuentran recogidas en la Figura 3.10, e introducción del bloque de la Figura 3.11.

4 RESULTADOS

A lo largo de este trabajo se ha desarrollado un sistema de control de actitud predictivo que permite obtener las señales de control que debe ejecutar la aeronave para seguir una referencia dada. Para ello se ha necesitado definir un modelo de predicción basado en las características de la aeronave, un horizonte de predicción y una condición de optimización para la obtención de la solución.

Posteriormente, a ese modelo original se le han añadido una serie de modificaciones para, bien mejorar la actuación del sistema o bien ser integrado con el sistema general creado en la referencia [1].

Tras completar el desarrollo del modelo y de las correcciones incluidas, se debe realizar un estudio de resultados y una comparativa entre los mismos. En ese sentido, en el presente apartado se van a exponer las gráficas obtenidas en las diferentes simulaciones realizadas para, posteriormente, llevar a cabo un análisis de las mismas y presentar las bases en las que se apoyarán las conclusiones obtenidas y expuestas en el capítulo 5

Para ello se va a dividir el capítulo en diferentes secciones que se corresponderán con cada una de las casuísticas siguientes

- Modelo de MPC original con control en las variables u, γ, β y ϕ ,
- Modelo de MPC original con control en las variables $u, \gamma, q, \beta, \phi$ y p .
- Modelo de MPC con corrección en permanente y control en las variables $u, \gamma, q, \beta, \phi$ y p .
- Modelo de MPC con saturaciones, corrección en permanente y control en las variables u, γ, q, v, ϕ y p .
- Comparación entre el modelo de MPC con saturaciones y los resultados obtenidos con la función quadprog. Cómputo de tiempos de ejecución del controlador

Como se podrá comprobar en las gráficas, se ha decidido obviar en la presentación de resultados las simulaciones con las variables de estado originales, ya que no aportan información relevante alguna

y uno de los objetivos futuros de este proyecto es su integración con el sistema general explicado en el capítulo 2, por lo que carece de sentido seguir trabajando con dichas variables.

Otro apunte que hay que tener en cuenta es que las variables que aparecen en las gráficas son dimensionales; sin embargo, el MPC está formulado en variables adimensionales, y de ahí que se designen los títulos de las secciones con las variables adimensionales.

Con respecto a las condiciones de simulación se ha decidido tomar 30 puntos de horizonte de predicción, es decir, 0.6 segundos. El tiempo de simulación es de 30 segundos y las referencias impuestas serán diferentes en cada simulación por lo que serán mostradas en las gráficas correspondientes.

Por último, cabe destacar que con la estructura propuesta se ha pretendido que el lector sea consciente de las etapas por las que ha pasado el proyecto en su desarrollo. De la misma manera, se favorece la identificación de las mejoras obtenidas con la introducción de las correcciones.

4.1 Modelo original de MPC con control en u , γ , β y ϕ

- Variables longitudinales.

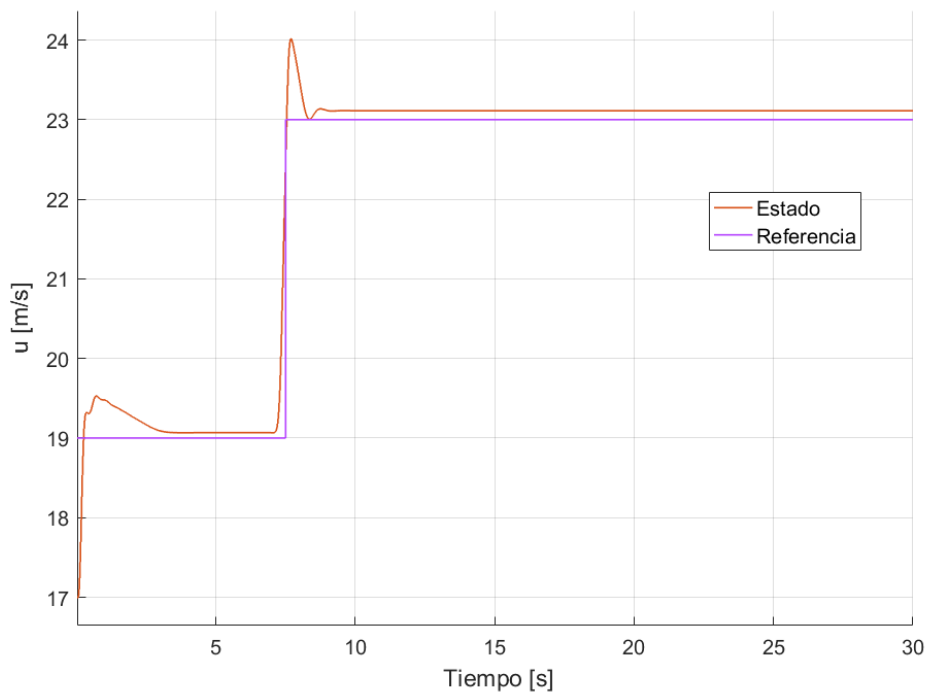


Figura 4.1 Caso 1, velocidad aerodinámica en el eje x.

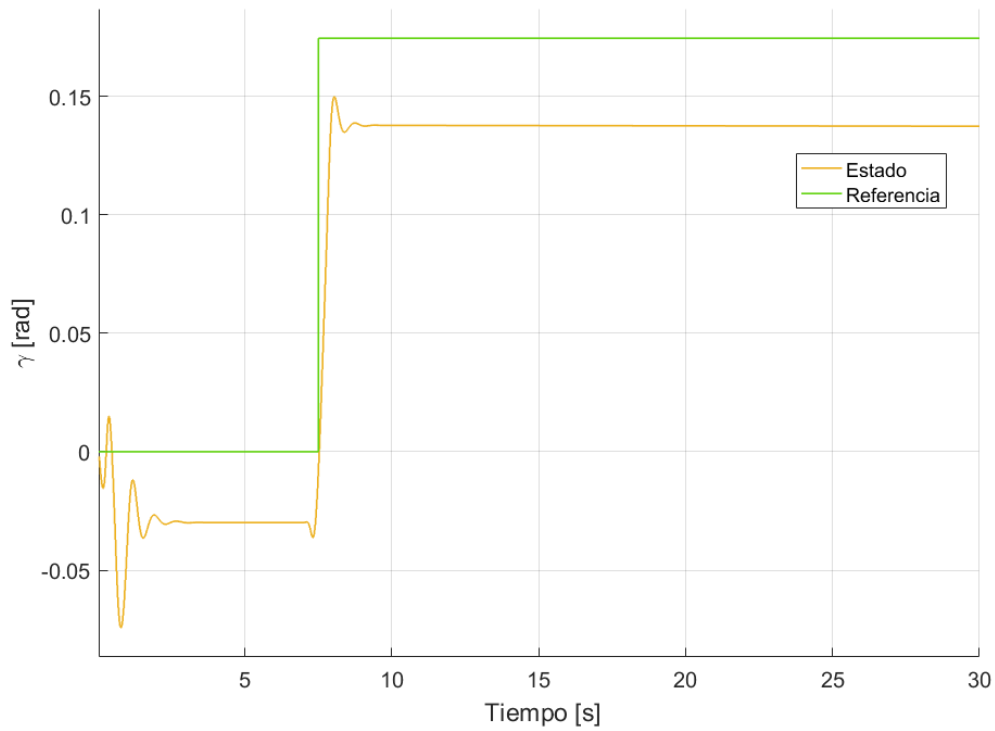


Figura 4.2 Caso 1, ángulo de trayectoria.

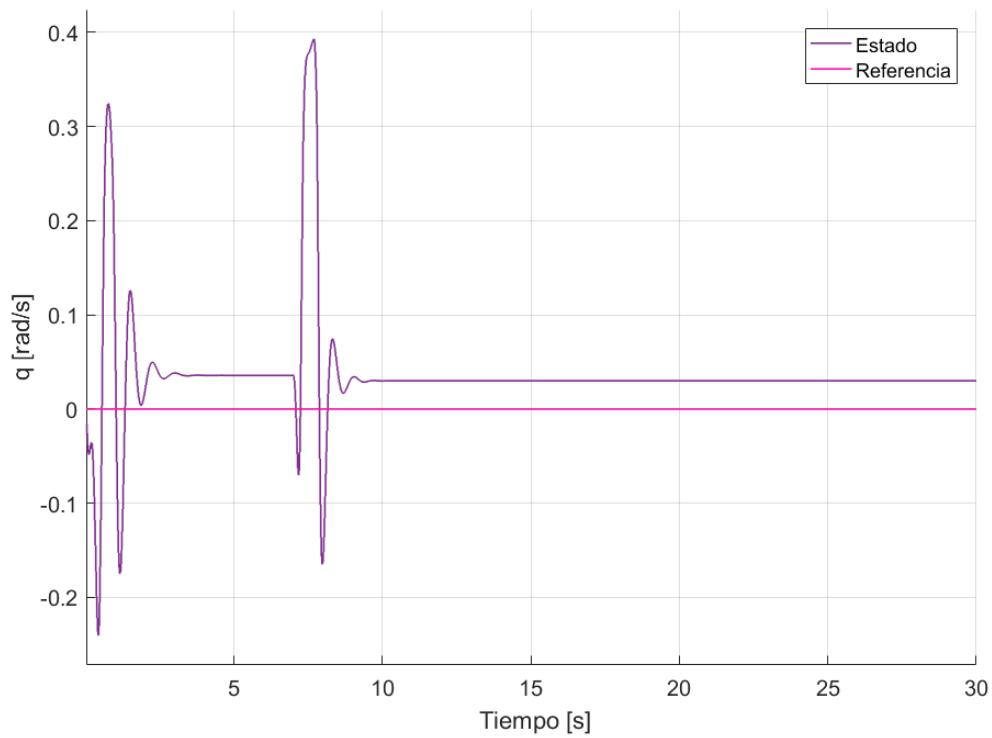


Figura 4.3 Caso 1, velocidad angular de cabeceo.

Observando las imágenes adjuntas (Figura 4.1, Figura 4.2 y Figura 4.3) se deben comentar dos cuestiones principales. La primera de ellas, y más evidente, es el error en régimen permanente que presentan todas las variables y que hace que el estado de la aeronave nunca alcance a la referencia deseada. La segunda, y menos clara, es la sobre-oscilación que aparece en las variables u y γ ; este resultado es debido a que no se está llevando a cabo un control en la variable q (puede comprobarse fácilmente en la Figura 4.3 Caso 1, velocidad angular de cabeceo.), el cual tiene un efecto estabilizador en la aeronave, tal y como se demuestra en la próxima sección (4.2).

- **Variables laterales-direccionales.**

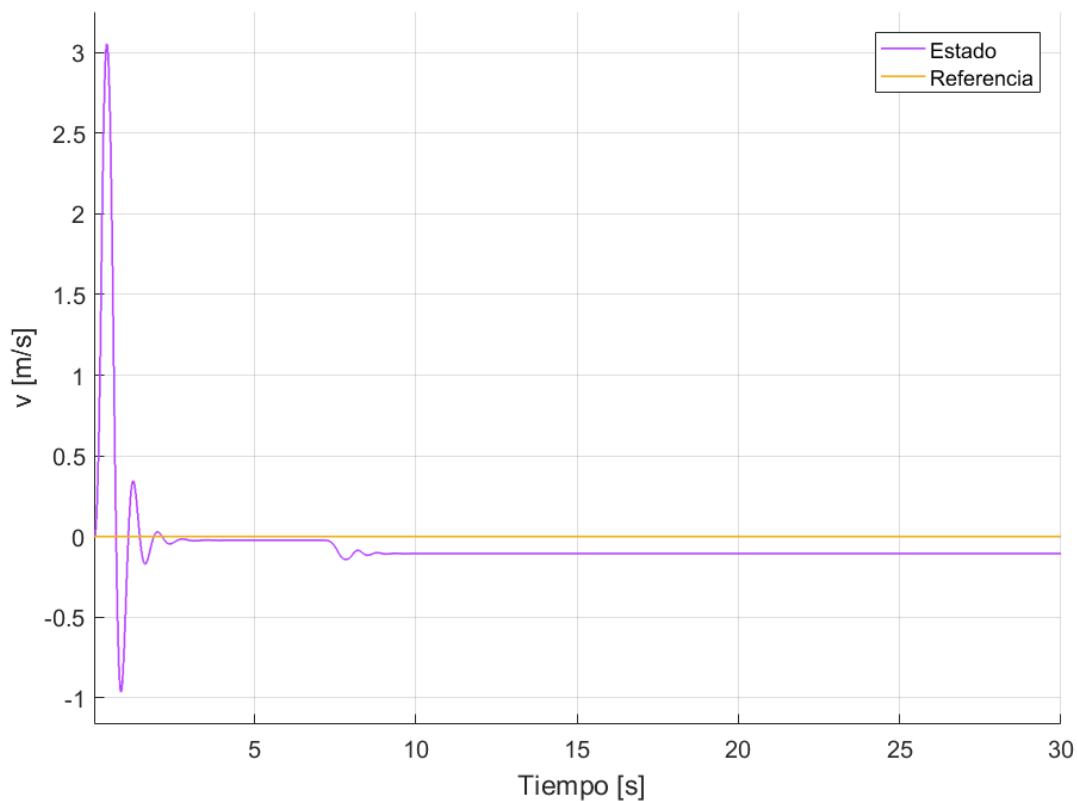


Figura 4.4 Caso 1, velocidad aerodinámica en el eje y.

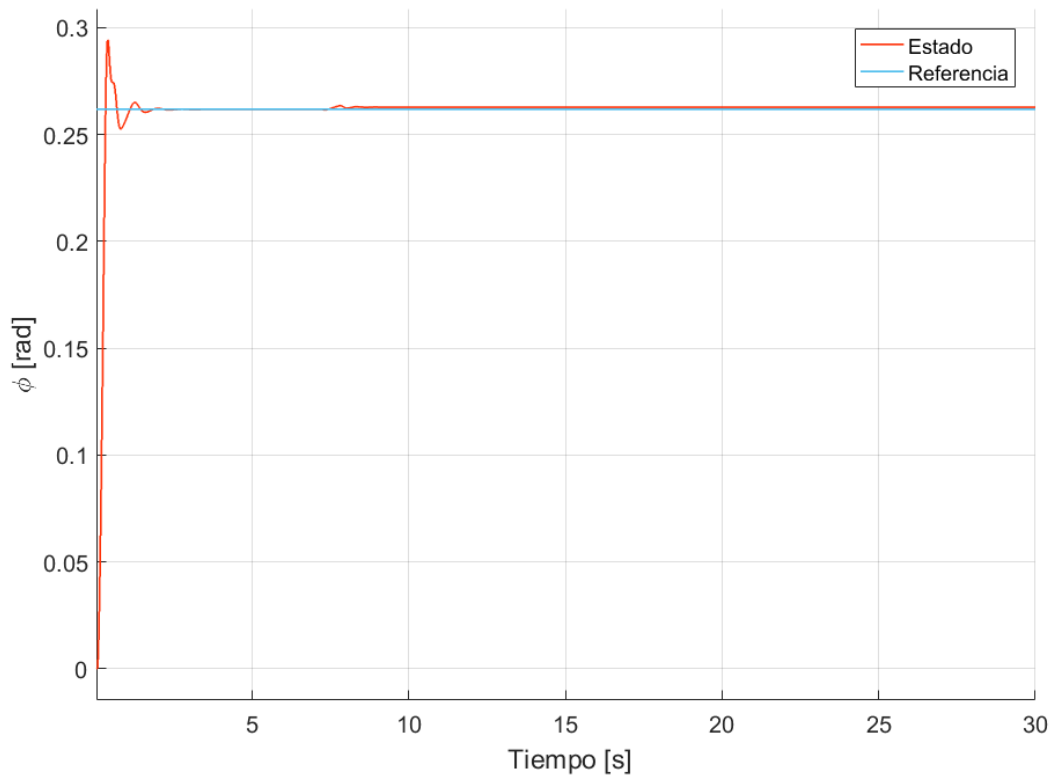


Figura 4.5 Caso 1, ángulo de balance.

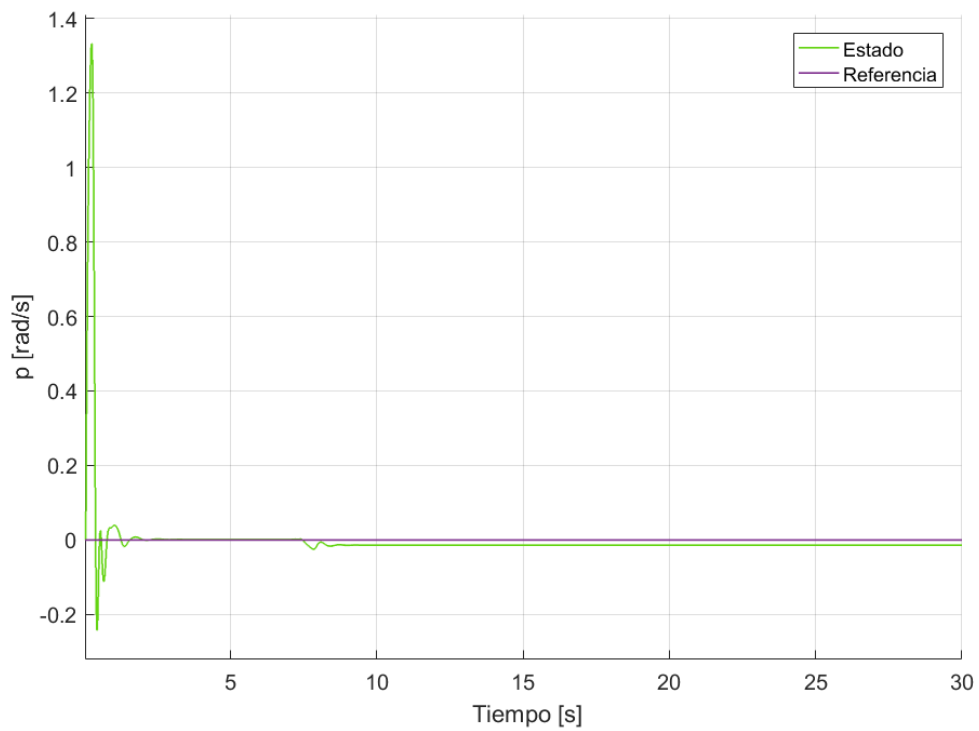


Figura 4.6 Caso 1, velocidad angular de alabeo.

Al igual que ocurría en el caso anterior, aparece un error en régimen permanente en todas las variables; sin embargo, éste es menor que el de la dinámica longitudinal. La razón de ello se basa en que la dinámica lateral-direccional de la aeronave es más estable y, en consecuencia, las desviaciones afectan menos. Por otro lado, también aparece la sobre-oscilación mencionada con anterioridad, consecuencia de la ausencia de control en la variable p .

4.2 Modelo original de MPC con control en $u, \gamma, q, \beta, \phi$ y p

- Variables longitudinales.

