PROYECTO FIN DE CARRERA

Técnicas de Control Automático. Aplicaciones en Aeronáutica

Autor:
Diego Andrés Granado Sánchez-Campa

Tutora:
Mª de las Mercedes Pérez de la Parte

PROYECTO FIN DE CARRERA

Técnicas de Control Automático. Aplicaciones en Aeronáutica

por

Diego Andrés Granado Sánchez-Campa

Presentado en la

Escuela Superior de Ingenieros
Universidad de Sevilla

para la obtención del
Título de Ingeniero de Telecomunicación

Sevilla, Junio de 2004

Índice general

AGRADECIMIENTOS

Antes de comenzar, debo dar las gracias a la persona que ha hecho todos los esfuerzos posibles para que el presente proyecto se haya podido llevar a cabo: gracias a mi tutora de proyecto Mercedes Pérez de la Parte.

Muchas gracias también a los profesores Ismael Alcalá, Carlos Vivas, y Daniel Limón Marruedo, quienes me han facilitado información y bibliografía de gran utilidad en el desarrollo del presente proyecto.

Capítulo 1

Introducción

1.1 INTRODUCCIÓN

Desde que el 17 de diciembre de 1903 los hermanos Wright realizasen el primer vuelo pilotado de una aeronave, se han desarrollado infinitos avances en el campo de la aeronáutica. En los últimos años se ha puesto el acento en la automatización del vuelo, disminuyendo la carga de trabajo del piloto y aumentando la seguridad en las operaciones aéreas. En las dos últimas décadas ha aparecido una nueva generación de aeronaves conocidas como Glass Cockpit (cabinas sofisticadas), las cuales sólo necesitan dos pilotos debido a su alto grado de automatización. Paradójicamente hay una tendencia a culpar al alto grado de automatización por la mayoría de los accidentes aéreos actuales, sin embargo hay que concretar que los accidentes se deben a la falta de adaptación del piloto al Glass Cockpit [1].

Actualmente los sistemas de control en la industria de procesos deben satisfacer criterios económicos, asociados con el mantenimiento de las variables de proceso en sus referencias minimizando dinámicamente una función de coste de operación, criterios de seguridad y medioambientales, y de calidad en la producción, la cual debe satisfacer ciertas especificaciones sujetas a una demanda normalmente variable.

Por ello, se puede considerar que en la actualidad el objetivo de todo sistema de control consiste en actuar sobre las variables manipuladas de forma que puedan satisfacerse múltiples y cambiantes criterios de funcionamiento (económicos, de seguridad, medioambientales o de calidad) en presencia de cambios en las características del proceso.

El amplio abanico de metodologías actuales de control de procesos se enfrenta al cumplimiento de este objetivo. La diferencia entre las diversas técnicas radica básicamente en los compromisos hechos en la formulación matemática de los criterios de funcionamiento y en la elección de la manera de representar el proceso. La representación matemática de muchos de estos criterios se lleva a cabo en la forma de funciones objetivo dinámicas y de restricciones mientras que el proceso se representa como un modelo dinámico con incertidumbres asociadas. La importancia de las incertidumbres está siendo cada vez más reconocida y por tanto incluida explícitamente en la formulación de los controladores.

En el presente proyecto serán tratadas algunas técnicas de control automático que constituyen poderosas herramientas para afrontar estos retos, y serán enfocadas a los sistemas de control automático de vuelo (SCAV). En el presente proyecto se considerarán técnicas de Control Predictivo Generalizado (GPC), técnicas de Regulación Lineal Cuadrática (LQR), y técnicas de control robusto (H).

El sistema a controlar será una aeronave en dos de sus fases de vuelo, a saber, fase de crucero y fase de aterrizaje. El modelo para describir la dinámica de una aeronave se obtiene aplicando las leyes fundamentales de la mecánica (leyes de Newton) y eventualmente aplicando algunos principios del análisis estructural cuando aparecen deformaciones estructurales significativas. Las fuerzas que actúan sobre una aeronave son: la atracción gravitatoria, las fuerzas de propulsión, y las fuerzas aerodinámicas. Estas últimas son en general funciones no lineales de la velocidad relativa y actitud de la aeronave (ángulos) respecto a la corriente incidente (equivalente a la velocidad de vuelo) y de su geometría. Esta geometría puede cambiar por deformaciones estructurales, o por la deflexión de superficies de control (timón de profundidad y dirección, alerones, flaps, spoilers, etc).

En el presente proyecto se trabajará con un modelo lineal de aeronave, el cual puede obtenerse acotando las variaciones de las variables a valores pequeños tomando como referencia un punto de equilibrio (por ejemplo, tomando a la aeronave en vuelo recto y nivelado). Para pequeñas perturbaciones, las fuerzas aerodinámicas pueden expresarse por medio de desarrollos en series de Taylor, y despreciar los términos no lineales. Para la mayor parte de las aeronaves, por su geometría las ecuaciones pueden simplificarse algo mas, y desacoplarlas en dos grupos de tres ecuaciones diferenciales. Un grupo describe la dinámica longitudinal, es decir, del movimiento del avión en un plano perpendicular al plano de las alas (velocidades horizontal y vertical, y ángulo de cabeceo). El otro grupo describe la dinámica lateral, esto es, el movimiento en los dos planos perpendiculares al anterior (ángulos de guiñada y alabeo, y velocidad de deslizamiento lateral).

La aeronáutica mundial se encuentra a las puertas de una nueva era de la aviación, la era del crecimiento sostenible, caracterizada por la necesidad de un transporte aéreo más asequible, más limpio, menos ruidoso, más seguro y mejor protegido. La investigación y el desarrollo tecnológico serán esenciales para dar respuesta a este reto [2].

Este auge de la aviación me ha tocado personalmente más de cerca por la elección de Sevilla para las tareas de ensamblaje del avión europeo de transporte militar A400M, lo que ha revolucionado el mercado laboral de la provincia. Esto es una realidad presente para todos los sevillanos, pues el calendario del programa A400M ya ha cumplido la fase de "definición conceptual del avión" y afronta ahora el diseño detallado del aparato. A final de 2006 deberá comenzar a trabajar la línea final de montaje del avión para que el primer vuelo de pruebas del A400M se realice a final de 2007. La entrega del primer aparato, destinado a la Fuerza Aérea francesa, se realizará en septiembre de 2009. Este auge de la aviación en la provincia de Sevilla ha sido, la principal motivación para la elección de este proyecto fin de carrera, a lo que hay que añadir el interés que siempre a despertado en mí la pregunta “¿por qué vuelan los aviones?”.