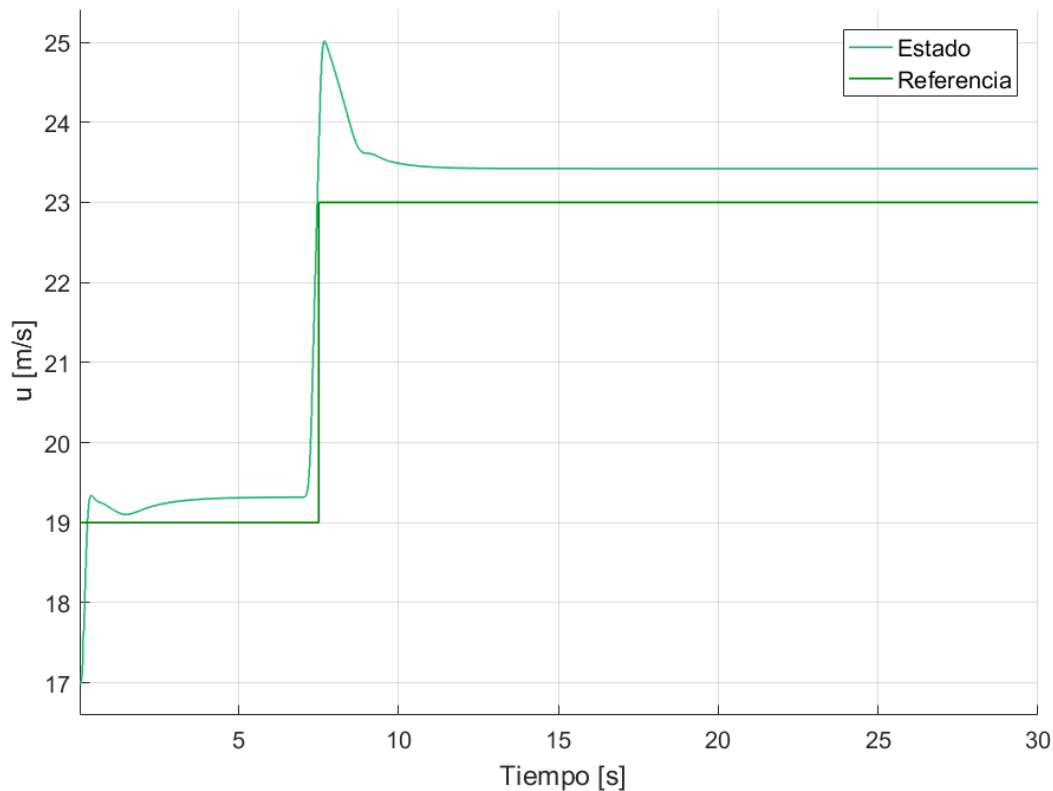


Figura 4.7 Caso 2, velocidad aerodinámica en el eje x.

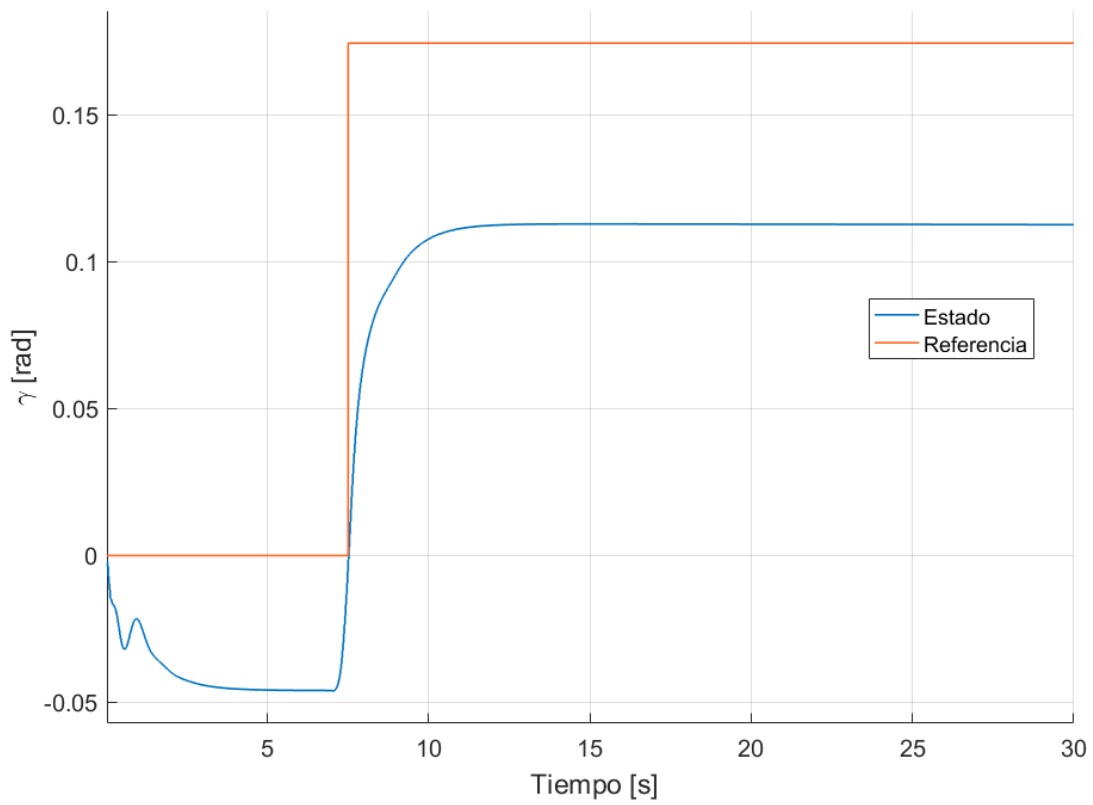


Figura 4.8 Caso 2, ángulo de trayectoria.

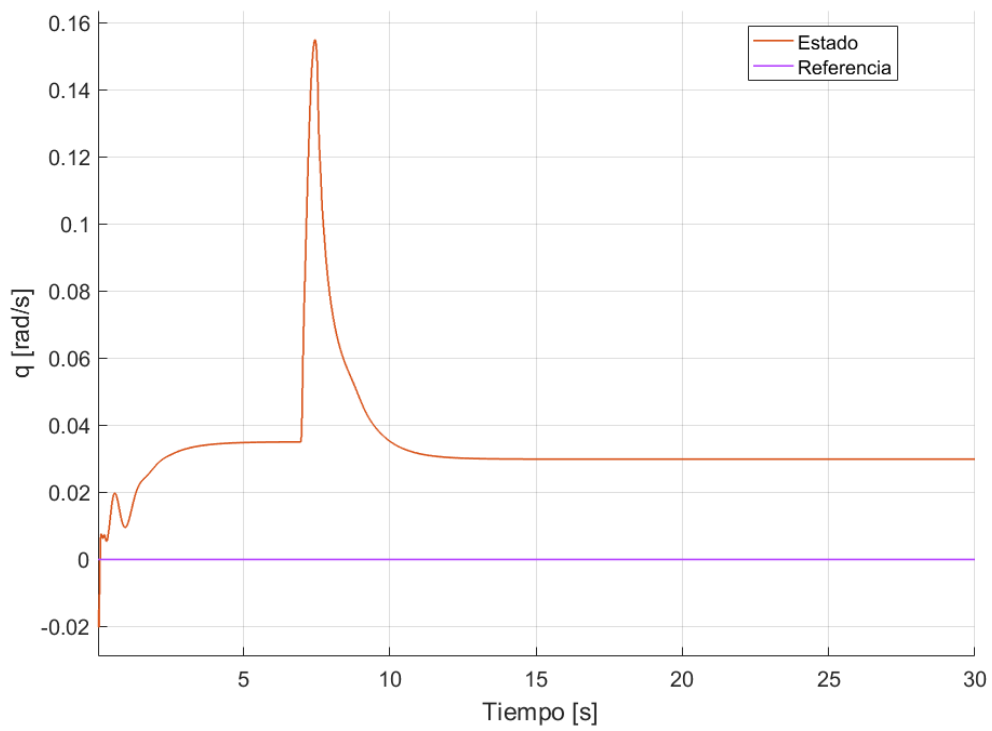


Figura 4.9 Caso 2, velocidad angular de cabeceo.

Como puede comprobarse en la Figura 4.9 en este caso se ha implementado un leve control en la variable q con el objetivo de estabilizar la aeronave y suavizar su respuesta. Esta acción, como se observa en las gráficas adjuntas (Figura 4.7 y Figura 4.8), supone una estabilización de la dinámica de la aeronave, consiguiéndose una respuesta en el resto de variables menos abrupta. No obstante, aparece una pérdida de la capacidad de control en las variables u y γ y, en consecuencia, un aumento del error en permanente. Ello se debe a que el sistema sólo presenta dos señales de control, por lo que, a priori, sólo se podrían controlar dos variables de estado. Sin embargo, se está forzando al MPC a controlar en tres variables, y de ahí que surja una reducción del control en las otras dos variables.

Esta discrepancia será solventada, tal y como queda recogido en las correspondientes secciones de este capítulo (4.3 y 4.4), mediante la introducción de las correcciones del error en permanente y de las saturaciones.

- **Variables laterales-direccionales.**

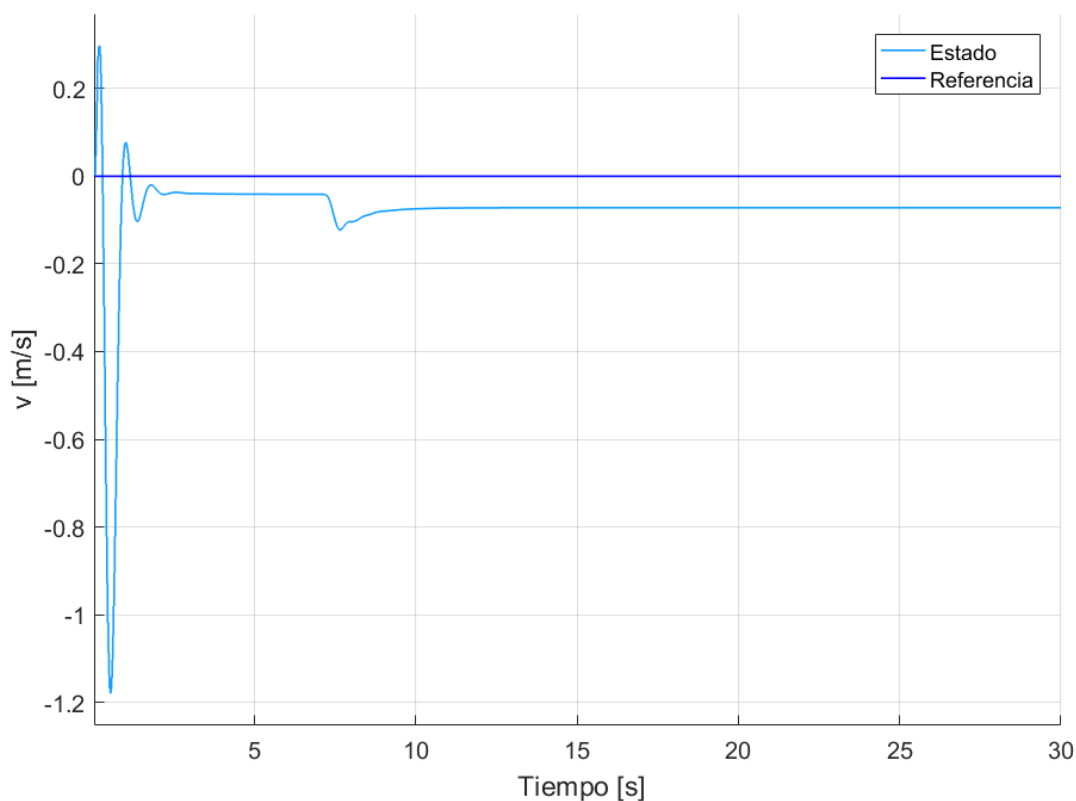


Figura 4.10 Caso 2, velocidad aerodinámica en el eje y.

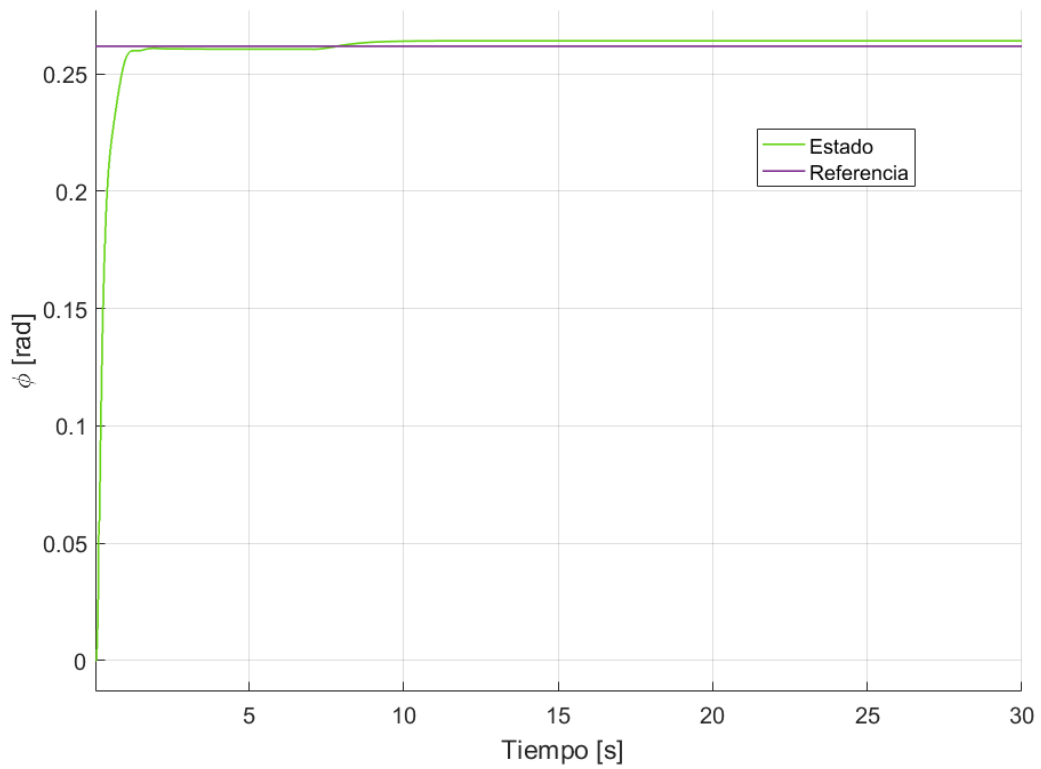


Figura 4.11 Caso 2, ángulo de balance.

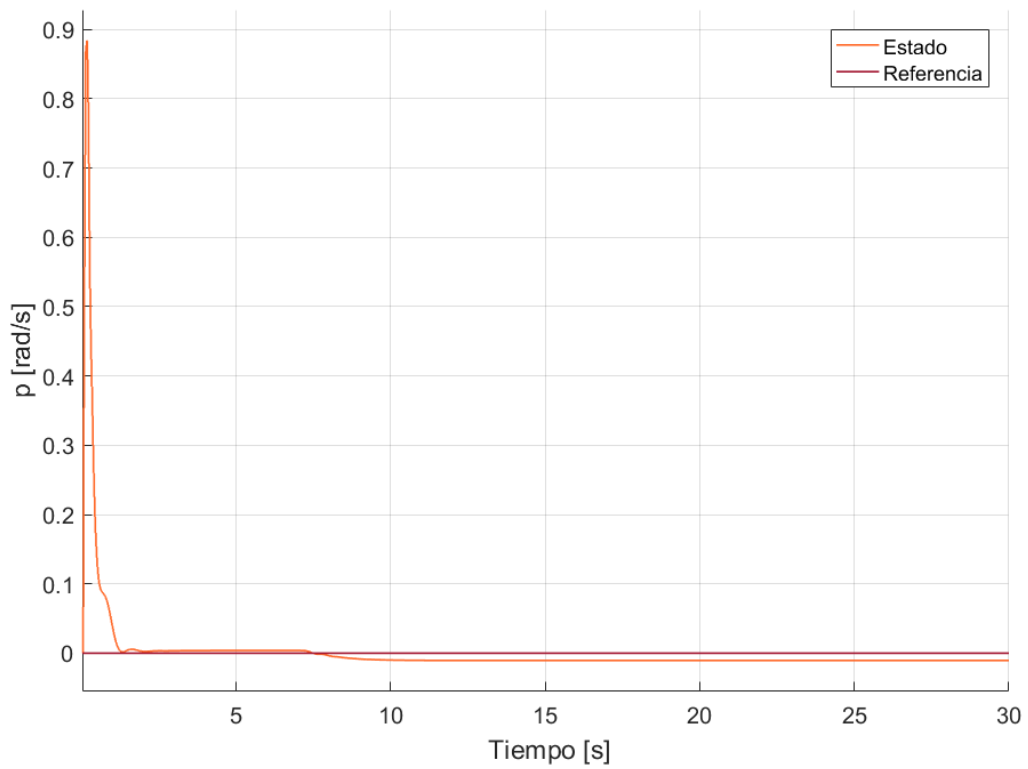


Figura 4.12 Caso 2, velocidad angular de alabeo.

Al igual que ocurre para las variables del problema longitudinal, el efecto de introducir un control leve en la variable p implica una suavización de la respuesta de la aeronave; sin embargo, también lleva asociado una pérdida de la capacidad de control en las otras variables. No obstante, dado que tanto el error en permanente como las sobre-oscilaciones de las variables laterales-direccionales eran menores, este efecto se ve reducido y prácticamente no se percibe.

En lo que sigue, el resto de simulaciones serán llevadas a cabo con el control en la seis variables expuestas debido a que, como se ha mencionado, la mejora en el comportamiento frente a sobre-oscilaciones es considerable, mientras que la pérdida de capacidad de control es subsanada con la introducción de las posteriores correcciones.

4.3 Modelo de MPC con corrección en permanente y control en $u, \gamma, q, \beta, \phi$ y p

- Variables longitudinales

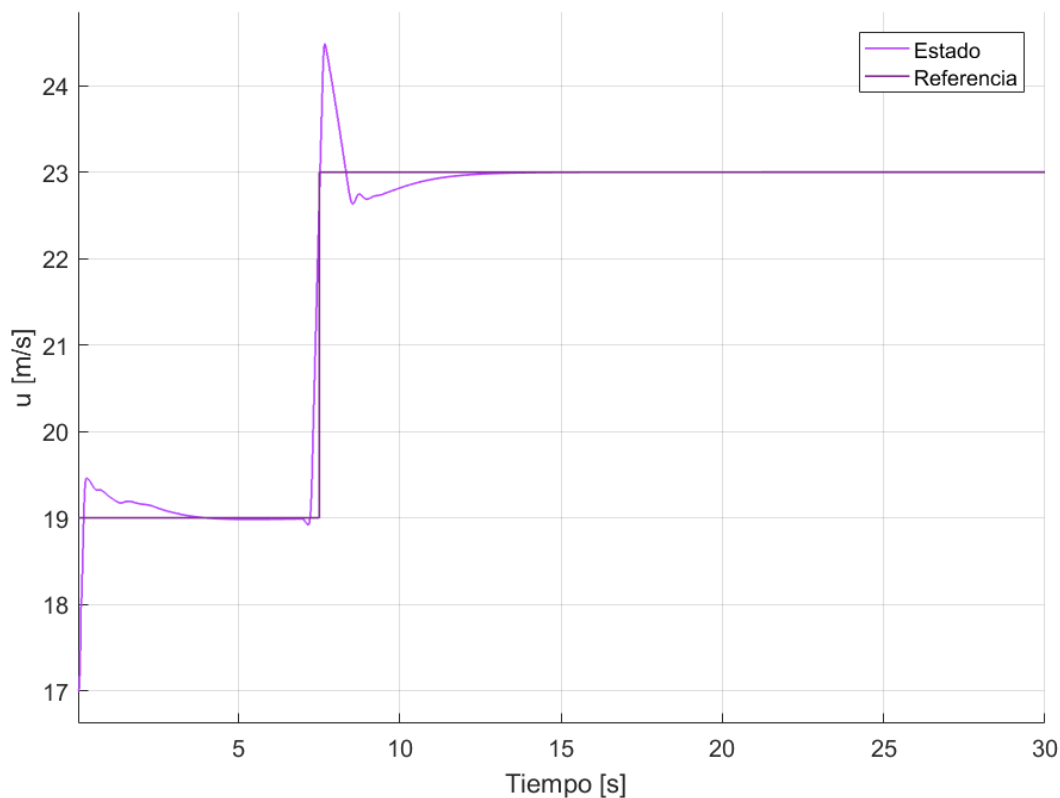


Figura 4.13 Caso 3, velocidad aerodinámica en el eje x.

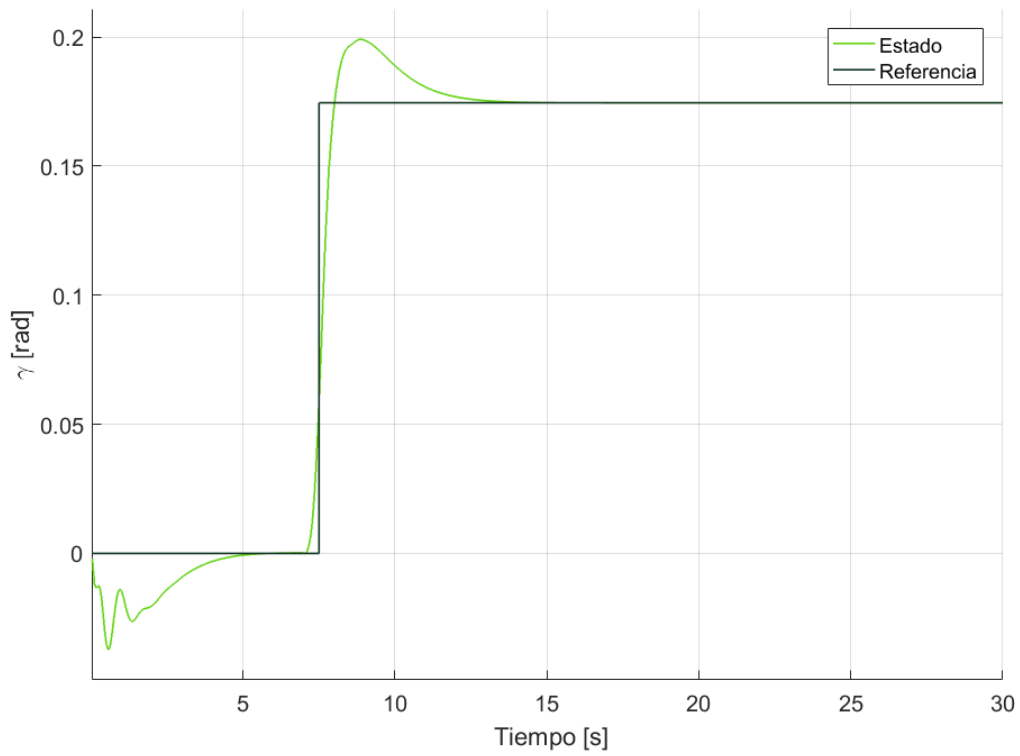


Figura 4.14 Caso 3, ángulo de trayectoria.

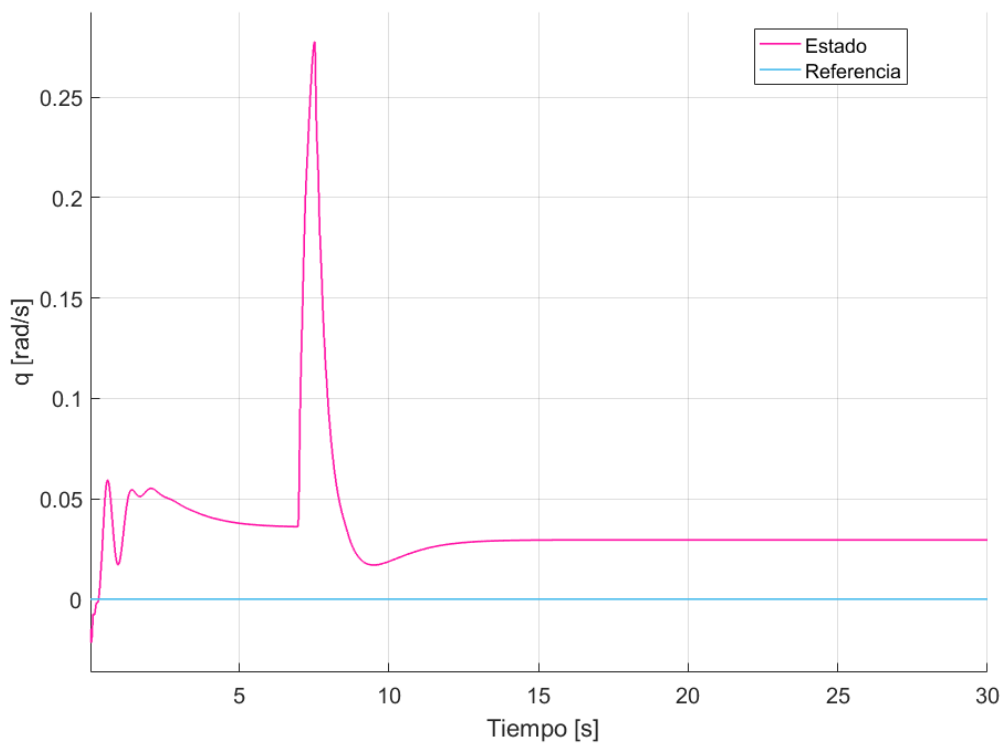


Figura 4.15 Caso 3, velocidad angular de cabeceo.

Tal y como queda recogido en las gráficas anteriores (Figura 4.13, Figura 4.14 y Figura 4.15), con

la introducción de la corrección del error en permanente se ha conseguido el objetivo fijado, es decir, se ha eliminado dicho error en la señal; esto permite que el estado de la aeronave alcance la referencia deseada.

- **Variables laterales-direccionales**

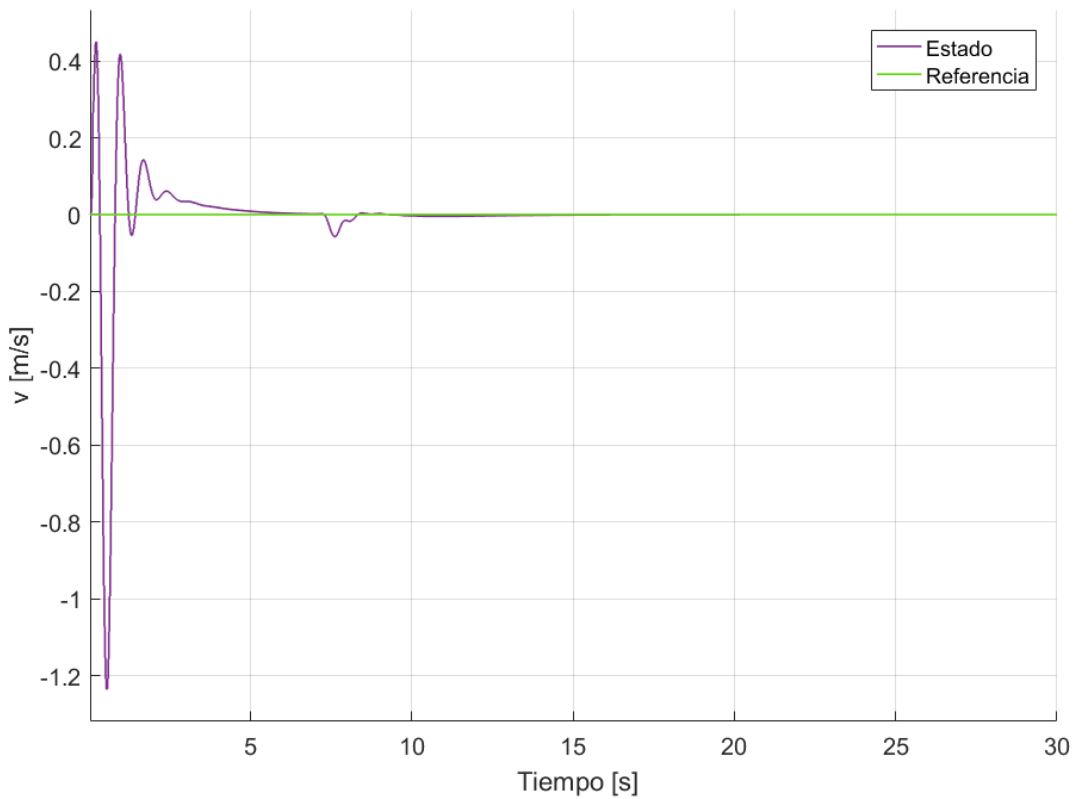


Figura 4.16 Caso 3, velocidad aerodinámica en el eje y.

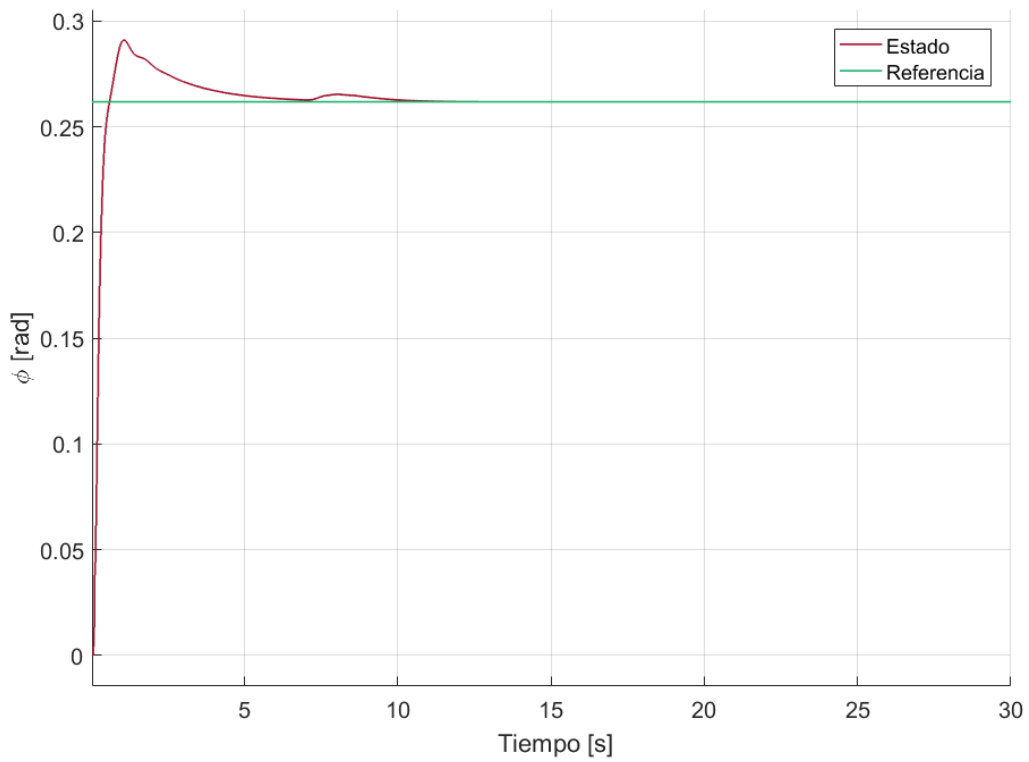


Figura 4.17 Caso 3, ángulo de balance.

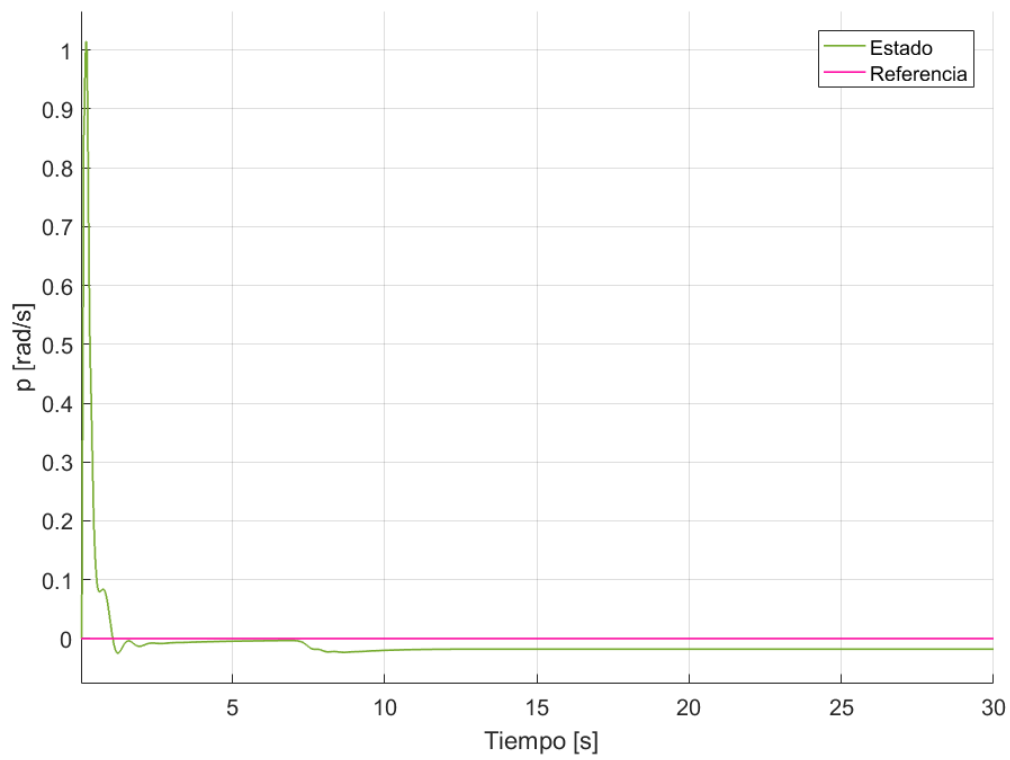


Figura 4.18 Caso 3, velocidad angular de alabeo.

Al igual que ocurre para el caso de las variables longitudinales, la respuesta de la aeronave en la dinámica lateral-direccional se ve mejorada con la eliminación del error en permanente y, por tanto, consigue alcanzar la señal de referencia.

No obstante, la introducción de esta modificación lleva un problema asociado puesto que, al forzar al controlador a corregir el error en permanente facilitándole una referencia modificada, se está empeorando su comportamiento en la parte transitoria. Este efecto, que se aprecia sobre todo en el control longitudinal, ya que es el más complejo de los dos desde el punto de vista del control, se ve reducido con la introducción de las saturaciones en el controlador; sin embargo, sigue estando latente. Por ello se plantea como línea futura de trabajo la mejora del sistema de corrección del error en permanente para intentar solventar este problema.

En lo que sigue, puesto que ya no aporta información relevante, se obviará la representación de ambas velocidades angulares, las cuales fueron introducidas con el fin de mostrar la mejora que supone en el sistema de control de actitud un control leve en ellas.

4.4 Modelo de MPC con saturaciones, corrección en permanente y control en $u, \gamma, q, \beta, \phi$ y p

- Variables longitudinales

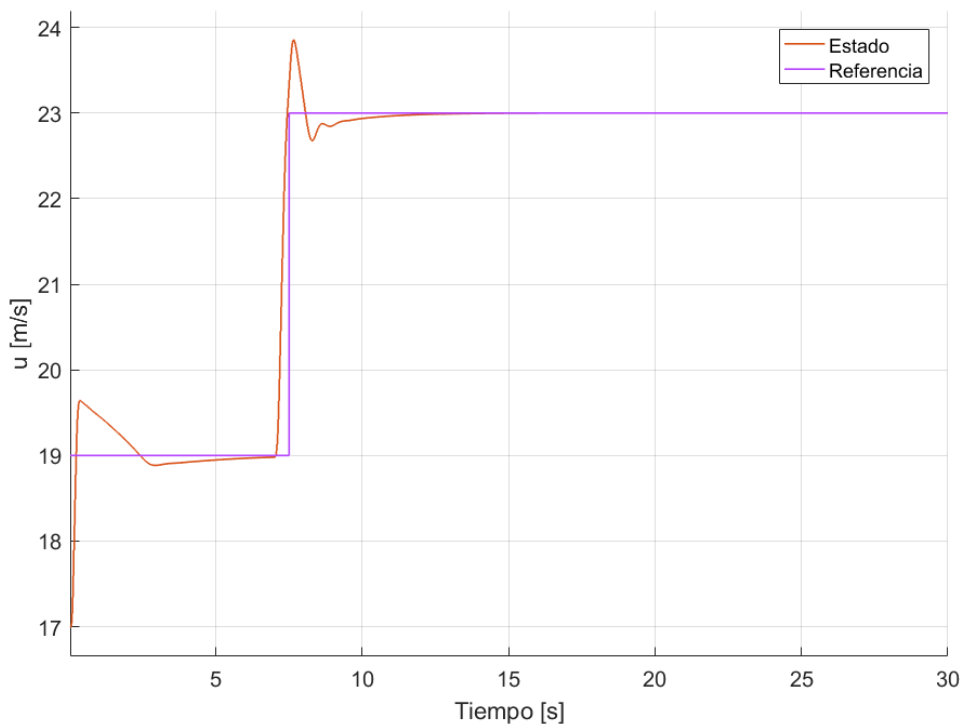


Figura 4.19 Caso 4, velocidad aerodinámica en el eje x.

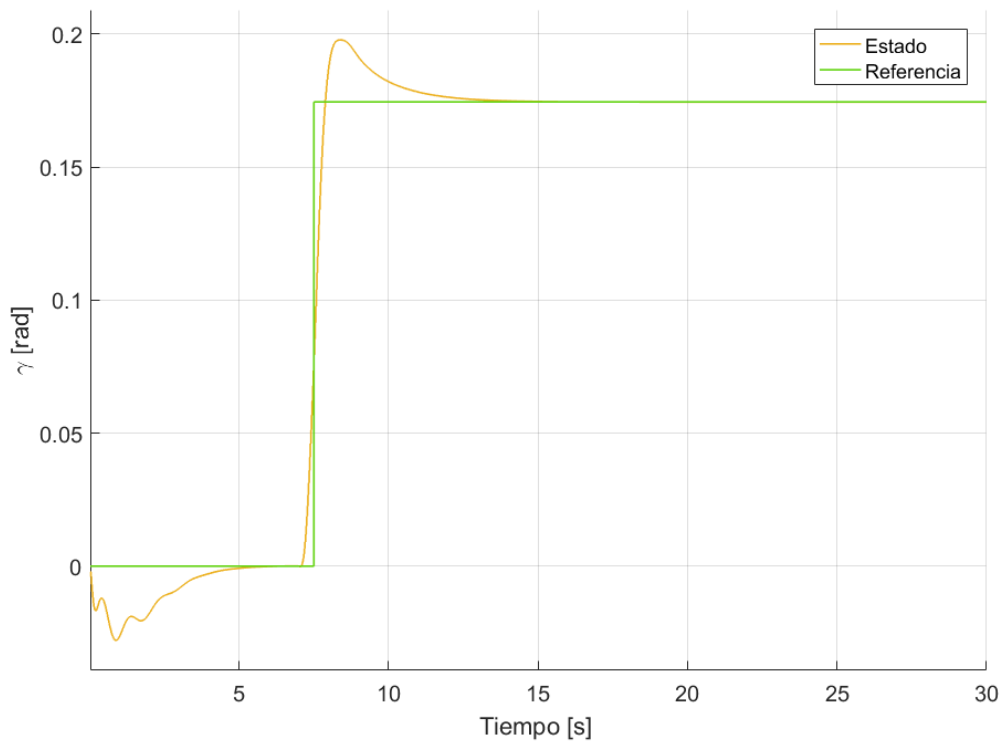


Figura 4.20 Caso 4, ángulo de trayectoria.

- **Variables laterales-direccionales**

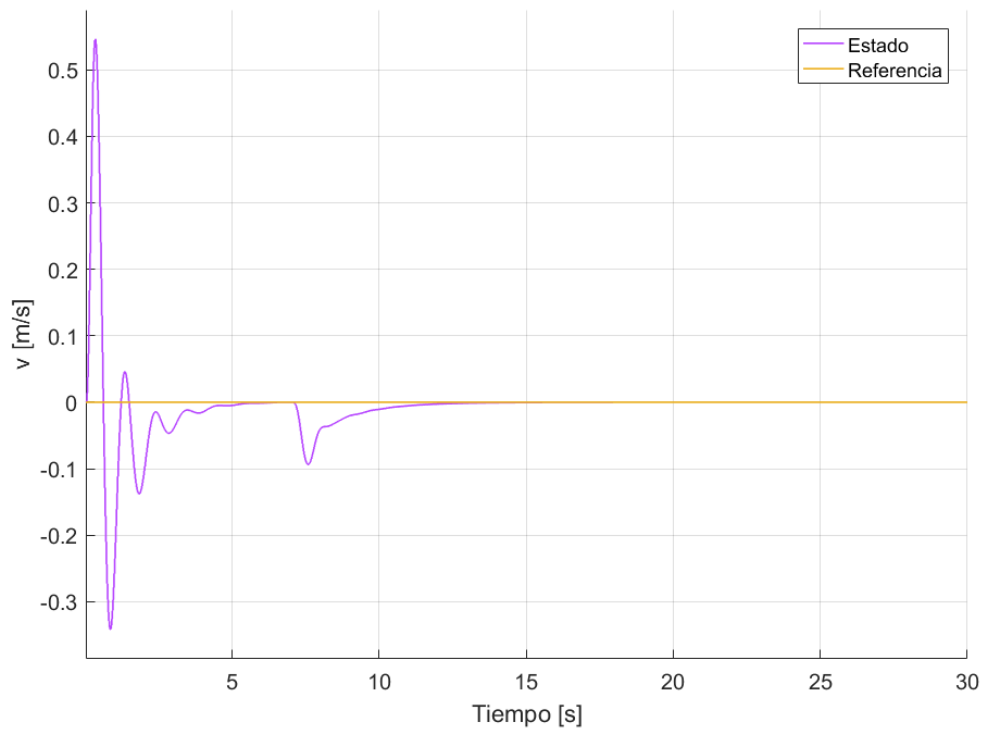


Figura 4.21 Caso 4, velocidad aerodinámica en el eje y.