1.2 OBJETIVOS

Los objetivos principales a alcanzar en el desarrollo de este proyecto son:

- Obtención de una dinámica de aeronave desacoplada para las fases de vuelo de crucero y de aterrizaje. Conocimiento de las superficies y variables del sistema de control de una aeronave y sus efectos sobre el vuelo.

- Repasar y ampliar los conocimientos de las técnicas de control automático avanzado adquiridos durante la carrera de Ingeniería de Telecomunicación.

- Implementar estrategias avanzadas de control del modelo de aeronave en las fases de crucero y de aterrizaje en Matlab/Simulink.

1.3 ESTADO DEL ARTE

A lo largo de la historia muchos ingenieros han dedicado sus esfuerzos al estudio de los sistemas de control aplicados a la aviación. Fruto de ello es que hoy se dispone de un gran numero de trabajos sobre los sistemas de control automático de vuelo. A continuación se presentan algunos artículos y libros que han servido de base al presente proyecto:

- Compendio de Aviónica Digital. Sistemas de Control y Guiado, tomo 2 [3]. Incluye un resumen de sistemas de control automático del vuelo y de los sistemas aviónicos digitales. Ha sido de gran ayuda en el análisis del movimiento de la aeronave y en la obtención de los modelos dinámicos equivalentes.

- Advances in Aircraft Flight Control [30]. Proporciona una revisión de muchos de los programas de control de vuelo de aviones comunes. Ha sido inspiración para el control con el Regulador Lineal Cuadrático (LQR), proporcionando una visión detallada de los problemas de control inherentes al vuelo de una aeronave.

- Automatic Aproach and Landing for Propulsión Controlled Aircraft by H control [27]. Se propone un control robusto H para el aterrizaje de una aeronave en situación crítica de fallo hidráulico, haciendo uso solo del empuje.

- An Intelligent Landing System Based on a Human Skill Model [25]. Se desarrolla un sistema de control de aterrizaje automático basado en un modelo de destreza de piloto. El modelo se expresa con un mapeo no lineal entrada/salida del estado de una aeronave a las señales de control proporcionadas por un piloto experto. En este sistema, todas las entradas y salidas son almacenadas en una base de datos, conocida a priori, y se construye un modelo local, cuya reacción ante los estados habituales de la aeronave, se asemeje a la operación de un piloto humano. Se introduce una técnica de adaptación de ganancia para mejorar la robustez. Ha sido de gran utilidad en el estudio del proceso de aterrizaje de una aeronave.

También han sido de gran utilidad para el desarrollo de este proyecto la lectura parcial de los siguiente libros especializados de control:

- Model Predictive Control [16]. Proporciona una amplia base teórico-práctica del control predictivo basado en modelo.

- Predictive Control with Constraints [20]. Introduce el control predictivo basado en modelo (MPC) y hace un estudio detallado de la inclusión de restricciones en los modelos.

- Multivariable Feedback Control: Analysis and Design [29]. Presenta un estudio del control de sistemas por realimentación, estudiando los cambios que produce dicha realimentación en la dinámica de los sistemas y las mejoras inducidas en la estabilidad. En este libro se presenta también una rigurosa introducción al análisis y diseño de controladores robustos para sistemas multivariables.

1.4 ESTRUCTURACIÓN DE CONTENIDOS

Tras este primer capítulo introductorio, se ha dividido el trabajo en los siguientes:

- Capítulo 2, en el que se introducen algunos conceptos previos ligados a la aviación, y se obtienen las descripciones internas correspondientes a las dinámicas longitudinal y lateral desacopladas de una aeronave.

- Capítulo 3, en el que se estudia la técnica de control LQR y se aplica al control de una aeronave en fase crucero.

- Capítulo 4, en el que se estudia la técnica de control predictivo GPC y se aplica en su solución analítica al control de una aeronave en fase crucero.

- Capítulo 5, donde se aplica un control GPC iterativo con restricciones al control de una aeronave en fase crucero.

- Capítulo 6, se formula un control PID en tiempo discreto estándar y se analiza su capacidad de control sobre la dinámica de una aeronave en fase crucero.

- Capítulo 7, en el que se resume el proceso de aterrizaje de una aeronave así como los problemas y dificultades de control que implica.

- Capítulo 8, en el que se aplica un controlador LQR para el control del aterrizaje de una aeronave.

- Capítulo 9, en el que se introduce el control robusto H , y se aplica al control de aterrizaje de una aeronave.

- Capítulo 10, en el que se incluyen las conclusiones del presente proyecto.

Capítulo 2

Obtención de un modelo de aeronave en fase crucero

2.1 SISTEMAS BÁSICOS DE REFERENCIA

En este punto se introducen los sistemas de referencia que posteriormente se usarán en la determinación de las ecuaciones que rigen el movimiento del avión.

La mayoría de conceptos usados en el desarrollo de este capítulo han sido tomados del Compendio de Aviónica Digital [3]. También ha sido de gran ayuda para la comprensión de conceptos "la enciclopedia virtual de vuelo" [4].

2.1.1 Sistemas de ejes tierra

Es el mostrado en la figura 2.1 Oe, Xe, Ye, Ze. Donde Xe puede tener cualquier dirección dentro del plano horizontal; si bien generalmente apunta hacia el norte. Donde el subíndice "e" corresponde a la palabra en inglés "Earth" (tierra).