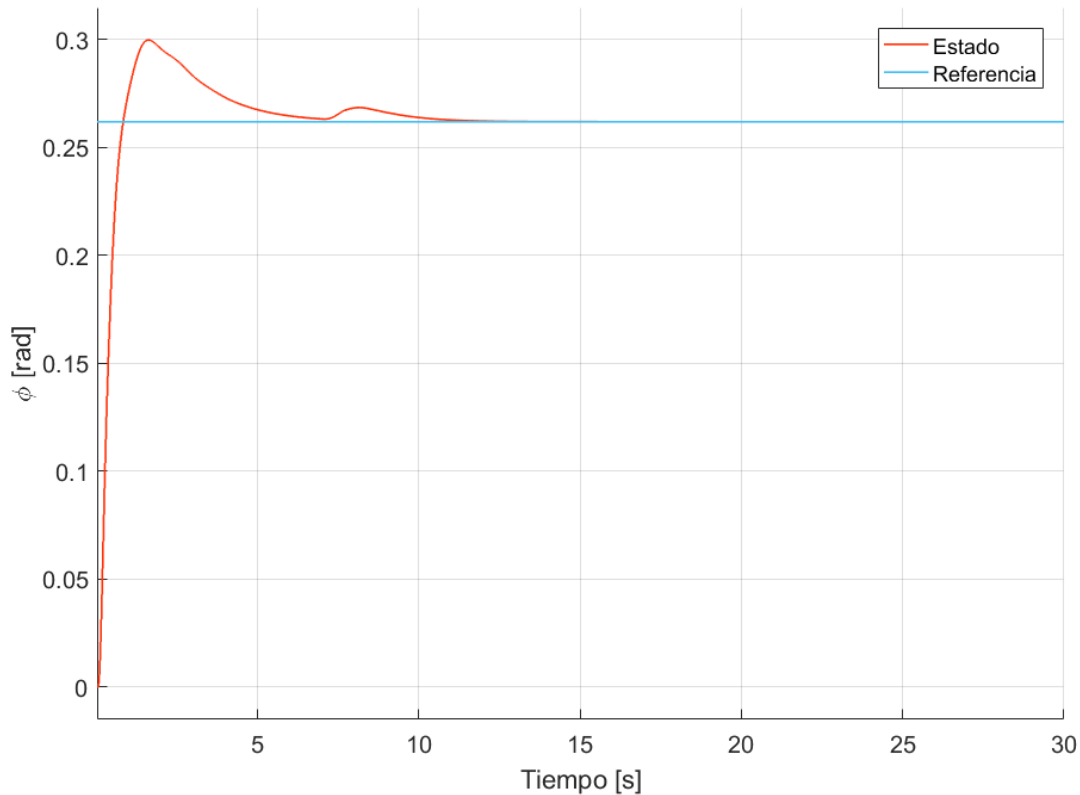


Figura 4.22 Caso 4, ángulo de balance.

Finalmente, con la implementación de ambas correcciones conjuntas se aprecia una importante mejora de los resultados obtenidos en cuanto a comportamiento y respuesta de la aeronave. De esta manera se han alcanzado los siguientes hitos

- Mejorar la estabilidad de la aeronave, eliminando sobre-oscilaciones del sistema con la introducción del control leve en las variables p y q .
- Alcanzar la referencia en régimen permanente eliminando el error asociado mediante una redefinición de la referencia que recibe el controlador.
- Evitar respuestas bruscas mediante la introducción de las restricciones en el problema de optimización, lo cual equivale a introducir un controlador con una formulación que tiene en cuenta las posibles saturaciones que puedan aparecer.

Todo ello puede verse sin más que realizar una comparativa entre las primeras simulaciones (Figura 4.1, Figura 4.2, Figura 4.4 y Figura 4.5) y las últimas (Figura 4.19, Figura 4.20, Figura 4.21 y Figura 4.22).

Para finalizar, una vez terminado el proceso comparativo de las diferentes etapas por las que ha pasado el proyecto, se expone una simulación completa, es decir, con todas las modificaciones incluidas, en la que se ha llevado al controlador a un estado más complejo que los anteriores. Esta representación tiene como objetivo demostrar el correcto funcionamiento del MPC ante situaciones más límites que las anteriores, para lo cual se han modificado las referencias y el tiempo de simulación se ha aumentado a 100 segundos.

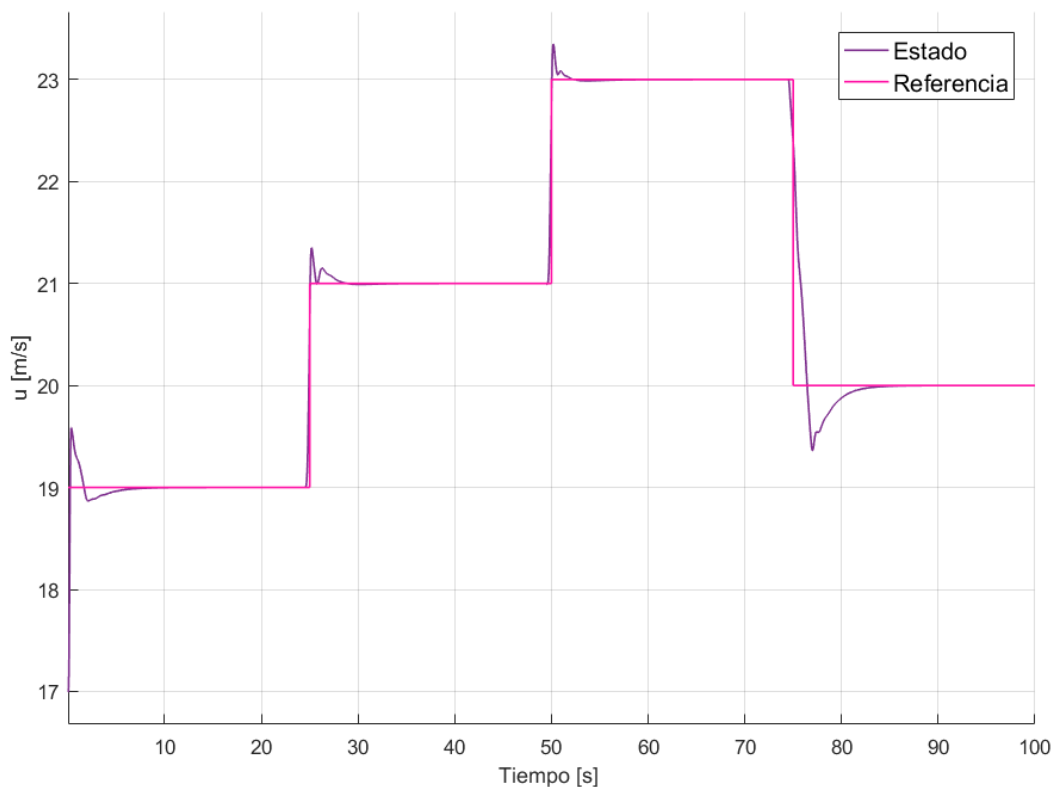


Figura 4.23 Simulación completa, velocidad aerodinámica en el eje x.

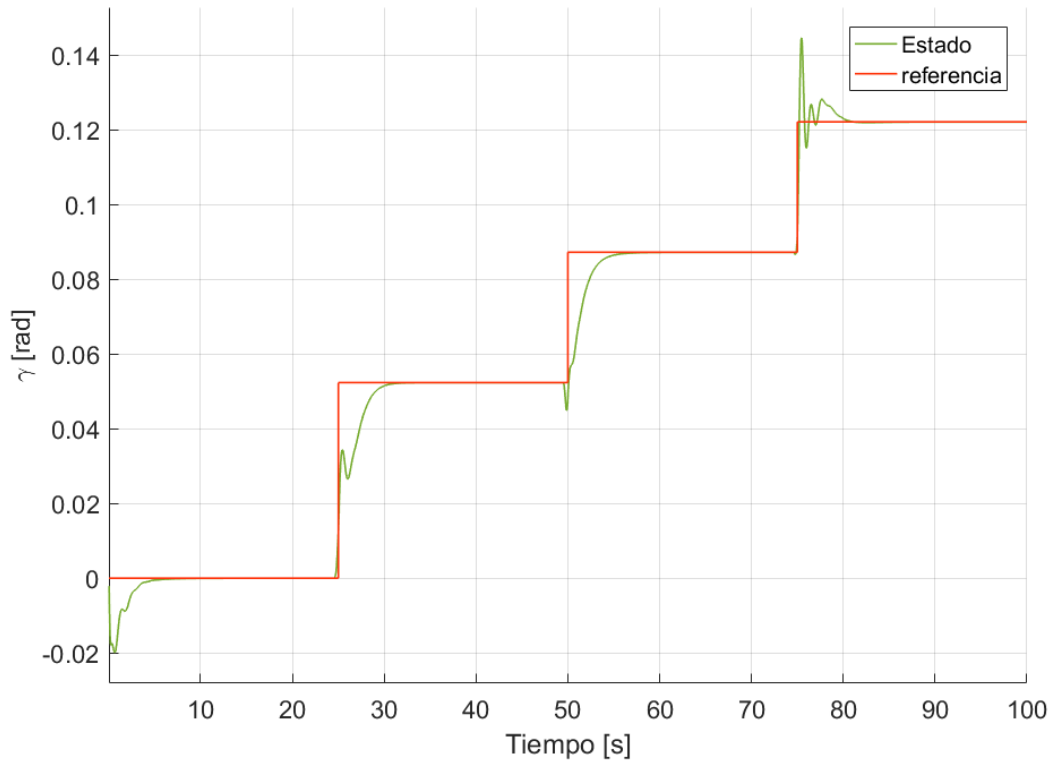


Figura 4.24 Simulación completa, ángulo de trayectoria.

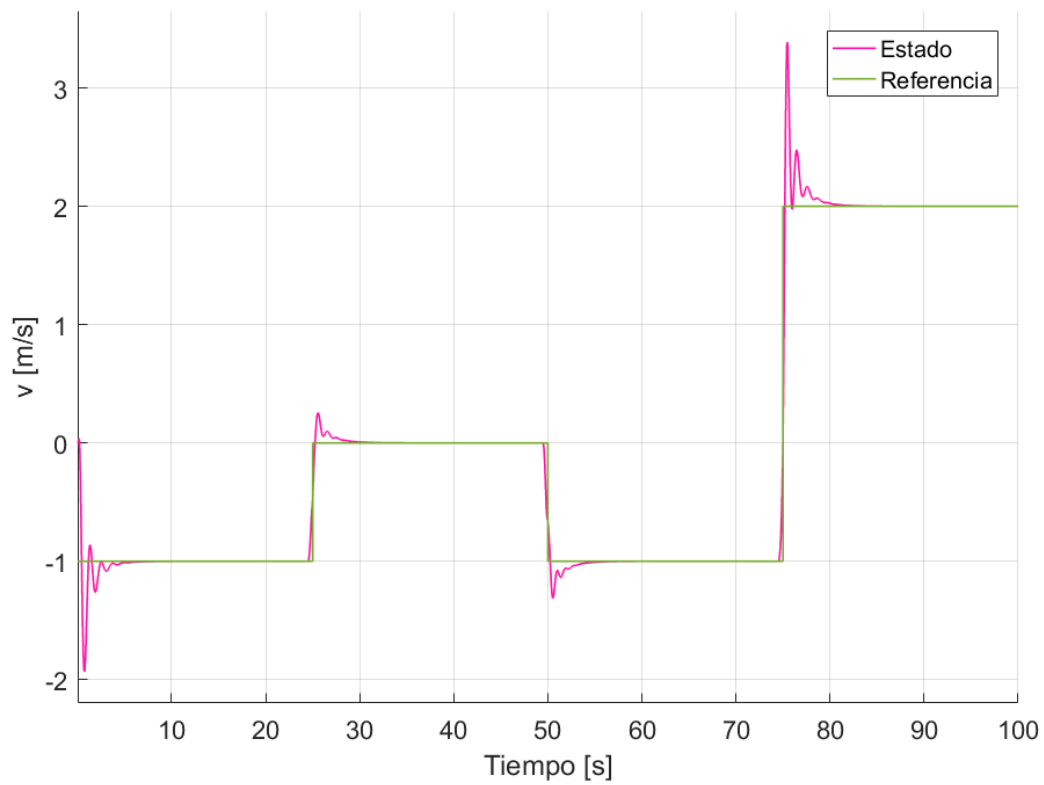


Figura 4.25 Simulación completa, velocidad aerodinámica en el eje y.

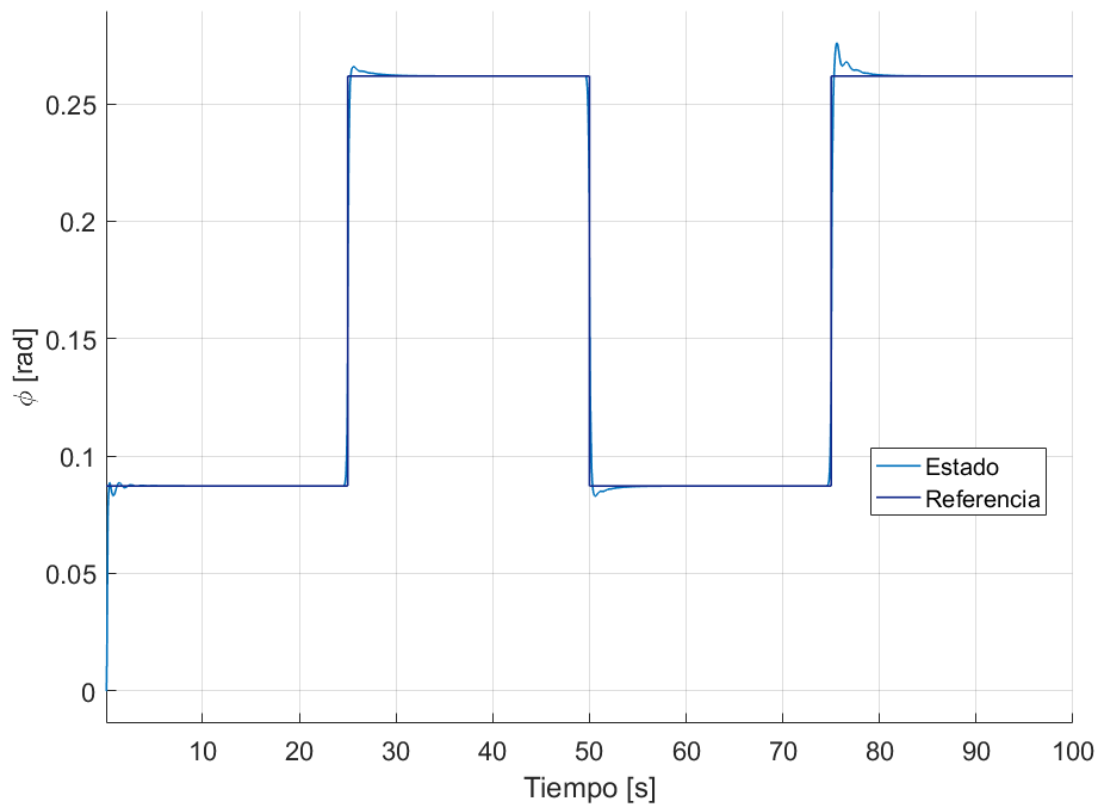


Figura 4.26 Simulación completa, ángulo de balance.

4.5 Comparativa entre quadprog y corrección de saturaciones. Tiempos de computación.

A lo largo de la presente sección se va a realizar una comparativa de los resultados obtenidos con la resolución del problema de optimización mediante la función quadprog de Matlab y la corrección de saturaciones propuestas para un máximo de 15 iteraciones. De esta manera, se mostrarán tanto las diferencias existentes entre las soluciones de ambos métodos como los tiempos de cálculo de cada uno de ellos.

Se comenzará con una comparación de ambas soluciones, para lo cual se superponen, para cada una de las variables, los resultados obtenidos de ambos métodos en una única gráfica.

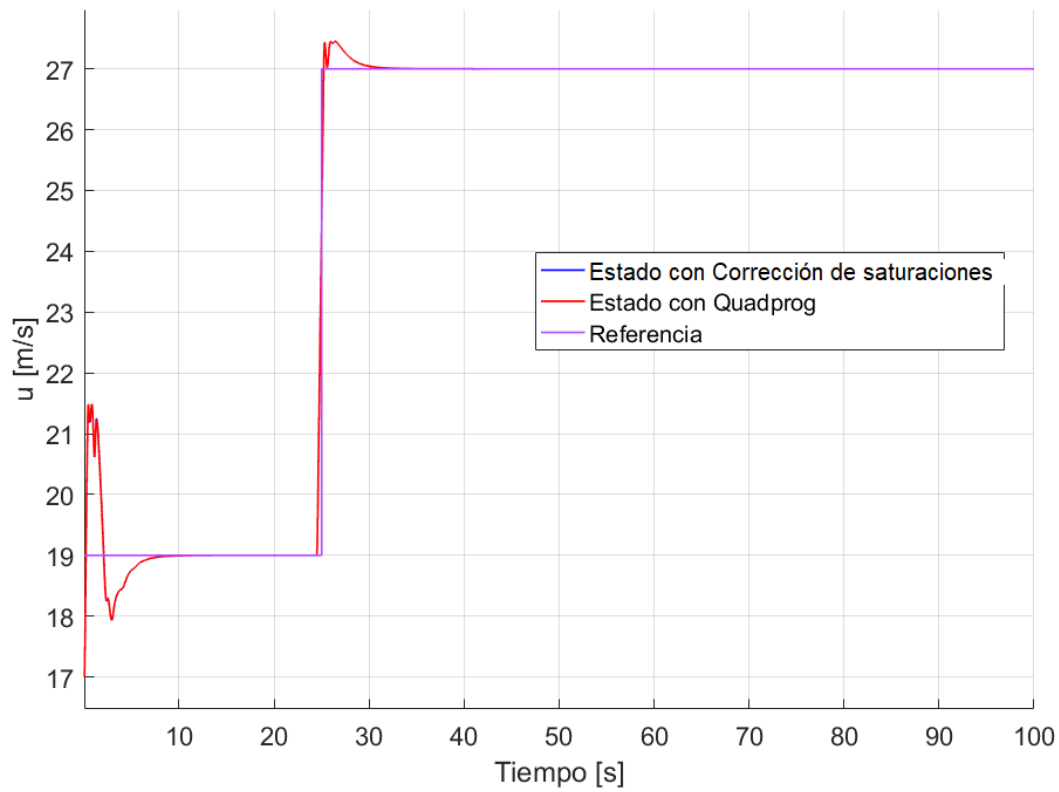


Figura 4.27 Caso 5, velocidad aerodinámica en el eje x.

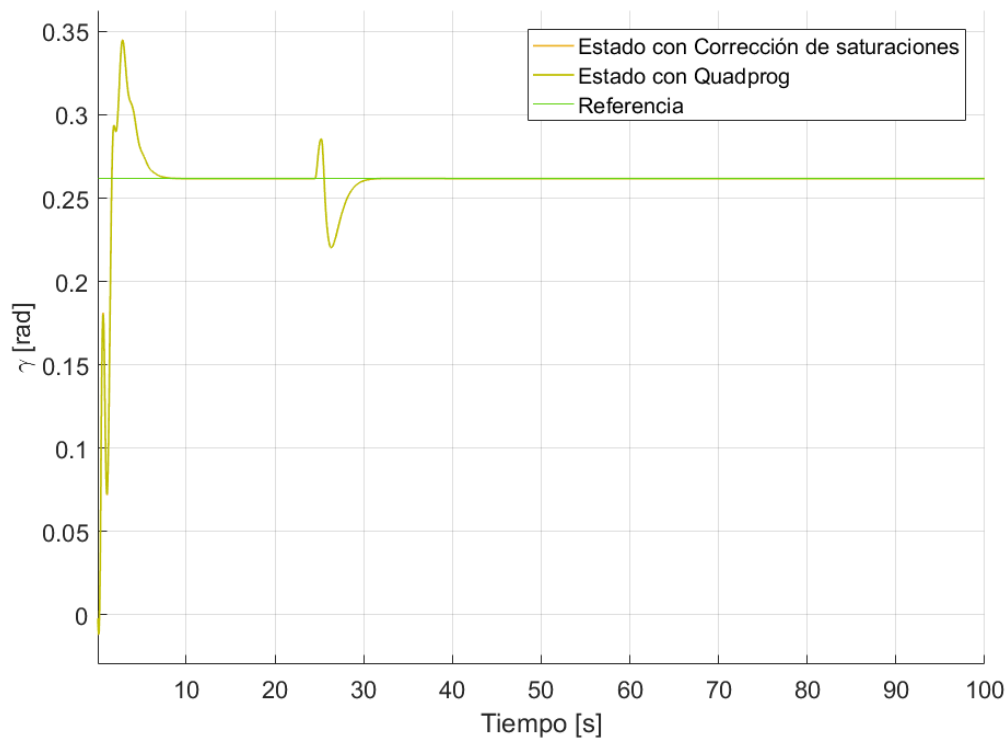


Figura 4.28 Caso 5, ángulo de trayectoria.

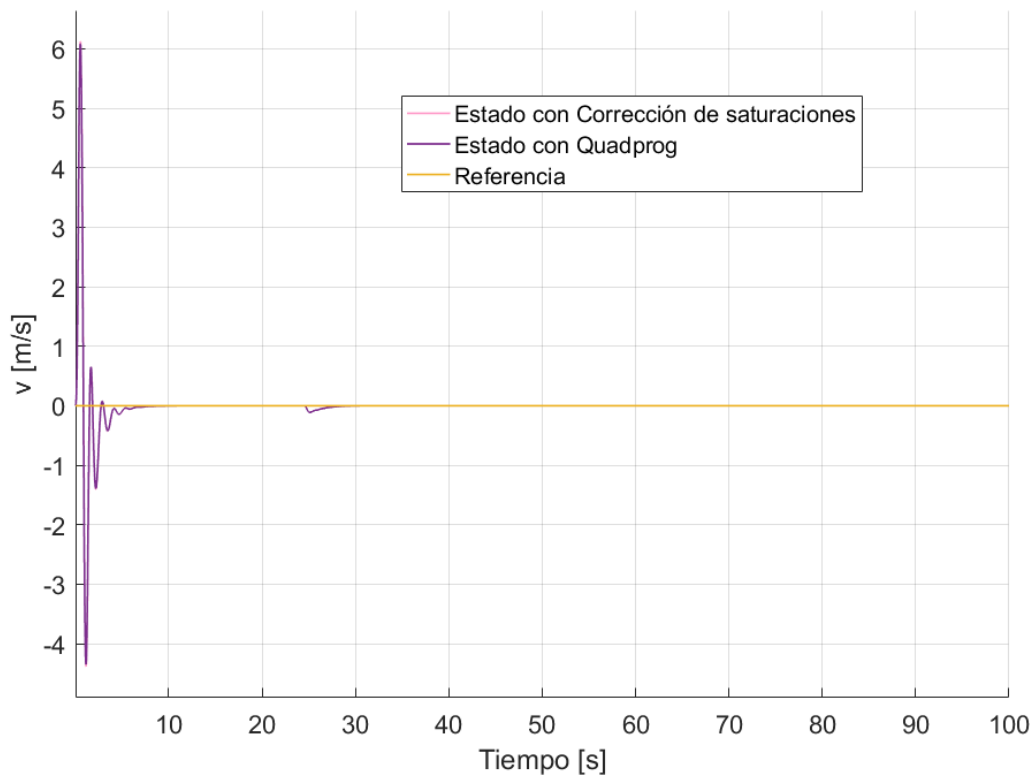


Figura 4.29 Caso 5, velocidad aerodinámica en el eje y.

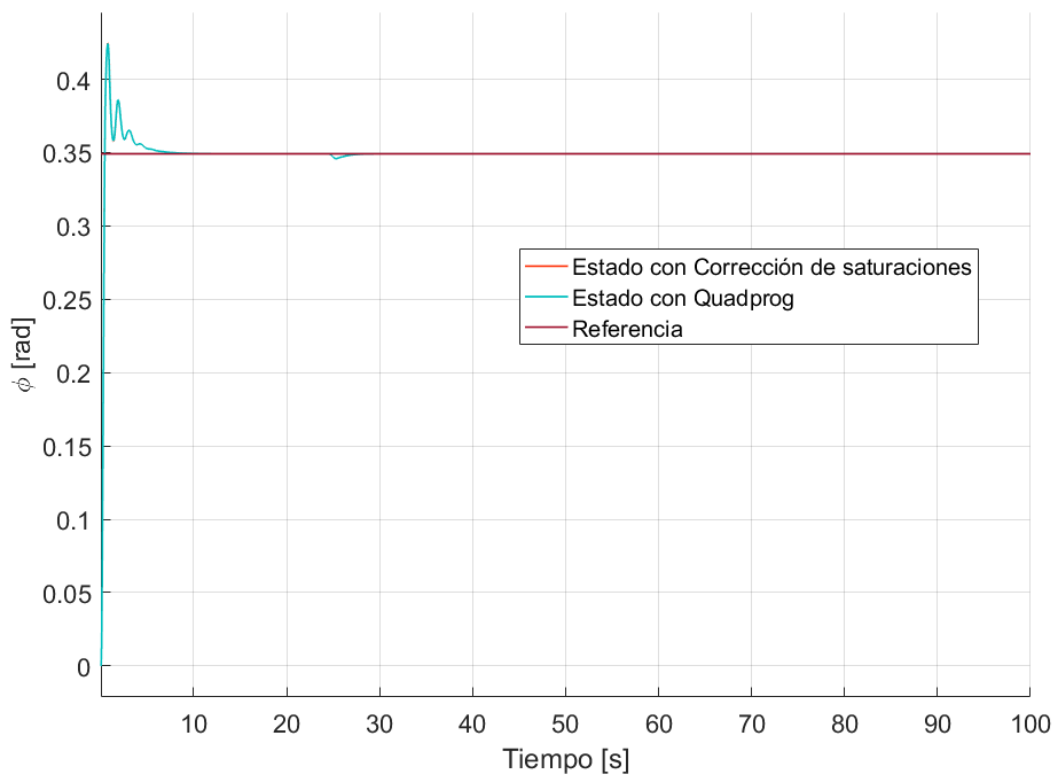


Figura 4.30 Caso 5, ángulo de balance.

Como puede comprobarse en las gráficas adjuntas (Figura 4.27, Figura 4.28, Figura 4.29 y Figura 4.30) las diferencias existentes entre utilizar la función `quadprog` de Matlab y el método desarrollado en este trabajo son mínimas (Véase ampliación en Figura 4.31 y Figura 4.32).

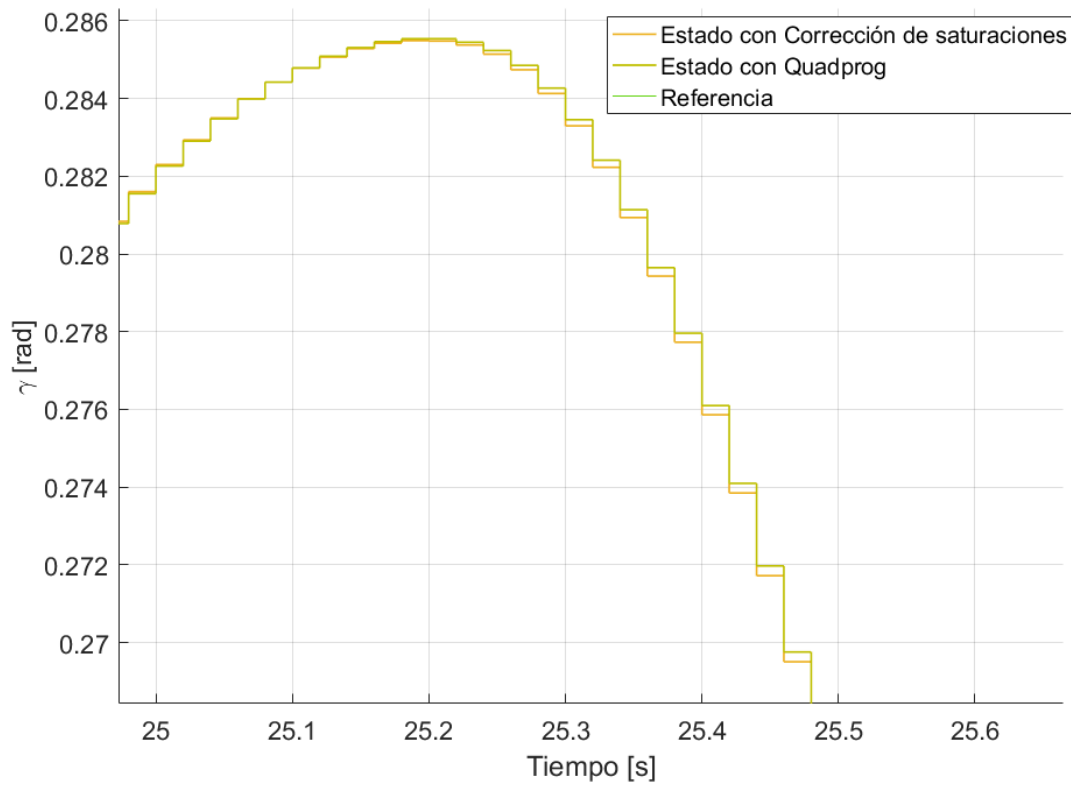


Figura 4.31 Zoom ángulo de trayectoria.

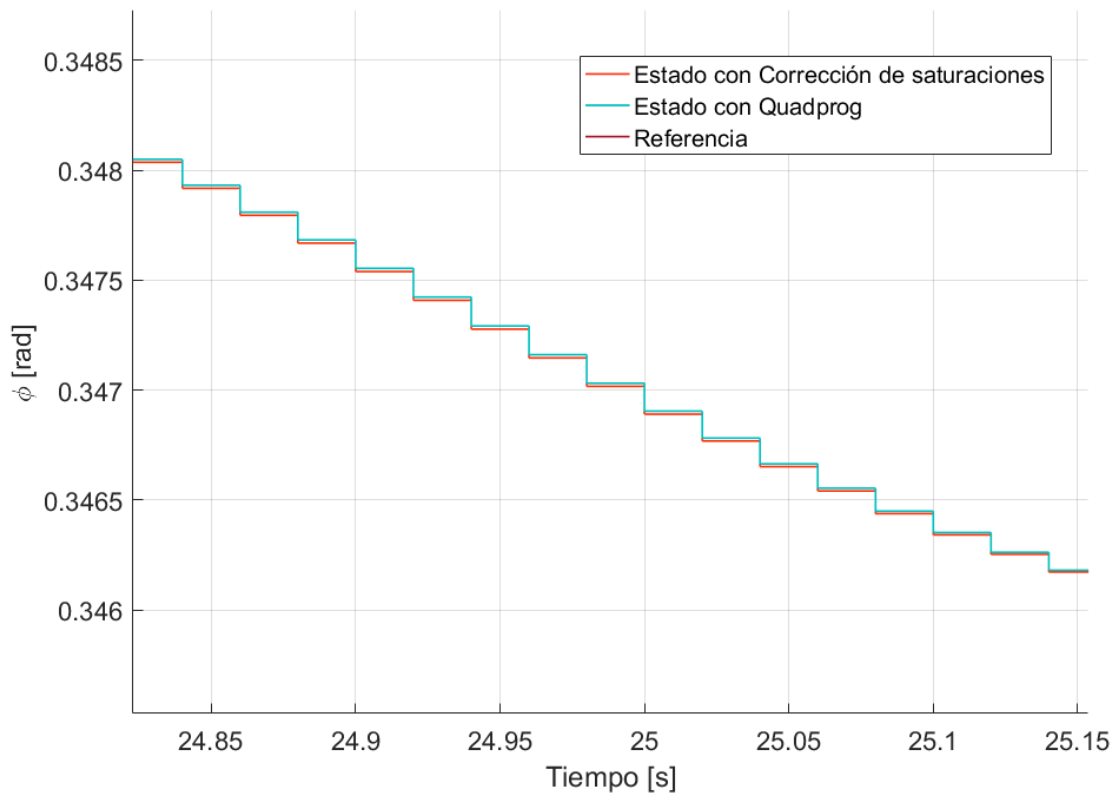


Figura 4.32 Zoom ángulo de balance.

Realizando una simulación en la que se lleva a la aeronave a puntos cercanos a la inestabilidad comienzan a aparecer ciertas diferencias entre ambos métodos (Figura 4.33, Figura 4.34, Figura 4.35 y Figura 4.36). La principal razón de ello se debe a que, al resolver el problema de optimización con restricciones completo, la función quadprog presenta una mayor estabilidad en la solución que el método diseñado, el cual, como ya se comentó, ofrece como solución un pseudo-óptimo del problema.

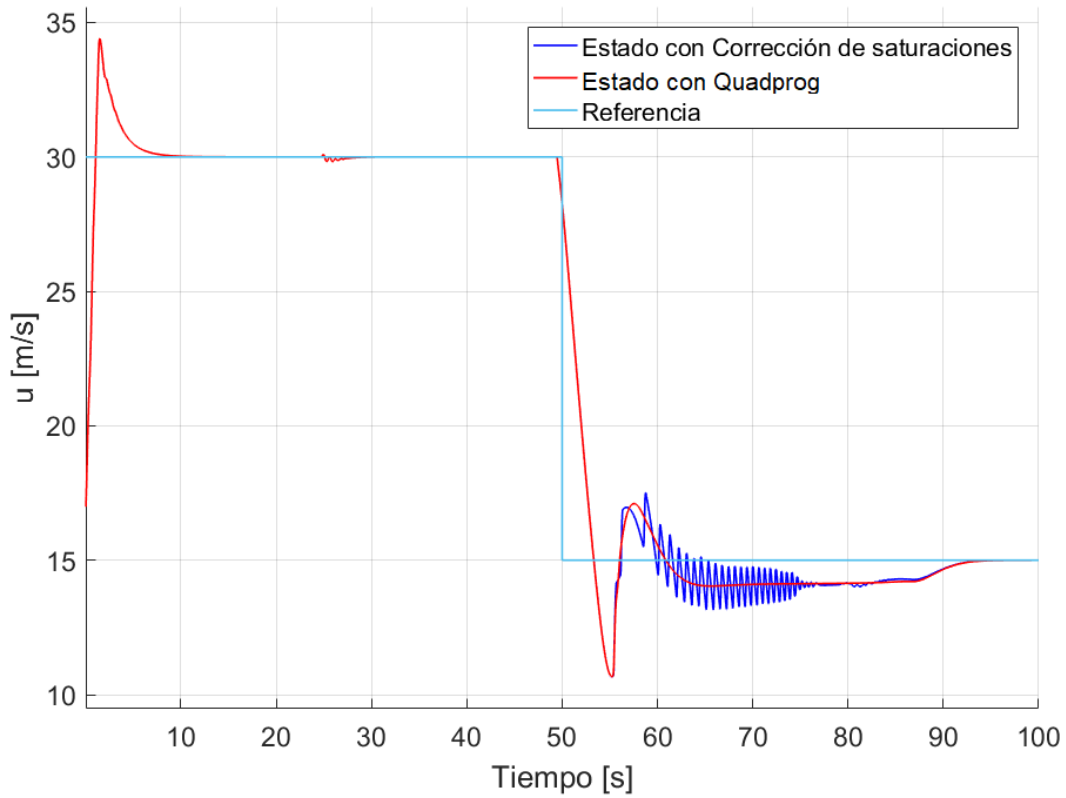


Figura 4.33 Comparativa de métodos, velocidad aerodinámica en el eje x.

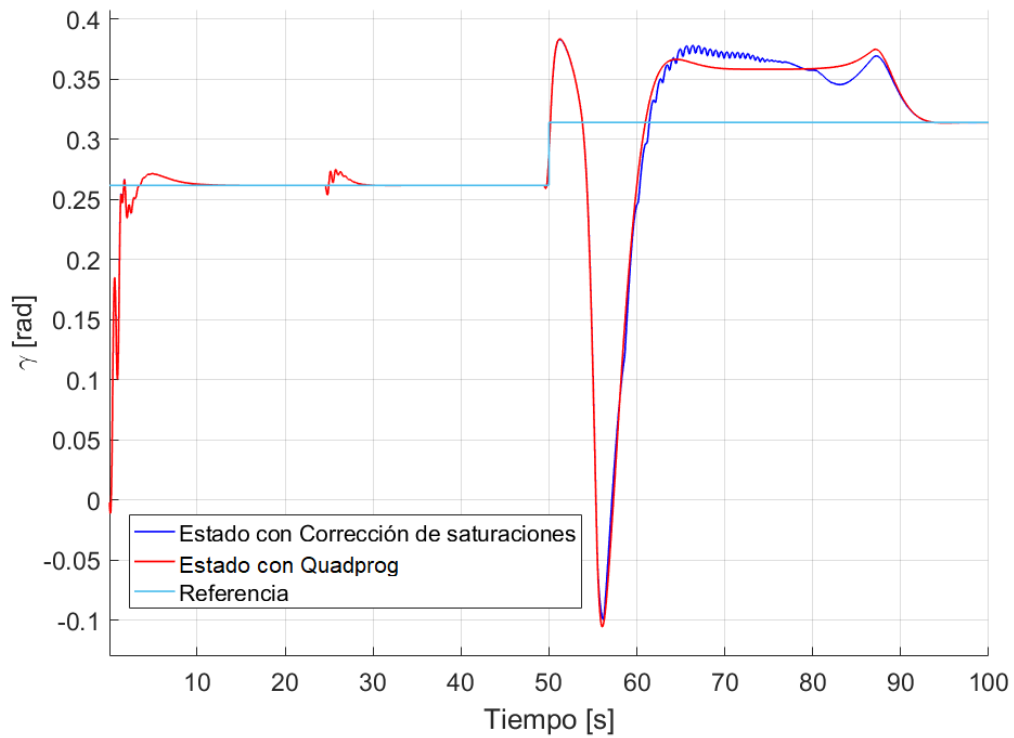


Figura 4.34 Comparativa de métodos, ángulo de trayectoria.