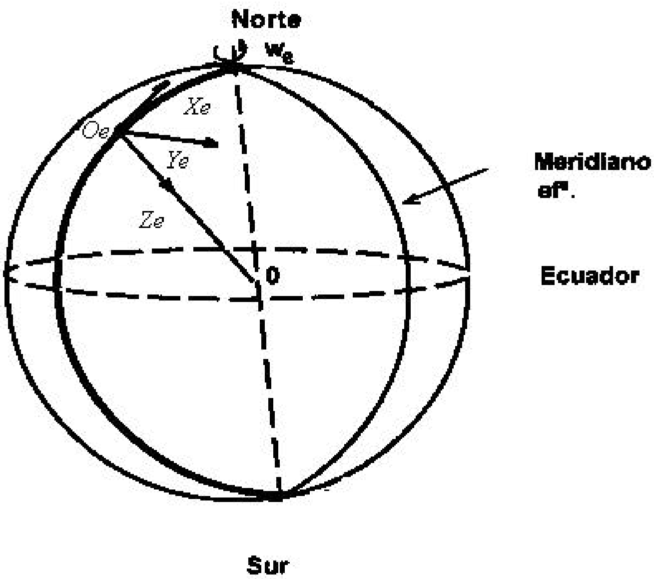


figura 2.1
(compendio de avionica digital)

2.1.2 Sistema de ejes cuerpo del avión

En este sistema: el origen Ob es cualquier punto del plano de simetría del avión; el eje Xb (eje longitudinal) está en el plano de simetría del avión, paralelo a una línea de referencia del avión y dirigido hacia delante; el eje Yb (eje transversal) es perpendicular al plano de simetría y a Xb; y el eje Zb está sobre el plano de simetría, perpendicular a Xb y hacia abajo, en vuelo normal nivelado. El subíndice "b" corresponde a la palabra en inglés "Body" (cuerpo).

En la figura 2.2 se muestran los ejes cuerpo del avión y los ángulos que se definen con los giros sobre cada eje, a saber: cabeceo (pitch = è) al girar sobre Yb; alabeo (roll = ö) al girar sobre Xb; y guiñada (yaw = ø) al girar sobre Zb.

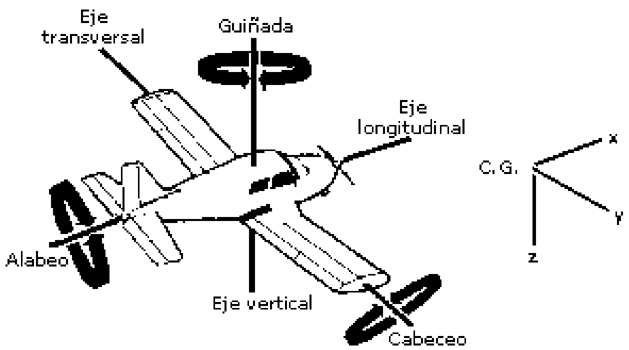


figura 2.2
(enciclopedia virtual de vuelo)

Ejes del avión y movimientos sobre ellos.

2.1.3 Sistema de ejes de estabilidad

Cuando la velocidad del viento relativo V se encuentra alineada con el eje Xb, lo que ocurre en condiciones de vuelo estacionario, al sistema de ejes cuerpo del avión se le denomina sistema de ejes de estabilidad.

2.2 ALGUNOS CONCEPTOS PREVIOS

En este apartado se definen algunos conceptos muy utilizados en el mundo de la aviación. En particular se definirán los distintos ángulos de referencia.

Actitud del avión: este término se refiere a la orientación o referencia angular de los ejes longitudinal (Xb) y transversal (Yb) del avión con respecto al horizonte, y se especifica en términos de: posición de morro o ángulo de cabeceo (pitch = è) y posición de las alas o ángulo de alabeo (roll = ö).

Trayectoria de vuelo: es la dirección seguida por el perfil aerodinámico durante su desplazamiento en el aire; es decir es la trayectoria que siguen las alas y por tanto el avión.

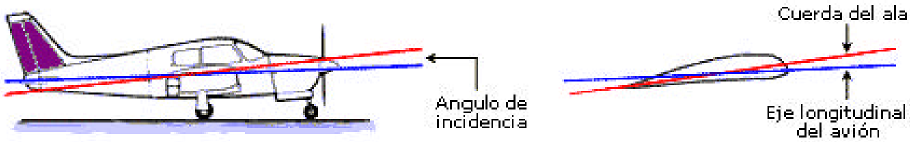
<u>Viento relativo:</u> es el flujo de aire que produce el avión al desplazarse. El viento relativo es paralelo a la trayectoria de vuelo y de dirección opuesta. Su velocidad es la relativa del avión con respecto a la velocidad de la masa de aire en que este se mueve

Trayectoria de vuelo y viento relativo

figura 2.3
(enciclopedia virtual de vuelo)

Es importante destacar que no debe asociarse la trayectoria de vuelo, ni por tanto el viento relativo, con la actitud de morro del avión; por ejemplo, una trayectoria de vuelo recto y nivelado puede llevar aparejada una actitud de morro ligeramente elevada.

<u>Ángulo de incidencia:</u> el ángulo de incidencia es el ángulo agudo formado por la cuerda del ala con respecto al eje longitudinal del avión. Este ángulo es fijo, pues responde a consideraciones de diseño y no es modificable por el piloto. Para estudios posteriores se considerará este ángulo nulo.



Angulo de incidencia.

figura 2.4 (enciclopedia virtual de vuelo)

<u>Ángulo de ataque (á):</u> el ángulo de ataque es el ángulo agudo formado por la cuerda del ala y la dirección del viento relativo. Este ángulo es variable, pues depende de la dirección del viento relativo y de la posición de las alas con respecto a este, ambos extremos controlados por el piloto. Es conveniente tener muy claro el concepto de ángulo de ataque pues el vuelo está directa y estrechamente relacionado con el mismo.

<u>Ángulo de trayectoria o senda de vuelo (ã):</u> es el ángulo agudo entre el viento relativo y el horizonte.

Una vez definidos estos ángulos es importante poder relacionarlos. Para ello es útil la figura 2.5:

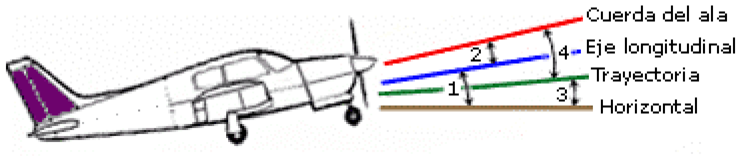


figura 2.5 (enciclopedia virtual de vuelo)

De la figura 2.5 se puede deducir la siguiente relación entre ángulos:

$$\mathit{cabeceo}(1) + \mathit{incidencia}(2) = \mathit{trayectoria}(3) + \mathit{ataque}(4)$$
$$q + \mathit{incidencia}(2) = g + a$$

Si se considera un ángulo de incidencia nulo se llega a: ã = è-á

2.3 SUPERFICIES Y MANDOS DE CONTROL

Además de que un avión vuele, es necesario que este vuelo se efectúe bajo control del piloto, es decir, que el avión se mueva respondiendo a sus órdenes. Por otro lado, es de gran interés contar con dispositivos que, a voluntad del piloto, aporten sustentación adicional (o no-sustentación) facilitando la realización de ciertas maniobras.