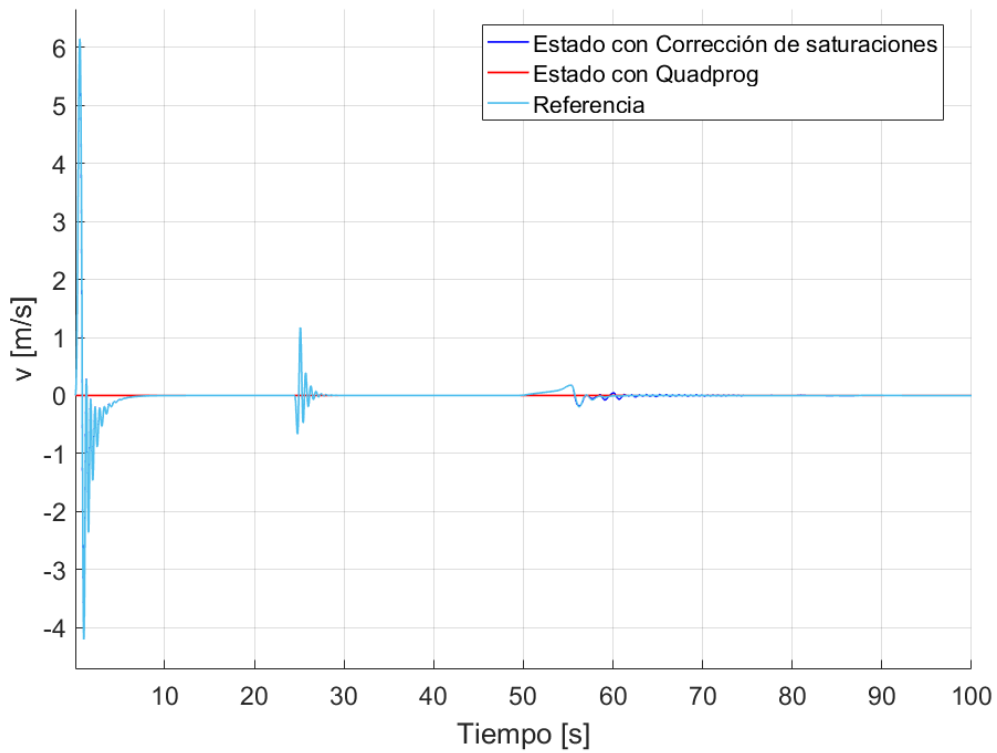


Figura 4.35 Comparativa de métodos, velocidad aerodinámica en el eje y.

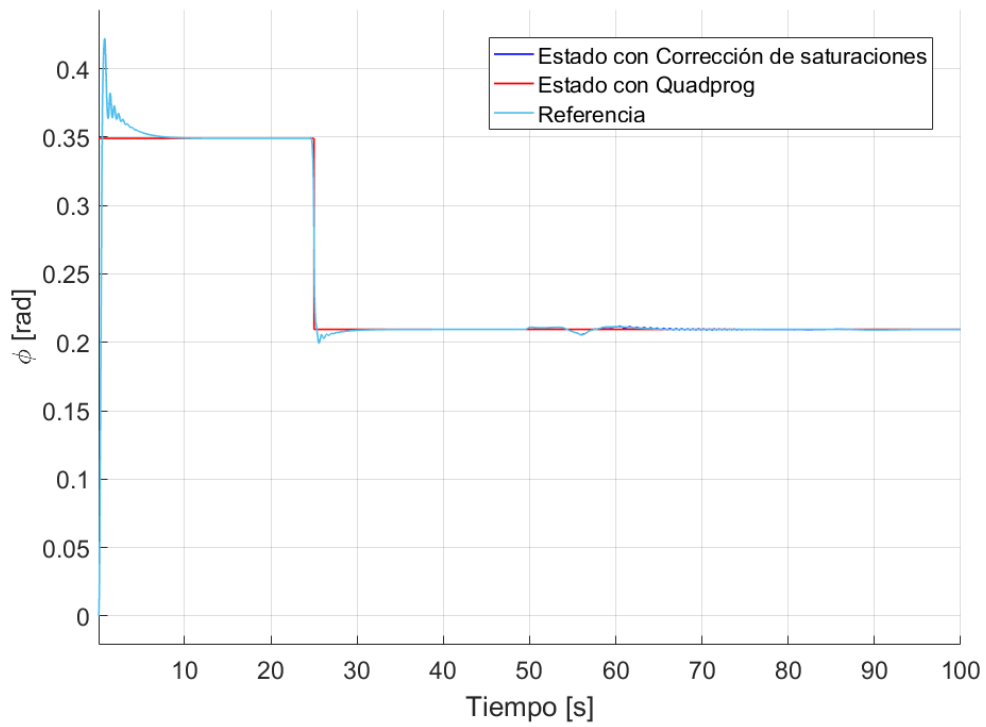


Figura 4.36 Comparativa de métodos, ángulo de balance.

Finalmente, con respecto a los tiempos de computación se han obtenido los siguientes resultados

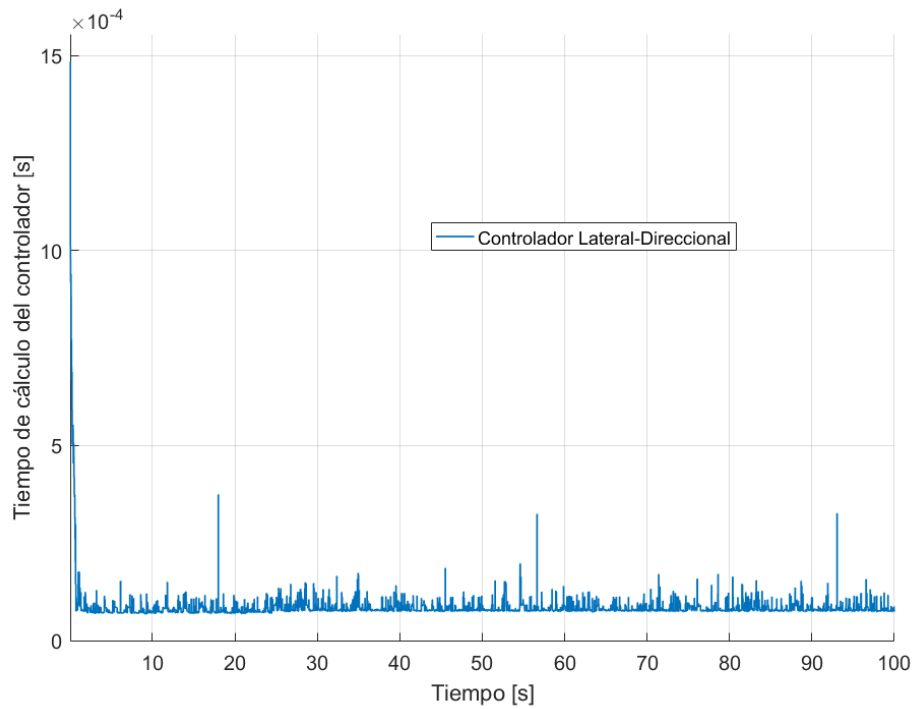


Figura 4.37 Tiempos de cálculo para el método diseñado, lateral-direccional.

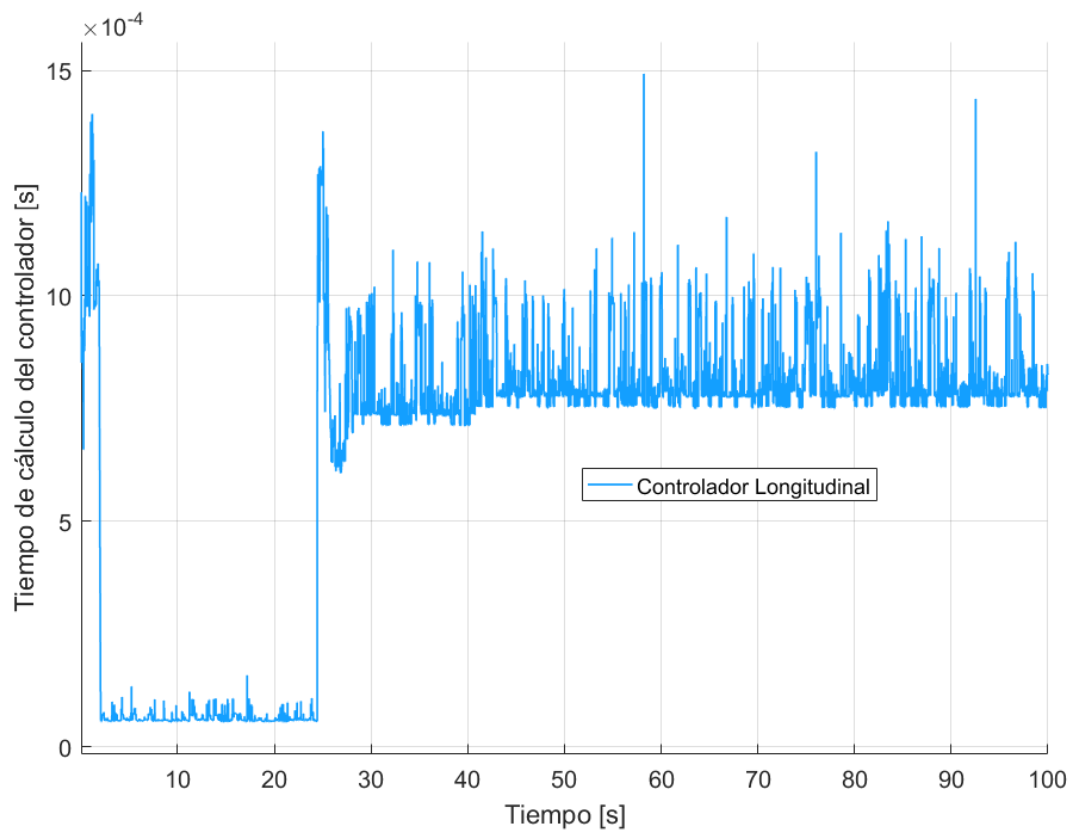


Figura 4.38 Tiempos de cálculo para el método diseñado, longitudinal.

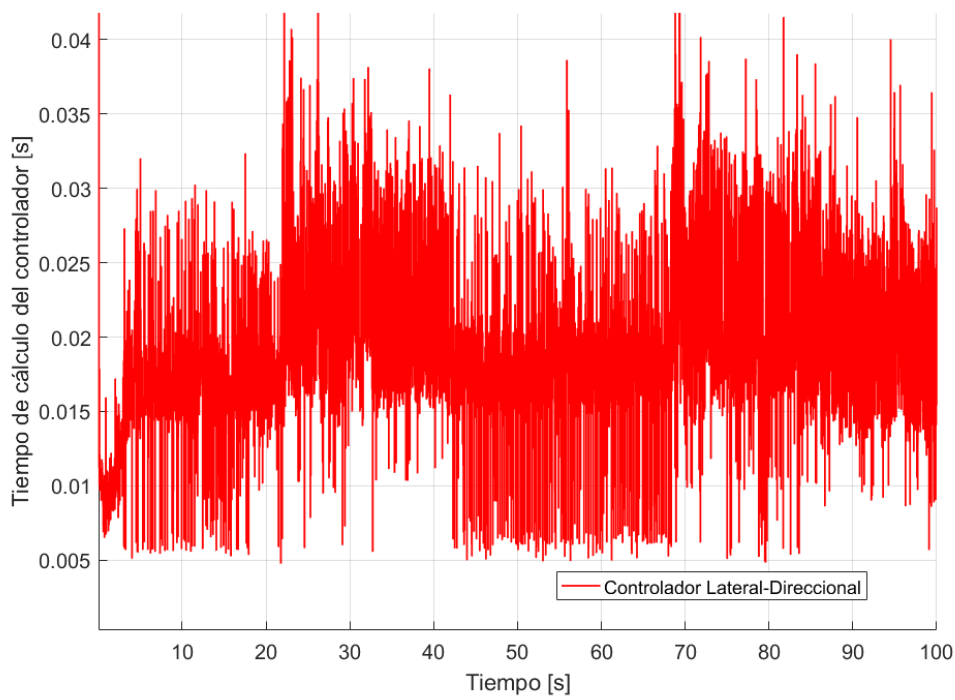


Figura 4.39 Tiempos de cálculo del quadprog, lateral-direccional.

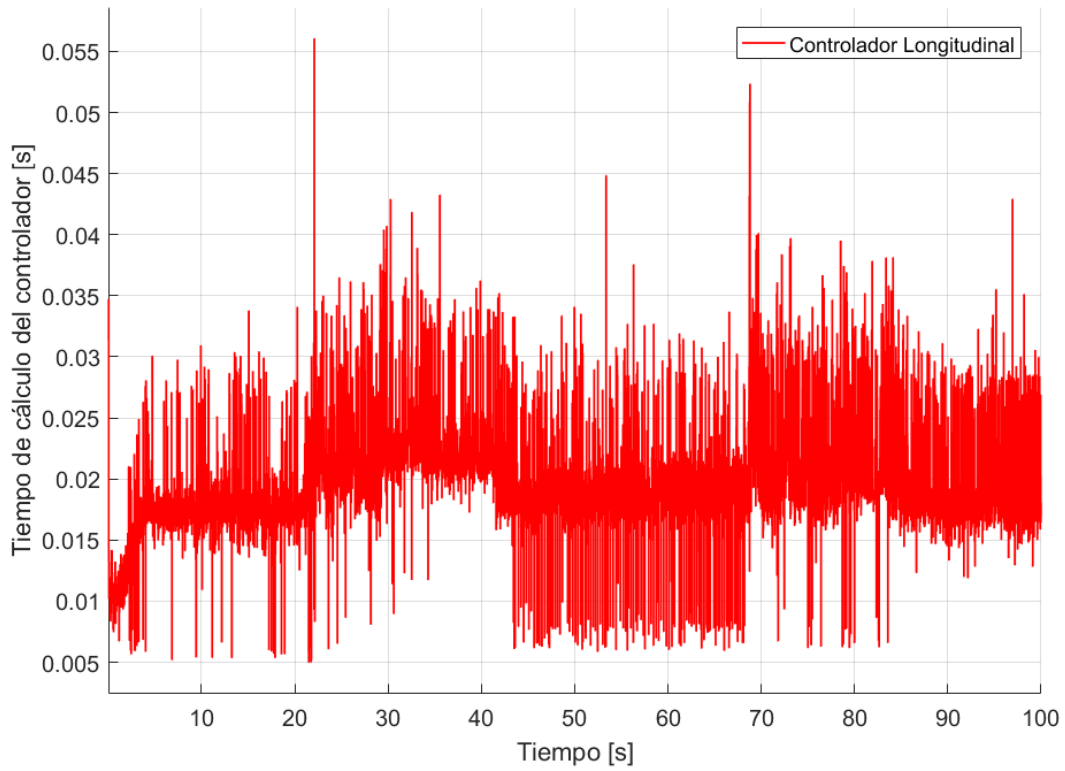


Figura 4.40 Tiempos de cálculo del quadprog, longitudinal.

Para finalizar, en base a los resultados expuestos durante el desarrollo de esta sección, se pueden sacar las siguientes conclusiones

- El tiempo de cálculo del método diseñado en este trabajo es bastante inferior al que necesita la función quadprog.
- La suma de los tiempos de cálculo de ambos controladores para el caso del método de saturaciones es inferior al límite impuesto por el intervalo de muestreo.
- La suma de los tiempos de cálculo de ambos controladores para el caso de la función quadprog supera ampliamente el margen permitido.
- Las diferencias existentes entre ambos métodos son mínimas exceptuando situaciones cercanas a la inestabilidad, las cuales se entienden que no se van a producir en condiciones habituales de vuelo.

Por todo ello se puede afirmar que el método de corrección de saturaciones diseñado en este trabajo cumple sobradamente con las condiciones impuestas, a la par que presenta un comportamiento muy similar al método de optimización con restricciones completo con un menor tiempo de computación y gasto de memoria y, por consiguiente, es perfectamente apto para su implementación en el controlador.

5 CONCLUSIONES Y LÍNEAS FUTURAS

Tras el diseño del modelo de MPC, la introducción de las modificaciones en el mismo, su implementación en el entorno de simulación de partida, la consecución de los resultados respectivos y el análisis comparativo realizado, se pretende dar el cierre a este proyecto ofreciendo las conclusiones obtenidas a lo largo del desarrollo del mismo y las líneas futuras de trabajo que podrían llevarse a cabo.

El objetivo fundamental de este proyecto consistía en el diseño de un sistema de control de actitud basado en los métodos de control predictivo para su posterior integración dentro de un sistema general que simula el comportamiento real de un UAV. En este sentido, el proyecto se ha centrado en el diseño de un modelo de controlador básico al cual se le han ido añadiendo una serie de mejoras o modificaciones para, bien mejorar el comportamiento y respuesta del MPC, o bien para adaptar el modelo y permitir su integración posterior en el sistema global mencionado con anterioridad. Finalmente, tras este proceso, se han obtenido una serie de resultados que han permitido sentar las bases de las conclusiones que se exponen en este capítulo.

De esta forma, siguiendo la estructura de desarrollo del proyecto, se van a analizar las diferentes etapas por las que ha pasado el diseño del controlador, extrayendo las conclusiones de cada una de ellas por separado. Del mismo modo, se exponen líneas de trabajo futuras que podrían derivarse de cada apartado y un objetivo final del proyecto.

- **Diseño del modelo de MPC básico**

El diseño del MPC se ha basado en un modelo de 6 G.D.L del avión linealizado en torno a un punto o condición de referencia, lo cual ha supuesto que se haya tenido que elegir una formulación linealizada para el modelo de controlador. Esto permite reducir el sistema dinámico con ecuaciones diferenciales original a un problema con ecuaciones incrementales discretizadas en el tiempo. Todo ello, unido a las diversas simplificaciones introducidas a lo largo del desarrollo del modelo, implica una reducción en la precisión del mismo y, consecuentemente, en la respuesta del sistema. Pese a ello, como ha podido

comprobarse, la aplicación del conjunto de simplificaciones ha resultado satisfactoria, obteniéndose buenos resultados en las simulaciones realizadas.

Por otro lado, la introducción de un modelo de fuerzas aerodinámicas y propulsivas simplificado ha permitido desacoplar la dinámica lateral-direccional de la longitudinal, lo cual ha supuesto una reducción en la complejidad del modelo que ha permitido el diseño de dos controladores independientes, sin una pérdida considerable de la precisión en la solución.

En ese sentido, una tarea interesante que podría llevarse a cabo en el futuro sería la eliminación de algunas de las hipótesis realizadas, para lo cual sería necesaria una reformulación del modelo, con el objetivo de realizar un análisis comparativo con el modelo propuesto en esta disertación y valorar la mejora obtenida frente a la complejidad que supone la introducción de una formulación más exacta.

Por último, una vez finalizado y simulado el modelo se comprueba que, a pesar de los buenos resultados obtenidos, se hace necesaria la introducción de una serie de correcciones que mejoren el comportamiento del mismo.

- **Introducción de la corrección del error en permanente**

Con el objetivo de reducir el error que aparece en el controlador cuando se alcanza el régimen permanente se introduce una corrección que permite al MPC eliminar la diferencia existente entre el estado de la aeronave y la referencia deseada en régimen permanente. En ese sentido, se ofrece una solución que se asemeja a la introducción de un controlador integral en el modelo y que, claramente, elimina el mencionado error.

Sin embargo, la implementación de esta solución conlleva un empeoramiento del comportamiento de la aeronave en el régimen transitorio. En este sentido, se plantea como línea futura de este trabajo la mejora de este sistema de manera que se evite la mencionada discrepancia.

- **Introducción de las restricciones en el problema de optimización**

Junto con la corrección del error en permanente aparece la necesidad de incluir los efectos de las saturaciones dentro de la formulación del MPC. La razón de ello se fundamenta, principalmente, en la diferencia que surge entre la actuación del controlador y la real cuando se le demandan a la aeronave cambios bruscos en su estado que provocan la saturación de las variables de control. Esta diferencia puede ocasionar oscilaciones importantes en la respuesta de la misma o incluso la entrada en inestabilidad del sistema.

Por ello se ha propuesto en este proyecto un modelo de controlador que resuelve un problema de optimización con restricciones en el que se tienen en cuenta las saturaciones de la aeronave. Para ello, con la intención de conseguir tiempos de cálculo pequeños, se aplica una formulación que no ofrece la solución óptima del problema con restricciones, sino que ofrece el resultado de un problema de optimización reducido que se ha designado como pseudo-óptimo. Las simulaciones llevadas a cabo demuestran que este resultado es muy similar al obtenido mediante la resolución del problema completo y que supone una reducción del tiempo de cálculo importante, permitiendo su integración dentro de un modelo real con restricciones computacionales en tiempo y memoria. De hecho, el tiempo de cálculo tan pequeño permite abrir una línea futura de trabajo en la cual se realice una optimización de este sistema para mejorar su respuesta ante situaciones o estados complejos.