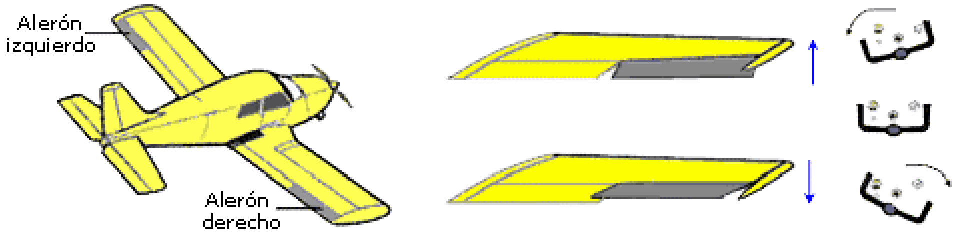
Para lograr una u otra funcionalidad se emplean superficies aerodinámicas, denominándose primarias a las que proporcionan control y secundarias a las que modifican la sustentación.

Las superficies de mando y control modifican la aerodinámica del avión provocando un desequilibrio de fuerzas, una o más de ellas cambian de magnitud. Este desequilibrio, es lo que hace que el avión se mueva sobre uno o más de sus ejes, incremente la sustentación, o aumente la resistencia.

En este capítulo se trabajará con una aeronave en fase crucero, por lo sólo son de interés las superficies primarias, pues las secundarias solo se despliegan en maniobras a baja velocidad (despegue o aterrizaje). Estas superficies de control son tres: alerones, timón de profundidad y timón de dirección (se podría añadir el control del empuje o tracción como una cuarta posibilidad de control).

Alerones: son unas superficies móviles, situadas en la parte posterior del extremo de cada ala, cuyo accionamiento provoca el movimiento de alabeo del avión sobre su eje longitudinal. Su ubicación en el extremo del ala se debe a que en esta parte es mayor el par de fuerza ejercido.

Los alerones tienen un movimiento asimétrico. Al girar el volante hacia un lado, el alerón del ala de ese lado sube y el del ala contraria baja, ambos en un ángulo de deflexión proporcional a la cantidad de giro dado al volante. El alerón arriba en el ala hacia donde se mueve el volante implica menor curvatura en esa parte del ala y por tanto menor sustentación, lo cual provoca que ese ala baje; el alerón abajo del ala contraria supone mayor curvatura y sustentación lo que hace que ese ala suba. Esta combinación de efectos contrarios es lo que produce el movimiento de alabeo hacia el ala que desciende.



Alerones y mando de control.

figura 2.6 (enciclopedia virtual de vuelo)

Timón de profundidad: es la superficie o superficies móviles situadas en la parte posterior del empenaje horizontal de la cola del avión. Su accionamiento provoca el movimiento de cabeceo del avión (morro arriba o morro abajo) sobre su eje transversal. Obviamente, el movimiento de cabeceo del avión provoca la modificación del ángulo de ataque; es decir que el mando de control del timón de profundidad controla el ángulo de ataque.

Al tirar del volante de control, esta superficie sube, mientras que al empujarlo baja. El timón arriba produce menor sustentación en la cola, con lo cual esta baja y por tanto el morro sube (mayor ángulo de ataque). El timón abajo aumenta la sustentación en la cola, esta sube y por tanto el morro baja (menor ángulo de ataque). De esta manera se produce el movimiento de cabeceo del avión y por extensión la modificación del ángulo de ataque.

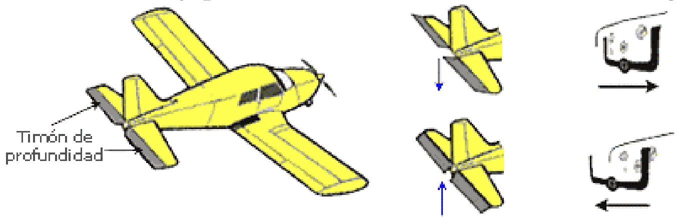


figura 2.7 (enciclopedia virtual de vuelo)

Timón de profundidad y mando de control.

Timón de dirección: es la superficie móvil montada en la parte posterior del empenaje vertical de la cola del avión. Su movimiento provoca el movimiento de guiñada del avión sobre su eje vertical. Suele tener una deflexión máxima de 30º a cada lado.

Al pisar el pedal derecho, el timón de dirección gira hacia la derecha, provocando una reacción aerodinámica en la cola que hace que esta gire a la izquierda, y por tanto el morro del avión gire (guiñada) hacia la derecha. Al pisar el pedal izquierdo, sucede lo contrario: timón a la izquierda, cola a la derecha y morro a la izquierda.

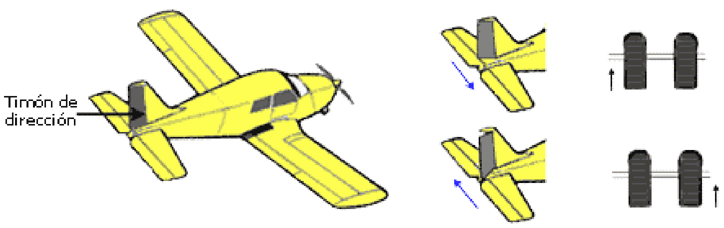


figura 2.8 (enciclopedia virtual de vuelo)

Timón de dirección y pedales de control.

2.4 INTRODUCCIÓN A LA DINÁMICA DEL AVIÓN

La obtención del modelo dinámico del avión se ha realizado en dos secciones. En la presente sección se obtendrá, partiendo de las leyes elementales de la física, un primer modelo intermedio. En la siguiente sección (sección 2.5) se introducirá una notación simplificada estándar. También se introducirán los desplazamientos de las superficies de control obteniéndose finalmente un modelo sencillo del movimiento de la aeronave.

2.4.1 Aplicación de las leyes de Newton

Se comienza pues, el proceso de simplificación, para obtener las ecuaciones del movimiento de la aeronave, introduciendo la hipótesis 1 de que “el avión es un cuerpo rígido”. En consecuencia, su movimiento puede ser descrito completamente por una traslación de su centro de gravedad y por una rotación alrededor de este punto.

Se supondrá también, hipótesis 2, que se tiene “un sistema inercial de referencia” (sistema en reposo o en movimiento rectilíneo y uniforme respecto a las estrellas fijas) . Esto implica ciertas restricciones válidas para vuelos normales de los aviones ordinarios, pero no aceptables en el caso de operaciones fuera de la atmósfera.

Con estas hipótesis se puede aplicar las ecuaciones de Newton al estudio del movimiento de la aeronave:

$$\frac{d(m\overline{V})}{dt} = \sum \overline{F} \quad \text{(teorema cantidad de movimiento)} \quad (2.1)$$

$$\frac{d(\overline{H})}{dt} = \sum \overline{M} \quad \text{(teorema momento cinetico)} \quad (2.2)$$

siendo

$\sum \overline{F} =$ Resultante del sistema de fuerzas exteriores

$\overline{V} =$ Velocidad absoluta del centro de gravedad (C.deG.) del avion

$m =$ Masa del avion

$\overline{H} = \overline{\overline{I}}\,\overline{w} =$ Momento dinamico respecto a C.deG.

$\overline{w} =$ velocidad angular del avion con respecto al C.deG.

$\overline{\overline{I}} =$ Tensor de inercia definido como

$$\overline{\overline{I}} = \begin{bmatrix} Ix & -Ixy & -Ixz \\ -Ixy & Iy & -Iyz \\ -Ixz & -Iyz & Iz \end{bmatrix} \text{con} \begin{cases} Ix = Ixx = \int (y^2 + z^2)dm = \text{momento inercia} \\ Ixy = \int xy\,dm = \text{producto inercia} \end{cases}$$

Si se introduce la nueva hipótesis 3 que “la masa del avión y su distribución es constante” la ecuación (2.1) puede escribirse como:

$$\sum \overline{F} = \frac{d(m\overline{V})}{dt} = \frac{d\overline{V}}{dt}m + \frac{dm}{dt}\overline{V} = \frac{d\overline{V}}{dt}m$$

Escribiendo las ecuaciones (2.1) y (2.2), bajo las hipótesis hechas, en el sistema de ejes cuerpo del avión, se tiene:

$$\overline{F} = m(\frac{\partial \overline{V}}{\partial t} + \overline{w} \wedge \overline{V}) \qquad (2.3)$$

$$\overline{M} = \frac{\partial \overline{H}}{\partial t} + \overline{w} \wedge \overline{H} \qquad (2.4)$$

Siendo $\overline{F}(Fx, Fy, Fz)$, $\overline{M}(Mx, My, Mz)$, $\overline{V}(Vx, Vy, Vz)$ *y* $\overline{w}(wx, wy, wz)$ *las fuerzas, momentos y velocidades (lineales y angulares) absolutas, proyectadas en los ejes de referencia cuerpo – avion.*

La figura 2.9 muestra la posición de los ejes cuerpo del avión:

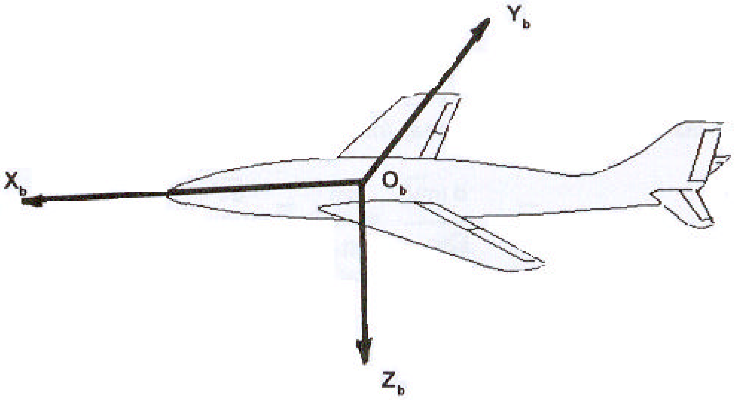


figura 2.9
(compendio de avionica digital)

Desarrollando las ecuaciones (2.3) y (2.4) teniendo en cuenta que $\overline{w} \wedge \overline{H} = \overline{w} \wedge (\overline{\overline{I}}\overline{w})$ e introduciendo ahora la hipótesis 4, a saber que “el plano XbZb, es un plano de simetría del avión” Ixy=Iyz=0, se puede obtener el siguiente conjunto de ecuaciones, donde la letras primadas representan derivadas temporales:

$$\left.\begin{aligned}
Fx &= m(V'x - \omega_z Vy + \omega_y Vz) \\
Fy &= m(V'y - \omega_z Vx - \omega_z Vz) \\
Fz &= m(V'z - \omega_y Vx + \omega_x Vy) \\
Mx &= Ix\omega'_x - Ixz\omega'_z + (Iz - Iy)\omega_y\omega_z - Ixz\omega_x\omega_y \\
My &= Iy\omega'_y - (Iz - Ix)\omega_x\omega_z + Ixz(\omega_x{}^2 - \omega_z{}^2) \\
Mz &= Ixz\omega'_x + Iz\omega'_z - (Ix - Iy)\omega_x\omega_y + Ixz\,\omega_y\omega_z
\end{aligned}\right\} (2.5)$$

Estas son las ecuaciones diferenciales del movimiento del avión, en las que los primeros términos de las ecuaciones representan la acción combinada de todas las fuerzas externas (en el caso de las tres primeras ecuaciones), a saber, fuerzas aerodinámicas, gravitatorias y propulsoras.

Las fuerzas y los momentos aerodinámicos son funciones de V y ù. Las componentes de la fuerza gravitatoria en las 3 primeras ecuaciones dependen de la orientación del avión con respecto a los ejes fijos o inerciales.

2.4.2 Fuerzas y momentos externos

Las fuerzas F y los momentos M externos incluyen los componentes aerodinámicos, gravitatorios y propulsores, según indican las ecuaciones siguientes:

$$\overline{F} = \overline{F}a + \overline{F}g + \overline{F}p$$
$$\overline{M} = \overline{M}a + \overline{M}p \qquad (\overline{M}g = 0 \ pues\ \overline{F}g \ pasa\ por\ el\ C.deG.)$$

en las que los subíndices a, g y p identifican las componentes aerodinámicas, gravitatorias y propulsoras, respectivamente.