- **Adaptación de las variables de estado**

Como última modificación al modelo de MPC original se realiza una transformación en las variables de estado del sistema, de manera que sea posible la integración futura con el modelo de sistema global desarrollado en el capítulo 2 de esta memoria.

En este sentido, como desarrollo futuro, se propone la integración del MPC en el sistema general junto con el sistema de guiado diseñado en la tesis doctoral que ha servido de base para este proyecto.

- **Conclusión global**

Como objetivo final del proyecto se propone que, tras la consecución de todos los desarrollos futuros propuestos, el modelo sea implementado en una aeronave para, de esta manera, poder analizar y poner a prueba el controlador dentro de una situación real en la que la incertidumbre gana mucho protagonismo.

Finalmente, para concluir esta memoria, a pesar de la cantidad de estudios e investigaciones relativas a los UAVs, se trata de un campo muy abierto en la actualidad, especialmente en el ámbito de los sistemas de control predictivo. En ese sentido, se ha permitido introducir en el desarrollo del proyecto las nuevas líneas de trabajo relativas a los modelos de control predictivo que se están implementando en la actualidad. Además, tal y como ha quedado patente, existen muchas opciones sobre las que se puede actuar.

Por último, las diferentes simulaciones que se han llevado a cabo en el proyecto han permitido al autor del mismo comprobar el potencial de aplicación que presentan estos sistemas de control de actitud en el ámbito aeronáutico, sobre todo con las nuevas tecnologías y microprocesadores de gran capacidad de computación que están apareciendo en el mercado.

REFERENCIAS

- [1] F. R. Gavilán Jiménez, «Sistemas de Control y Guiado para Vehículos Aéreos No Tripulados: Diseño de Algoritmos y Sistemas Embarcados.» Sevilla, Noviembre, 2012.
- [2] F. R. Gavilán Jiménez, Ecuaciones Generales de la Dinámica del Avión, 2015.
- [3] F. R. Gavilán Jiménez, Derivadas de Estabilidad, 2015.
- [4] B. Pamadi, Performance, Stability, Dynamics, and Control of Airplanes, 1998.
- [5] B. L. Stevens y F. L. Lewis, Aircraft Control and Simulation, 2003.
- [6] E. B. y L. D. Reid, Dynamics of Flight, Stability and Control, 1996.
- [7] R. Slattery y Y. Zhao, Trajectory synthesis for air traffic automation, 1997.
- [8] A. I. Propoi, Use of linear programming methods for sunthesizing sampled-data automatic sysems., 1963.
- [9] D. Simon, «Model Predictive Control in Flight Control Design, Stability and Reference Tracking,» 2014.
- [10] M. Cannon, C21 Model Predictive Control, 2016.
- [11] O. Vila Rovira, Modelización de aeronaves no tripuladas con Simulink, Julio de 2011, pp. 4.1 - 4.23.
- [12] M. Cannon, Introduction, Model Predictive Control, 2016.
- [13] M. Cannon, Prediction and optimization, Model Predictive Control, 2016.
- [14] M. Cannon, Closed loop properties of MPC, Model Predictive Control, 2016.

-
- [15] M. Cannon, Robustness to disturbances, Model Predictive Control, 2016.
- [16] C. Jones y M. Zeilinger, «Introduction,» de *Real - Time Model Predictive Control*, 2016.
- [17] C. Jones y M. Zeilinger, «Lecture 1,» de *Real - Time Model Predictive Control*, 2016.
- [18] C. Jones y M. Zeilinger, «Lecture 2,» de *Real - Time Model Predictive Control*, 2016.
- [19] C. Jones y M. Zeilinger, «Lecture 3,» de *Real - Time Model Predictive Control*, 2016.
- [20] C. Jones y M. Zeilinger, «Lecture 4,» de *Real - Time Model Predictive Control*, 2016.
- [21] University of Sheffield;, «Department of Automatic Control and Systems Engineering,» J A Rossiter, 2017. [En línea]. Available: https://www.sheffield.ac.uk/acse/staff/jar/mpc_introduction. [Último acceso: 10 2017].
- [22] Aertec Solutions, «Aertec Solutions,» Aertec Solutions, 5 2017. [En línea]. Available: <http://www.aertecsolutions.com/2017/02/06/los-primeros-sistemas-de-guiado/>. [Último acceso: 7 2017].

Apéndice A. Montaje matrices F y G

- Longitudinal

```
function [F_1,G_1]=longitudinal_vfin(A_1,B_1,C,Np)

[f1_1,c1_1]=size(C*A_1);
[f2_1,c2_1]=size(C*B_1);

%Montamos matrices iniciales de F y G que luego se rellenarán

F_1=zeros(Np*f1_1,c1_1);
G_1=zeros(Np*f2_1,Np*c2_1);

L1=1;
L2=1;
J1=1;

for k=1:Np

    % Primero creamos una matriz G con B en la diagonal

    G_1(L1:L1+f2_1-1,L2:L2+c2_1-1)=C*B_1(:,,:);

    % Después rellenamos el resto de la matriz

    if L1~=1 && L2~=1

        n1=(k-1)*f2_1+1;
        n2=1;
        cont=1;

        while cont<=k

            G_1(n1:n1+f2_1-1,n2:n2+c2_1-1)=C*A_1^(k-cont)*B_1;
            n2=n2+c2_1;
            cont=cont+1;

        end

    end

    L1=L1+f2_1;
    L2=L2+c2_1;

    F_1(J1:J1+f1_1-1,1:c1_1)=C*(A_1(:,,:))^k;
    J1=J1+f1_1;

end
```

```
end
```

- **Lateral-direccional**

```
function [F_2,G_2]=lat_direc_vfin(A_2,B_2,Np)
```

```
[f1_2,c1_2]=size(A_2);
[f2_2,c2_2]=size(B_2);
```

```
%Montamos matrices iniciales de F y G que luego se rellenarán
```

```
F_2=zeros(Np*f1_2,c1_2);
G_2=zeros(Np*f2_2,Np*c2_2);
```

```
L1=1;
L2=1;
J1=1;
```

```
for k=1:Np
```

```
    % Primero creamos una matriz G con B en la diagonal
```

```
    G_2(L1:L1+f2_2-1,L2:L2+c2_2-1)=B_2(:,:);
```

```
    % Después rellenamos el resto de la matriz
```

```
    if L1~=1 && L2~=1
```

```
        n1=(k-1)*f2_2+1;
        n2=1;
        cont=1;
```

```
        while cont<=k
```

```
            G_2(n1:n1+f2_2-1,n2:n2+c2_2-1)=A_2^(k-cont)*B_2;
            n2=n2+c2_2;
            cont=cont+1;
```

```
        end
```

```
    end
```

```
    L1=L1+f2_2;
    L2=L2+c2_2;
```

```
    F_2(J1:J1+f1_2-1,1:c1_2)=(A_2(:,:))^k;
    J1=J1+f1_2;
```

```
end
```

```
end
```


Apéndice B. Montaje matrices R y Q

```

function [R_1,R_2]=matrix_R_vfin(R_valor1,R_valor2,Np,nvu_lg,nvu_ld)

R_1=zeros(Np*nvu_lg,Np*nvu_lg);
R_2=zeros(Np*nvu_ld,Np*nvu_ld);

R_v1=[R_valor1/1;R_valor1/1]; %d_e la primera
R_v2=[1*R_valor2/1;.1*R_valor2/1];

for i=1:Np

    j=nvu_ld*i;
    k=nvu_lg*i;

    R_1(j-(nvu_lg-1):j,j-(nvu_lg-1):j)=diag(R_v1);
    R_2(k-(nvu_ld-1):k,k-(nvu_ld-1):k)=diag(R_v2);

end

end

function [Q_1,Q_2]=matrix_Q_vfin(delta_lg,delta_ld,Np,nvx_lg,nvx_ld)

Q_1=zeros(Np*nvx_lg,Np*nvx_lg);
Q_2=zeros(Np*nvx_ld,Np*nvx_ld);

p_u=5;
p_q=1e-1;
% p_w=0;
% p_theta=10;
p_gamma = 5;

p_v=1;
p_p=4e-1;
p_r=0;
p_phi=6;
p_psi=0;

for i=1:Np

    j=nvx_lg*i;
    k=nvx_ld*i;

    % Q_1(j-(nvx_lg-1):j,j-(nvx_lg-1):j)=[p_u/delta_lg(1)^2 0 0 0;0
    p_w/delta_lg(2)^2 0 0; 0 0 p_q/delta_lg(3)^2 0; 0 0 0 p_theta/delta_lg(4)^2];
    Q_1(j-(nvx_lg-1):j,j-(nvx_lg-1):j)=[p_u/delta_lg(1).^2 0 0; 0
    p_gamma/delta_lg(5).^2 0; 0 0 p_q/delta_lg(3).^2]; %control con gamma
    Q_2(k-(nvx_ld-1):k,k-(nvx_ld-1):k)=[p_v/delta_ld(1).^2 0 0 0 0; 0
    p_p/delta_ld(2).^2 0 0 0; 0 0 p_r/delta_ld(3).^2 0 0; 0 0 0
    p_phi/delta_ld(4).^2 0;0 0 0 0 p_psi/delta_ld(5).^2];

end

end

```

Apéndice C. Generador de trayectorias

- **Longitudinal**

```
function [y_ref_par_1,error_PILong] = ref_generator_Long_vfin
(ref,t,Np,t0,tf,state,error_PILong,C)

Ke_Long=[0.2; 0.2];
Ki_Long =[0.05; 0.05];

tp = linspace(t0,tf,5);

y_ref_par_1p = zeros(4,30);
y_ref_par_1 = zeros (3,30);

us = 19;
MAC = 0.33;

cont = 1;

time1 = t;

for i = 1:Np

    if time1 < tp(1)

        y_ref_par_1p(1:4,cont)=[ref(1,1)/us; ref(3,1)/us;
ref(5,1)*MAC/(2*us); ref(8,1)];
        cont = cont+1;
        time1 = time1 + 0.02;

    elseif time1 < tp(2)

        y_ref_par_1p(1:4,cont)=[ref(1,2)/us; ref(3,2)/us;
ref(5,2)*MAC/(2*us); ref(8,2)];
        cont = cont+1;
        time1 = time1 + 0.02;

    elseif time1 < tp(3)

        y_ref_par_1p(1:4,cont)=[ref(1,3)/us; ref(3,3)/us;
ref(5,3)*MAC/(2*us); ref(8,3)];
        cont = cont + 1;
        time1 = time1 + 0.02;

    elseif time1 < tp(4)

        y_ref_par_1p(1:4,cont)=[ref(1,4)/us; ref(3,4)/us;
ref(5,4)*MAC/(2*us); ref(8,4)];
        cont = cont + 1;
        time1 = time1 + 0.02;

    elseif time1 >= tp(4)

        y_ref_par_1p(1:4,cont)=[ref(1,5)/us; ref(3,5)/us;
ref(5,5)*MAC/(2*us); ref(8,5)];
        cont = cont + 1;
        time1 = time1 + 0.02;

    end
end
```

```

% CORRECCION DEL ERROR PERMANENTE

error_PILong(1,i) = error_PILong(1,i) + y_ref_par_lp(1,i)-state(1);

y_ref_par_lp(1,i) = y_ref_par_lp(1,i) + Ke_Long(1)*(y_ref_par_lp(1,i)-
state(1)) + Ki_Long(1)*0.02*2*19/3*error_PILong(1,i);

error_PILong(2,i) = error_PILong(2,i) + y_ref_par_lp(4,i)-
state(4)+state(2);

y_ref_par_lp(4,i) = y_ref_par_lp(4,i) + Ke_Long(2)*(y_ref_par_lp(4,i)-
state(4)+state(2)) + Ki_Long(2)*0.02*2*19/3*error_PILong(2,i);

% CASO 1, Todo el horizonte de prediccion (se mantiene el state del
avion)

% if time1 >= tp(1) && time1 < tp(2)
%
%     if time1 > tp(1) + 0.25
%
%         error_PILong(1,i) = error_PILong(1,i) + y_ref_par_lp(1,i)-
state(1);
%
%         y_ref_par_lp(1,i) = y_ref_par_lp(1,i) +
Ke_Long(1)*(y_ref_par_lp(1,i)-state(1)) +
Ki_Long(1)*0.02*2*19/3*error_PILong(1,i);
%
%         error_PILong(2,i) = error_PILong(2,i) + y_ref_par_lp(4,i)-
state(4)+state(2);
%
%         y_ref_par_lp(4,i) = y_ref_par_lp(4,i) +
Ke_Long(2)*(y_ref_par_lp(4,i)-state(4)+state(2)) +
Ki_Long(2)*0.02*2*19/3*error_PILong(2,i);
%
%     else
%
%         error_PILong(1,i) = 0; % error_PILong(1,i);
%         error_PILong(2,i) = 0; % error_PILong(2,i);
%
%     end
%
% elseif time1 >= tp(2) && time1 < tp(3)
%
%     if time1 > tp(2) + 0.25
%
%         error_PILong(1,i) = error_PILong(1,i) + y_ref_par_lp(1,i)-
state(1);
%
%         y_ref_par_lp(1,i) = y_ref_par_lp(1,i) +
Ke_Long(1)*(y_ref_par_lp(1,i)-state(1)) +
Ki_Long(1)*0.02*2*19/3*error_PILong(1,i);
%
%         error_PILong(2,i) = error_PILong(2,i) + y_ref_par_lp(4,i)-
state(4)+state(2);
%
%         y_ref_par_lp(4,i) = y_ref_par_lp(4,i) +
Ke_Long(2)*(y_ref_par_lp(4,i)-state(4)+state(2)) +
Ki_Long(2)*0.02*2*19/3*error_PILong(2,i);
%
%     else
%

```

```

%         error_PILong(1,i) = 0; % error_PILong(1,i);
%         error_PILong(2,i) = 0; % error_PILong(2,i);
%
%     end
%
%     elseif time1 >= tp(3) && time1 < tp(4)
%
%         if time1 > tp(3) + 0.25
%
%             error_PILong(1,i) = error_PILong(1,i) + y_ref_par_lp(1,i)-
state(1);
%
%             y_ref_par_lp(1,i) = y_ref_par_lp(1,i) +
Ke_Long(1)*(y_ref_par_lp(1,i)-state(1)) +
Ki_Long(1)*0.02*2*19/3*error_PILong(1,i);
%
%             error_PILong(2,i) = error_PILong(2,i) + y_ref_par_lp(4,i)-
state(4)+state(2);
%
%             y_ref_par_lp(4,i) = y_ref_par_lp(4,i) +
Ke_Long(2)*(y_ref_par_lp(4,i)-state(4)+state(2)) +
Ki_Long(2)*0.02*2*19/3*error_PILong(2,i);
%
%         else
%
%             error_PILong(1,i) = 0; % error_PILong(1,i);
%             error_PILong(2,i) = 0; % error_PILong(2,i);
%
%         end
%
%     elseif time1 >= tp(4)
%
%         if time1 > tp(4) + 0.25
%
%             error_PILong(1,i) = error_PILong(1,i) + y_ref_par_lp(1,i)-
state(1);
%
%             y_ref_par_lp(1,i) = y_ref_par_lp(1,i) +
Ke_Long(1)*(y_ref_par_lp(1,i)-state(1)) +
Ki_Long(1)*0.02*2*19/3*error_PILong(1,i);
%
%             error_PILong(2,i) = error_PILong(2,i) + y_ref_par_lp(4,i)-
state(4)+state(2);
%
%             y_ref_par_lp(4,i) = y_ref_par_lp(4,i) +
Ke_Long(2)*(y_ref_par_lp(4,i)-state(4)+state(2)) +
Ki_Long(2)*0.02*2*19/3*error_PILong(2,i);
%
%         else
%
%             error_PILong(1,i) = 0; % error_PILong(1,i);
%             error_PILong(2,i) = 0; % error_PILong(2,i);
%
%         end
%
%     end
%
end

y_ref_par_1 = C*y_ref_par_lp;
y_ref_par_1 = reshape(y_ref_par_1,Np*3,1);

```

```
end
```

- **Lateral-direccional**

```
function [y_ref_par_2, error_PILD] = ref_generator_LD_vfin
(ref,t,Np,t0,tf,state,error_PILD)

Ke_LD = [0.2; 0.2];
Ki_LD = [0.05; 0.05]; %SI la aumento reduzco error pero genero
sobreoscilacion

tp = linspace(t0,tf,5);
y_ref_par_2p = zeros(5,30);
y_ref_par_2 = zeros(30*5,1);

us = 19;
b = 0.33;

time = t;
cont = 1;

for i = 1:Np

    if time < tp(1)

        y_ref_par_2p(1:5,cont)=[ref(2,1)/us; ref(4,1)*b/(2*us);
ref(6,1)*b/(2*us); ref(7,1); ref(9,1)];
        cont = cont+1;
        time = time + 0.02;

    elseif time < tp(2)

        y_ref_par_2p(1:5,cont)=[ref(2,2)/us; ref(4,2)*b/(2*us);
ref(6,2)*b/(2*us); ref(7,2); ref(9,2)];
        cont = cont+1;
        time = time + 0.02;

    elseif time < tp(3)

        y_ref_par_2p(1:5,cont)=[ref(2,3)/us; ref(4,3)*b/(2*us);
ref(6,3)*b/(2*us); ref(7,3); ref(9,3)];
        cont = cont + 1;
        time = time + 0.02;

    elseif time < tp(4)

        y_ref_par_2p(1:5,cont)=[ref(2,4)/us; ref(4,4)*b/(2*us);
ref(6,4)*b/(2*us); ref(7,4); ref(9,4)];
        cont = cont + 1;
        time = time + 0.02;

    elseif time >= tp(4)

        y_ref_par_2p(1:5,cont)=[ref(2,5)/us; ref(4,5)*b/(2*us);
ref(6,5)*b/(2*us); ref(7,5); ref(9,5)];
        cont = cont + 1;
        time = time + 0.02;

    end

    % CORRECCION DEL ERROR PERMANENTE en todo el horizonte de prediccion
end
```

```

error_PILD(1,i) = error_PILD(1,i) + y_ref_par_2p(1,i)-state(1);

y_ref_par_2p(1,i) = y_ref_par_2p(1,i) + Ke_LD(1)*(y_ref_par_2p(1,i)-
state(1)) + Ki_LD(1)*0.02*2*19/3*error_PILD(1,i);

error_PILD(2,i) = error_PILD(2,i) + y_ref_par_2p(4,i)-state(4);

y_ref_par_2p(4,i) = y_ref_par_2p(4,i) + Ke_LD(2)*(y_ref_par_2p(4,i)-
state(4)) + Ki_LD(2)*0.02*2*19/3*error_PILD(2,i);

%   if time >= tp(1) && time < tp(2)
%
%       if time > tp(1) + 0.25
%
%           error_PILD(1,i) = error_PILD(1,i) + y_ref_par_2p(1,i)-state(1);
%
%           y_ref_par_2p(1,i) = y_ref_par_2p(1,i) +
Ke_LD(1)*(y_ref_par_2p(1,i)-state(1)) + Ki_LD(1)*0.02*2*19/3*error_PILD(1,i);
%
%           error_PILD(2,i) = error_PILD(2,i) + y_ref_par_2p(4,i)-state(4);
%
%           y_ref_par_2p(4,i) = y_ref_par_2p(4,i) +
Ke_LD(2)*(y_ref_par_2p(4,i)-state(4)) + Ki_LD(2)*0.02*2*19/3*error_PILD(2,i);
%
%       else
%
%           error_PILD(1,i) = error_PILD(1,i);
%           error_PILD(2,i) = error_PILD(2,i);
%
%       end
%
%   elseif time >= tp (2) && time < tp(3)
%
%       if time > tp(2) + 0.25
%
%           error_PILD(1,i) = error_PILD(1,i) + y_ref_par_2p(1,i)-state(1);
%
%           y_ref_par_2p(1,i) = y_ref_par_2p(1,i) +
Ke_LD(1)*(y_ref_par_2p(1,i)-state(1)) + Ki_LD(1)*0.02*2*19/3*error_PILD(1,i);
%
%           error_PILD(2,i) = error_PILD(2,i) + y_ref_par_2p(4,i)-state(4);
%
%           y_ref_par_2p(4,i) = y_ref_par_2p(4,i) +
Ke_LD(2)*(y_ref_par_2p(4,i)-state(4)) + Ki_LD(2)*0.02*2*19/3*error_PILD(2,i);
%
%       else
%
%           error_PILD(1,i) = error_PILD(1,i);
%           error_PILD(2,i) = error_PILD(2,i);
%
%       end
%
%   elseif time >= tp (3) && time < tp(4)
%
%       if time > tp(3) + 0.25
%
%           error_PILD(1,i) = error_PILD(1,i) + y_ref_par_2p(1,i)-state(1);