Considerando m la masa del avión y g la aceleración de la gravedad, las componentes gravitatorias de la fuerza Fg son:

$$Fgx = -mg\,senq$$
$$Fgy = mg\,senj\ \cos q$$
$$Fgz = mg\cos j\ \cos q$$

(è y ö relacionan ejes cuerpo-avión con ejes tierra como se muestra en figura 2.10)

En consecuencia las ecuaciones (2.5) se pueden escribir como sigue:

$$\left.\begin{array}{l}
-mg\,senq + Fax + Fpx = m(V'x - w_z Vy + w_y Vz) \\
mg\,senj\ \cos q + Fay + Fpy = m(V'y - w_z Vx - w_z Vz) \\
mg\cos j\ \cos q + Faz + Fpz = m(V'z - w_y Vx + w_x Vy) \\
Max + Mpx = Ixw'_x - Ixzw'_z + (Iz - Iy)w_y w_z - Ixzw_x w_y \\
May + Mpy = Iyw'_y - (Iz - Ix)w_x w_z + Ixz(w_x^2 - w_z^2) \\
Maz + Mpz = Ixzw'_x + Izw'_z - (Ix - Iy)w_x w_y + Ixzw_y w_z
\end{array}\right\} \quad (2.6)$$

Este conjunto de ecuaciones tiene 8 variables desconocidas (Vx, y, z, $ù_x$, $ù_y$, $ù_z$, ö y è), por consiguiente, se necesitan estas 6 ecuaciones más dos adicionales, para poder encontrar su solución, es decir, el movimiento del avión. Para definir estas dos nuevas ecuaciones se procede como se indica seguidamente.

Suponiendo, según ya se ha dicho, que el sistema tierra es un sistema de referencia inercial, la velocidad del C.deG. del avión es $\overline{V}e = d\overline{r}/dt$, siendo $\overline{r}$ el vector de posición del C.deG. con respecto al sistema tierra.

Para relacionar $\overline{V}e$ y $\overline{V}b$(Vx,Vy,Vz) se puede usar una matriz de transformación adecuada (atendiendo a la figura 2.10) como sigue:

$$\begin{bmatrix} Vex \\ Vey \\ Vez \end{bmatrix} = \begin{bmatrix} \cos q\cos y & \cos q\ seny & -senq \\ senj\ senq\cos y - \cos j\ seny & senj\ senq\ seny + \cos j\ \cos y & senj\ \cos q \\ \cos j\ senq\cos y + senj\ seny & \cos j\ senq\ seny - senj\ \cos y & \cos j\ \cos q \end{bmatrix}^{-1} \begin{bmatrix} Vbx \\ Vby \\ Vbz \end{bmatrix}$$

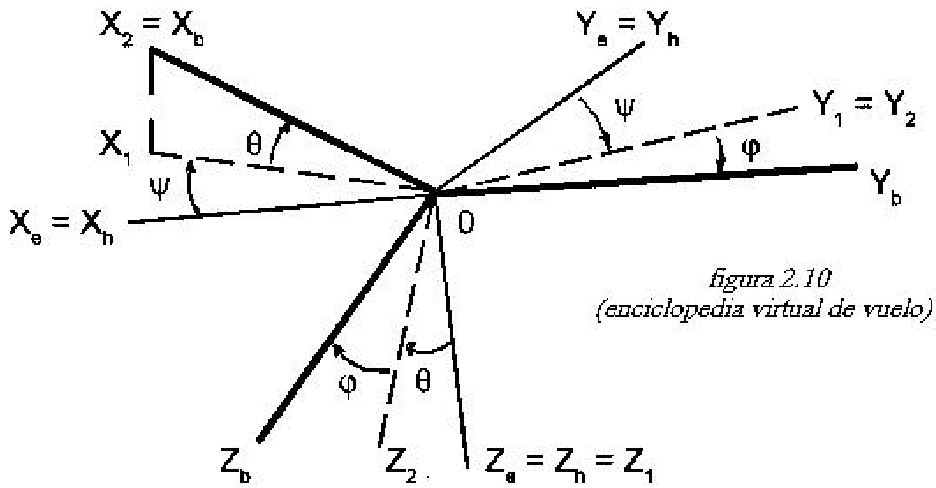


figura 2.10
(enciclopedia virtual de vuelo)

Representación de las tres rotaciones efectuadas para mover al sistema de ejes tierra, al sistema de ejes de referencia cuerpo-avión. Estas tres rotaciones coinciden con los ángulos de guiñada ø (yaw), de cabeceo è (pitch), y de alabeo ö (roll).

Estableciendo esto, y teniendo en cuenta que $\overline{w} = j'\overline{i}_b + q'\overline{j}_1 + y'\overline{k}_h$ se pueden definir las siguientes ecuaciones:

$$\left.\begin{array}{l} w_x = j' - y'\,senq \\ w_y = q'\cos j + y'\cos q\; senj \\ w_z = y'\cos q\cos j - q'\,senj \end{array}\right\} \quad (2.7)$$

Por tanto, se puede determinar la ecuación de la trayectoria del avión, si se conocen las variables Vx, Vy, Vz, $ù_x$, $ù_y$, $ù_z$ en función del tiempo.

2.4.3 Linealización de las ecuaciones del movimiento perturbado del avión

Se denominará movimiento perturbado de una aeronave a cualquier clase de movimiento de la misma distinto del vuelo estacionario.

Sin embargo, se considerarán solamente pequeñas perturbaciones, a partir de la condición de referencia estacionaria. En consecuencia, siguiendo el proceso de simplificación iniciado, se introduce una nueva hipótesis 5 que consiste en que “las perturbaciones, a partir del vuelo estacionario de referencia, son suficientemente pequeñas”, a fin de que resulten despreciables los términos de segundo orden (productos y cuadrados de las cantidades de perturbación) y que los cosenos y los senos de los ángulos relativos a la perturbación puedan igualarse a uno y a los ángulos mismos, respectivamente.

Entonces, denotando Vsx, Vsy y Vsz a las componentes de la velocidad lineal del avión, antes de la perturbación, y Vx(t), Vy(t) y Vz(t) las variaciones de las component es de la velocidad como consecuencia de la perturbación. (Nota: el subíndice s indica movimiento estacionario). Y si se procede análogamente para ù, y para ø, ö y è, se puede escribir:

$$
\begin{array}{lll}
Vx = Vsx + \Delta Vx & w_x = w_{sx} + \Delta w_x & j = j_s + \Delta j \\
Vy = Vsy + \Delta Vy & w_y = w_{sy} + \Delta w_y & q = q_s + \Delta q \\
Vz = Vsz + \Delta Vz & w_z = w_{sz} + \Delta w_x & y = y_s + \Delta y
\end{array}
$$

Asimismo se puede escribir para el caso general de fuerzas y momentos (aerodinámicos y de propulsión):

$$
\begin{array}{ll}
Fa = Fsa + \Delta Fa & Ma = Msa + \Delta Ma \\
Fp = Fsp + \Delta Fp & Mp = Msp + \Delta Mp
\end{array}
$$

y para las componentes de fuerzas gravitatorias:

$$
\begin{array}{l}
Fgx = -mg\, sen(q_{sw} + \Delta q) \\
Fgy = mg\, sen(j_s + \Delta j)\cos(q_s + \Delta q) \\
Fgz = mg\cos(j_s + \Delta j)\cos(q_s + \Delta q)
\end{array}
$$

Por tanto, teniendo en cuenta que se actúa con perturbaciones pequeñas, las ecuaciones linealizadas resultantes son:

$$
\left.
\begin{array}{l}
-mg\,\Delta q\cos q_s + \Delta Fax + \Delta Fpx = \\
= m(\Delta V'x - Vsy\Delta w_z - w_{sz}\Delta Vy + Vsz\Delta w_y + w_{sy}\Delta Vz) \\
\\
-mg\,\Delta q\, sen j_s + mg\,\Delta j\cos j_s\cos q_s + \Delta Fay + \Delta Fpy = \\
= m(\Delta V'y + Vsx\Delta w_z + w_{sz}\Delta Vx - Vsz\Delta w_x - w_{sx}\Delta Vz) \\
\\
-mg\,\Delta q\cos j_s\, sen q_s - mg\,\Delta j\, sen j_s\cos q_s + \Delta Faz + \Delta Fpz = \\
= m(\Delta V'z - Vsx\Delta w_y - w_{sy}\Delta Vx + Vsy\Delta w_x + w_{sx}\Delta Vy) \\
\\
\Delta Max + \Delta Mpx = \\
= Ix\,\Delta w'_x - Ixz\,\Delta w'_z + (Iz - Iy)(w_{sy}\Delta w_z + w_{sz}\Delta w_y) - Ixz(w_{sx}\Delta w_y + w_{sy}\Delta w_x) \\
\\
\Delta May + \Delta Mpy = \\
= Iy\,\Delta w'_y - (Iz - Ix)(w_{sx}\Delta w_z + w_{sz}\Delta w_x) + 2Ixz(w_{sx}\Delta w_x - w_{sz}\Delta w_z) \\
\\
\Delta Maz + \Delta Mpz = \\
= Iz\,\Delta w'_z - Ixz\,\Delta w'_x - (Ix - Iy)(w_{sx}\Delta w_y + w_{sy}\Delta w_x) + Ixz(w_{sy}\Delta w_z + w_{sz}\Delta w_y)
\end{array}
\right\} \quad (2.8)
$$

Análogamente, el conjunto de ecuaciones (2.7) puede escribirse como sigue:

$$
\left.
\begin{array}{l}
\Delta w_x = \Delta j' - y'_s\,\Delta q\cos q_s - \Delta y'\, sen q_s \\
\\
\Delta w_y = -\Delta j\,(q'_s\, sen j_s) + \Delta q'\cos j_s - \Delta j\,(y'_s\cos q_s\cos j_s) + \\
-\Delta q\,(y'_s\, sen q_s\, sen j_s) + \Delta y'(\cos q_s\, sen j_s) \\
\\
\Delta w_z = -\Delta j\,(y'_s\cos q_s\, sen j_s) - \Delta q\,(y'_s\, sen q_s\cos j_s) + \\
+\Delta y'(\cos q_s\cos j_s) - \Delta j\,(q'_s\cos j_s) - \Delta q'\, sen j_s
\end{array}
\right\} \quad (2.9)
$$

Estos dos sistemas de ecuaciones (2.8) y (2.9) constituyen las ecuaciones linealizadas generales, aplicables a las condiciones de vuelo estacionario, cuando se introducen perturbaciones pequeñas.

Es decir, se supone que antes de aplicar una perturbación o una orden de control (mediante la deflexión de una superficie de control) el avión se encuentra en la condición de equilibrio y las perturbaciones a partir de esta condición son pequeñas. Por consiguiente la suma de todas las fuerzas y momentos externos son cero antes de la perturbación, y los términos $\Delta\overline{F}s$ y $\Delta\overline{M}s$ (cambios de $\overline{F}$ y $\overline{M}$), en las ecuaciones (2.8), son las componentes existentes después de la presencia de una perturbación, o de un cambio controlado.

Bajo estas condiciones, y suponiendo (hipótesis 6) que, "antes de la aplicación de la perturbación, la velocidad del avión está situada en su plano de simetría y que su ángulo de alabeo es cero (alas niveladas) y también su velocidad angular"; lo que equivale a decir que el avión realizaba un vuelo rectilíneo, simétrico y estacionario, antes de la introducción de la perturbación, se puede escribir:

Vuelo simétrico implica que Vsy = 0 y $ø_s = 0$
Vuelo con alas niveladas supone que $ö_s = 0$
Vuelo estacionario implica que $ù_{sx} = ù_{sy} = ù_{sz} = 0$ $ö_s' = ø'_s = è'_s = 0$