```

```

%           y_ref_par_2p(1,i) = y_ref_par_2p(1,i) +
Ke_LD(1)*(y_ref_par_2p(1,i)-state(1)) + Ki_LD(1)*0.02*2*19/3*error_PILD(1,i);
%
%           error_PILD(2,i) = error_PILD(2,i) + y_ref_par_2p(4,i)-state(4);
%
%           y_ref_par_2p(4,i) = y_ref_par_2p(4,i) +
Ke_LD(2)*(y_ref_par_2p(4,i)-state(4)) + Ki_LD(2)*0.02*2*19/3*error_PILD(2,i);
%
%           else
%
%           error_PILD(1,i) = error_PILD(1,i);
%           error_PILD(2,i) = error_PILD(2,i);
%
%           end
%
%       elseif time >= tp (4)
%
%           if time > tp(4) + 0.25
%
%           error_PILD(1,i) = error_PILD(1,i) + y_ref_par_2p(1,i)-state(1);
%
%           y_ref_par_2p(1,i) = y_ref_par_2p(1,i) +
Ke_LD(1)*(y_ref_par_2p(1,i)-state(1)) + Ki_LD(1)*0.02*2*19/3*error_PILD(1,i);
%
%           error_PILD(2,i) = error_PILD(2,i) + y_ref_par_2p(4,i)-state(4);
%
%           y_ref_par_2p(4,i) = y_ref_par_2p(4,i) +
Ke_LD(2)*(y_ref_par_2p(4,i)-state(4)) + Ki_LD(2)*0.02*2*19/3*error_PILD(2,i);
%
%           % %
%           % %
%           % %           error_PILD(1,i) = 0; % error_PILD(1,i);
%           % %           error_PILD(2,i) = 0; % error_PILD(2,i);
%           % %
%
%           end
%
%       end
%
y_ref_par_2 = reshape(y_ref_par_2p,Np*5,1);
end

```

Apéndice D. Controladores

- **Longitudinal**

```
function
[us_opt_Ap_1_SS]=MPC_1_vfin(H_1p,y_ref_1,x_0_1,F_1,Q_1,Ap_1,G_1,us0_1,R_1,Au_
1,eq_Long,Np,LB_1,HB_1)

x_0_1p = x_0_1 - eq_Long';

us0_1(1)=us0_1(1)-(-0.08748);
us0_1(2)=us0_1(2)-(1061/6000);
t_qp_Long = 0;

eq_Long_p = zeros(90,1);

eq_Longp = [1 0.016 0];

for j = 1:Np

    eq_Long_p(3*j-2:3*j) = eq_Longp';

end

y_ref_1p = y_ref_1 - eq_Long_p;

f_1p=2*(F_1*x_0_1p-y_ref_1p)'*Q_1*G_1*Ap_1-2*us0_1'*R_1*A_u_1*Ap_1;

% us_opt_1=H_1\~-f_1'; %% DIMENSIONES 40*1
us_opt_Ap_1=H_1p\~-f_1p'; %% DIMENSIONES 10*1

us_opt_Ap_1_SS = Saturacion_Long(us_opt_Ap_1, f_1p, H_1p, Np);

% us_opt_Ap_1_SS = us_opt_Ap_1;

% coder.extrinsic('tic','toc')
% coder.extrinsic('quadprog')
%
% tic
%
% us_opt_Ap_1_qp = quadprog(H_1p,f_1p,[],[],[],[],LB_1,HB_1);
%
% t_qp_Long = toc;
%
% us_opt_Ap_1_SS = us_opt_Ap_1_qp;

end
```

- **Lateral-direccional**

```
function
[us_opt_Ap_2_SS]=MPC_2_vfin(H_2p,y_ref_2,x_0_2,F_2,Q_2,Ap_2,G_2,us0_2,R_2,Au_
2,eq_LD,Np,LB_2,HB_2)

x_0_2p = x_0_2 - eq_LD';

eq_LD_p = zeros(150,1);
t_qp_LD = 0;
```



```

for j = 1:Np
    eq_LD_p(5*j-4:5*j) = eq_LD';
end

y_ref_2p = y_ref_2 - eq_LD_p;

f_2p=2*(F_2*x_0_2p-y_ref_2p)'*Q_2*G_2*Ap_2-2*us0_2'*R_2*Au_2*Ap_2;
% f_2=2*(F_2*x_0_2-y_ref_2)'*Q_2*G_2-2*us0_2'*R_2*Au_2;

% us_opt_2=H_2\(-f_2');
us_opt_Ap_2=H_2p\(-f_2p');

us_opt_Ap_2_SS = Saturacion_LD(us_opt_Ap_2, f_2p, H_2p, Np);

% us_opt_Ap_2_SS = us_opt_Ap_2;

% coder.extrinsic('tic','toc')
% coder.extrinsic('quadprog')
%
% tic
%
% us_opt_Ap_2_qp = quadprog(H_2p,f_2p,[],[],[],[],LB_2,HB_2);
%
% t_qp_LD = toc;
%
% us_opt_Ap_2_SS = us_opt_Ap_2_qp;

end

```

Apéndice E. Saturaciones

- Longitudinal

```
function us_opt_Ap_1_SS = Saturacion_Long(us_opt_Ap_1, f_1p, H_1p, Np)

%El primero es d_e y el segundo d_t

coder.varsize('us_opt_1',[length(us_opt_Ap_1) 1],[1 0])
coder.varsize('flagv2',[length(us_opt_Ap_1) 1],[1 0])
coder.varsize('pos_matrix2',[length(us_opt_Ap_1) 2],[1 0])
coder.varsize('lgth_flag2',1,0)
m = 1;

flagv2 = zeros(length(us_opt_Ap_1),1);
us_opt_1 = us_opt_Ap_1;
cont2 = 1;
pos_matrix2 = zeros(length(us_opt_Ap_1),2);

for i = 1 : Np

    flagv2(2*i-1) = 1; % 1 es d_e y 0 es d_t

end

coder.varsize('H_1',[length(us_opt_Ap_1) length(us_opt_Ap_1)])
coder.varsize('f_1',[length(us_opt_Ap_1) 1],[1 0])
% coder.varsize('H_2pp',[length(us_opt_Ap_2) length(us_opt_Ap_2)])
% coder.varsize('f_2pp',[length(us_opt_Ap_2) 1],[1 0])

f_1 = f_1p';
H_1 = H_1p;

% máximos y mínimos

d_emin = -25*pi/180-(-0.08748);
d_emax = 25*pi/180-(-0.08748);

d_tmin = 0-1061/6000;
d_tmax = 1-1061/6000;

%Vamos a recalcular el problema como máximo 5 veces

for i = 1 : 20

    lgth_flag2 = length(us_opt_1);

    while m <= length(us_opt_1);

        if flagv2(m) == 1

            if us_opt_1(m) > d_emax

                pos_matrix2(cont2,:) = [m d_emax];
                cont2 = cont2 + 1;
                if m ~=1 && m ~= length(us_opt_1)

                    H_1pp = H_1([1:m-1 m+1:end],[1:m-1 m+1:end]);
```

```

        f_1pp = f_1([1:m-1 m+1:end])+H_1([1:m-1
m+1:end],m)*d_emax;
        flagv2 = flagv2([1:m-1 m+1:end]);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    elseif m == 1

        H_1pp = H_1(2:end,2:end);
        f_1pp = f_1(2:end)+H_1(2:end,1)*d_emax;
        flagv2 = flagv2(2:end);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    else % caso de que sature con d_emax en la ultima

        H_1pp = H_1(1:end-1,1:end-1);
        f_1pp = f_1(1:end-1)+H_1(1:end-1,end)*d_emax;
        flagv2 = flagv2(1:end-1);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    end

    elseif us_opt_1(m) < d_emin

        pos_matrix2(cont2,:) = [m d_emin];
        cont2 = cont2 + 1;

        if m ~=1 && m ~= length(us_opt_1)

            H_1pp = H_1([1:m-1 m+1:end],[1:m-1 m+1:end]);
            f_1pp = f_1([1:m-1 m+1:end])+H_1([1:m-1
m+1:end],m)*d_emin;
            flagv2 = flagv2([1:m-1 m+1:end]);

            us_opt_1 = H_1pp\ -f_1pp;
            H_1 = H_1pp;
            f_1 = f_1pp;

            break

        elseif m == 1

            H_1pp = H_1(2:end,2:end);
            f_1pp = f_1(2:end)+H_1(2:end,1)*d_emin;
            flagv2 = flagv2(2:end);

            us_opt_1 = H_1pp\ -f_1pp;
            H_1 = H_1pp;

```

```

        f_1 = f_lpp;

        break

    else % caso de que sature con d_emin en la ultima

        H_lpp = H_1(1:end-1,1:end-1);
        f_lpp = f_1(1:end-1)+H_1(1:end-1,end)*d_emin;
        flagv2 = flagv2(1:end-1);

        us_opt_1 = H_lpp\ -f_lpp;
        H_1 = H_lpp;
        f_1 = f_lpp;

        break

    end

    else % se corresponde con el caso de que no sature por d_e

        m = m + 1;

    end

    else % se corresponde con valores 0 del vecto flagv2

    if us_opt_1(m) > d_tmax

        pos_matrix2(cont2,:) = [m d_tmax];
        cont2 = cont2 + 1;

        if m ~=1 && m ~= length(us_opt_1)

            H_lpp = H_1([1:m-1 m+1:end],[1:m-1 m+1:end]);
            f_lpp = f_1([1:m-1 m+1:end])+H_1([1:m-1
m+1:end],m)*d_tmax;
            flagv2 = flagv2([1:m-1 m+1:end]);

            us_opt_1 = H_lpp\ -f_lpp;
            H_1 = H_lpp;
            f_1 = f_lpp;

            break

        elseif m == 1

            H_lpp = H_1(2:end,2:end);
            f_lpp = f_1(2:end)+H_1(2:end,1)*d_tmax;
            flagv2 = flagv2(2:end);

            us_opt_1 = H_lpp\ -f_lpp;
            H_1 = H_lpp;
            f_1 = f_lpp;

            break

        else % caso de que sature con d_tmax en la ultima

            H_lpp = H_1(1:end-1,1:end-1);
            f_lpp = f_1(1:end-1)+H_1(1:end-1,end)*d_tmax;
            flagv2 = flagv2(1:end-1);

```

```

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    end

elseif us_opt_1(m) < d_tmin

    pos_matrix2(cont2,:) = [m d_tmin];
    cont2 = cont2 + 1;

    if m ~=1 && m ~= length(us_opt_1)

        H_1pp = H_1([1:m-1 m+1:end],[1:m-1 m+1:end]);
        f_1pp = f_1([1:m-1 m+1:end])+H_1([1:m-1
m+1:end],m)*d_tmin;
        flagv2 = flagv2([1:m-1 m+1:end]);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    elseif m == 1

        H_1pp = H_1(2:end,2:end);
        f_1pp = f_1(2:end)+H_1(2:end,1)*d_tmin;
        flagv2 = flagv2(2:end);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    else % caso de que sature con d_tmin en la ultima

        H_1pp = H_1(1:end-1,1:end-1);
        f_1pp = f_1(1:end-1)+H_1(1:end-1,end)*d_tmin;
        flagv2 = flagv2(1:end-1);

        us_opt_1 = H_1pp\ -f_1pp;
        H_1 = H_1pp;
        f_1 = f_1pp;

        break

    end

else % se corresponde con el caso de que no sature por d_t

    m = m + 1;

end

end

end

```

```

m = 1;

if length(us_opt_1) == lgth_flag2

    break

end

end

if pos_matrix2(1,1) == 0

    us_opt_Ap_1_SS = us_opt_1;

else

n = size(pos_matrix2,1);

for p = 1 : n

    if pos_matrix2(p,1) ~= 0

        continue

    else

        pos_matrix2 = pos_matrix2(1:p-1,:);
        n = size(pos_matrix2,1);

        break

    end

end

for p = 1 : n

    q = n-p+1;

    if pos_matrix2(q,1) ~= 1 && pos_matrix2(q,1) ~= length(us_opt_1)-1

        us_opt_1 = [us_opt_1(1:pos_matrix2(q,1)-1); pos_matrix2(q,2);
us_opt_1(pos_matrix2(q,1):end)];

    elseif pos_matrix2(q,1) == 1

        us_opt_1 = [pos_matrix2(q,2); us_opt_1(1:end)];

    else

        us_opt_1 = [us_opt_1(1:end); pos_matrix2(q,2)];

    end

end

end

us_opt_Ap_1_SS = us_opt_1;

end
end

```

- **Lateral-direccional**

```
function us_opt_Ap_2_SS = Saturacion_LD(us_opt_Ap_2, f_2p, H_2p, Np)

%El primero es d_a y el segundo d_r

coder.varsize('us_opt_2',[length(us_opt_Ap_2) 1],[1 0])
coder.varsize('flagv',[length(us_opt_Ap_2) 1],[1 0])
coder.varsize('pos_matrix',[length(us_opt_Ap_2) 2],[1 0])
coder.varsize('lgth_flag',1,0)
m = 1;

flagv = zeros(length(us_opt_Ap_2),1);
us_opt_2 = us_opt_Ap_2;
cont = 1;
pos_matrix = zeros(length(us_opt_Ap_2),2);

for i = 1 : Np

    flagv(2*i-1) = 1; % 1 es d_a y 0 es d_r

end

coder.varsize('H_2',[length(us_opt_Ap_2) length(us_opt_Ap_2)])
coder.varsize('f_2',[length(us_opt_Ap_2) 1],[1 0])
% coder.varsize('H_2pp',[length(us_opt_Ap_2) length(us_opt_Ap_2)])
% coder.varsize('f_2pp',[length(us_opt_Ap_2) 1],[1 0])

f_2 = f_2p';
H_2 = H_2p;

% máximos y mínimos

d_amin = -25*pi/180;
d_amax = 25*pi/180;

d_rmin = -25*pi/180;
d_rmax = 25*pi/180;

%Vamos a recalcular el problema como máximo 5 veces

for i = 1 : 20

    lgth_flag = length(us_opt_2);

    while m <= length(us_opt_2);

        if flagv(m) == 1

            if us_opt_2(m) > d_amax

                pos_matrix(cont,:) = [m d_amax];
                cont = cont + 1;
                if m ~=1 && m ~= length(us_opt_2)

                    H_2pp = H_2([1:m-1 m+1:end],[1:m-1 m+1:end]);
                    f_2pp = f_2([1:m-1 m+1:end])+H_2([1:m-1
m+1:end],m)*d_amax;
                    flagv = flagv([1:m-1 m+1:end]);
```

```

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

elseif m == 1

    H_2pp = H_2(2:end,2:end);
    f_2pp = f_2(2:end)+H_2(2:end,1)*d_amax;
    flagv = flagv(2:end);

    us_opt_2 = H_2pp\ -f_2pp;
    H_2 = H_2pp;
    f_2 = f_2pp;

    break

else % caso de que sature con d_amax en la ultima

    H_2pp = H_2(1:end-1,1:end-1);
    f_2pp = f_2(1:end-1)+H_2(1:end-1,end)*d_amax;
    flagv = flagv(1:end-1);

    us_opt_2 = H_2pp\ -f_2pp;
    H_2 = H_2pp;
    f_2 = f_2pp;

    break

end

elseif us_opt_2(m) < d_amin

    pos_matrix(cont,:) = [m d_amin];
    cont = cont + 1;

    if m ~=1 && m ~= length(us_opt_2)

        H_2pp = H_2([1:m-1 m+1:end],[1:m-1 m+1:end]);
        f_2pp = f_2([1:m-1 m+1:end])+H_2([1:m-1
m+1:end],m)*d_amin;
        flagv = flagv([1:m-1 m+1:end]);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

elseif m == 1

    H_2pp = H_2(2:end,2:end);
    f_2pp = f_2(2:end)+H_2(2:end,1)*d_amin;
    flagv = flagv(2:end);

    us_opt_2 = H_2pp\ -f_2pp;
    H_2 = H_2pp;
    f_2 = f_2pp;

    break

```



```

else % caso de que sature con d_amin en la ultima
    H_2pp = H_2(1:end-1,1:end-1);
    f_2pp = f_2(1:end-1)+H_2(1:end-1,end)*d_amin;
    flagv = flagv(1:end-1);

    us_opt_2 = H_2pp\ -f_2pp;
    H_2 = H_2pp;
    f_2 = f_2pp;

    break

end

else % se corresponde con el caso de que no sature por d_a
    m = m + 1;

end

else % se corresponde con valores 0 del vecto flagv

if us_opt_2(m) > d_rmax

    pos_matrix(cont,:) = [m d_rmax];
    cont = cont + 1;

    if m ~=1 && m ~= length(us_opt_2)

        H_2pp = H_2([1:m-1 m+1:end],[1:m-1 m+1:end]);
        f_2pp = f_2([1:m-1 m+1:end])+H_2([1:m-1
m+1:end],m)*d_rmax;
        flagv = flagv([1:m-1 m+1:end]);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

elseif m == 1

        H_2pp = H_2(2:end,2:end);
        f_2pp = f_2(2:end)+H_2(2:end,1)*d_rmax;
        flagv = flagv(2:end);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

else % caso de que sature con d_rmax en la ultima

        H_2pp = H_2(1:end-1,1:end-1);
        f_2pp = f_2(1:end-1)+H_2(1:end-1,end)*d_rmax;
        flagv = flagv(1:end-1);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

```

```

        break

    end

elseif us_opt_2(m) < d_rmin

    pos_matrix(cont,:) = [m d_rmin];
    cont = cont + 1;

    if m ~=1 && m ~= length(us_opt_2)

        H_2pp = H_2([1:m-1 m+1:end],[1:m-1 m+1:end]);
        f_2pp = f_2([1:m-1 m+1:end])+H_2([1:m-1
m+1:end],m)*d_rmin;
        flagv = flagv([1:m-1 m+1:end]);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

    elseif m == 1

        H_2pp = H_2(2:end,2:end);
        f_2pp = f_2(2:end)+H_2(2:end,1)*d_rmin;
        flagv = flagv(2:end);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

    else % caso de que sature con d_rmin en la ultima

        H_2pp = H_2(1:end-1,1:end-1);
        f_2pp = f_2(1:end-1)+H_2(1:end-1,end)*d_rmin;
        flagv = flagv(1:end-1);

        us_opt_2 = H_2pp\ -f_2pp;
        H_2 = H_2pp;
        f_2 = f_2pp;

        break

    end

    else % se corresponde con el caso de que no sature por d_e

        m = m + 1;

    end

end

end

end

m = 1;

```

```

    if length(us_opt_2) == lgth_flag
        break
    end
end
if pos_matrix(1,1) == 0
    us_opt_Ap_2_SS = us_opt_2;
else
    n = size(pos_matrix,1);
    for p = 1 : n
        if pos_matrix(p,1) ~= 0
            continue
        else
            pos_matrix = pos_matrix(1:p-1,:);
            n = size(pos_matrix,1);
            break
        end
    end
    for p = 1 : n
        q = n-p+1;
        if pos_matrix(q,1) ~= 1 && pos_matrix(q,1) ~= length(us_opt_2)-1
            us_opt_2 = [us_opt_2(1:pos_matrix(q,1)-1); pos_matrix(q,2);
            us_opt_2(pos_matrix(q,1):end)];
        elseif pos_matrix(q,1) == 1
            us_opt_2 = [pos_matrix(q,2); us_opt_2(1:end)];
        else
            us_opt_2 = [us_opt_2(1:end); pos_matrix(q,2)];
        end
    end
    us_opt_Ap_2_SS = us_opt_2;
end
end

```