En consecuencia, las ecuaciones (2.8) y (2.9) pueden simplificarse como sigue:

$$\left.\begin{array}{l} -mg\,\Delta q\,\cos q_s + \Delta Fax + \Delta Fpx = m(\Delta V'x + Vsz\Delta w_y) \\ +mg\,\Delta j\,\cos q_s + \Delta Fay + \Delta Fpy = m(\Delta V'y + Vsx\Delta w_z - Vsz\Delta w_x) \\ -mg\,\Delta q\,sen q_s + \Delta Faz + \Delta Fpz = m(\Delta V'z - Vsx\Delta w_y) \\ \Delta Max + \Delta Mpx = Ix\,\Delta w'_x - Ixz\,\Delta w'_z \\ \Delta May + \Delta Mpy = Iy\,\Delta w'_y \\ \Delta Maz + \Delta Mpz = Iz\,\Delta w'_z - Ixz\,\Delta w'_x \end{array}\right\} \quad (2.10)$$

$$\left.\begin{array}{l} \Delta w_x = \Delta j' - \Delta y'\,sen q_s \\ \Delta w_y = \Delta q' \\ \Delta w_z = \Delta y'\cos q_s \end{array}\right\} \quad (2.11)$$

2.5 DINÁMICA DEL AVIÓN EN FASE CRUCERO

2.5.1 Adaptación al desarrollo en serie de Taylor

Es bien conocido, que si F es una función continua de un cierto número de variables (x, y, z, etc.) y si además de estas variables son ellas mismas funciones de otras variables, por ejemplo si se supone que z es función de t, se puede escribir F=F(x, y, z(t), ...), y aplicando el desarrollo en serie de Taylor:

$$F = F_0 + (¶f\,/\,¶x)dx + (¶f\,/\,¶y)dy + \ldots.. + (¶f\,/\,¶z)(dz/dt)dt + \ldots$$
$$donde \quad F_0 = F[x_0, y_0, z_0, \ldots]$$

Aplicando esta ecuación a cambios finitos pero pequeños de las variables puede expresarse como sigue:

$$F - F_0 = \Delta F = (¶f\ / ¶x)\Delta x + (¶f\ / ¶y)\Delta y + \ldots\ldots + (¶f\ / ¶z)(dz / dt)\Delta t + \ldots \quad (2.12)$$

siendo x, y, t, ... esos valores finitos, peque ños, con lo que los términos de segundo o mayor orden de este desarrollo pueden despreciarse.

Así, por ejemplo, en el caso de aplicar esto a las fuerzas y los momentos aerodinámicos, las variables serían la densidad del aire (r), la velocidad del avión (V), y por consiguiente la altura y las componentes de la velocidad lineal.

Además, a través de los coeficientes adimensionales C, son funciones de los números de Mach y de Reynolds, de los ángulos de ataque y de derrape, y de las componentes de velocidad lineal y angular y de sus derivadas.

A pesar de todo esto, las ecuaciones, que así se obtienen son aún demasiado complejas y resulta necesaria la introducción de nuevas hipótesis de simplificación.

Una nueva hipótesis (7) especifica que "el plano XZ es no sólo un plano de simetría geométrica e inercial, sino también un plano de simetría de aerodinámica ("aerodynamic trim symmetry"). Esto significa que el sistema de ejes-cuerpo del avión es un sistema de ejes-viento también".

Por supuesto, al seleccionar ahora el nuevo eje X en la dirección del viento, los nuevos valores de Ix e Iz no son constantes y son función del ángulo de ataque á. Sin embargo, teniendo en cuenta que la variación de á debida a una perturbación o un cambio de control (o a una combinación de ambos) es muy reducida, se desprecia aquí su influencia en los momentos inerciales del avión mencionados.

También se introduce la hipótesis (8) de que "el flujo de aire es un flujo cuasi-estacionario". Esto implica que los términos de orden más alto, en la serie de Taylor, debidos a las aceleraciones del cuerpo del avión, pueden despreciarse. En consecuencia, todas las derivadas relativas a las variaciones de las velocidades pueden ser despreciadas.

Finalmente se introduce una nueva hipótesis (9) que expresa que "los cambios de las propiedades de la atmósfera (es decir, la densidad del aire y la velocidad del sonido; consecuencia de haber supuesto perturbaciones pequeñas; en altura, en este caso) son despreciables".

2.5.2 Fuerzas y momentos aerodinámicos y propulsores

La fuerza aerodinámica total y estacionaria aplicada al avión, se considera desglosada en las componentes de "sustentación" (L) y de "resistencia" (D).

La sustentación L es perpendicular a la trayectoria de vuelo y viene dada por

$$L = (1/2)(r\,C_L S V^2)$$

siendo C_L el coeficiente (adimensional) de sustentación, ñ la densidad del aire, S es el área característica del avión y V es la velocidad del avión con relación al aire.

La resistencia viene dada por $D = (1/2)(r C_D S V^2)$ siendo C_D el coeficiente (adimensional) de resistencia.

La figura 2.11 indica la orientación de estas fuerzas y de la velocidad del avión, con relación al sistema de ejes-cuerpo.

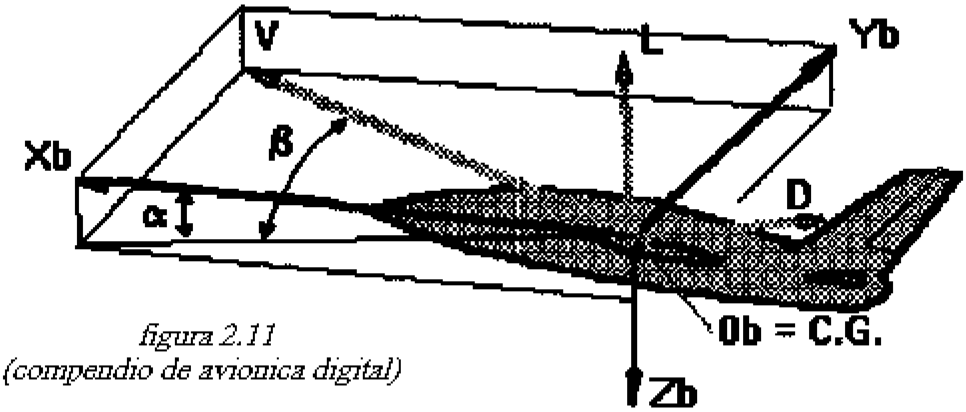


figura 2.11
(compendio de avionica digital)

Las componentes de la velocidad V sobre los ejes-cuerpo del avión son:

$$\left.\begin{aligned} Vbx &= V\cos b\ \cos a \\ Vby &= Vsenb \\ Vbz &= V\cos b\ sena \end{aligned}\right\} \quad (2.13)$$

siendo á el ángulo de ataque, dado por $a = \arctan(Vbz/Vbx) \quad (2.14)$; y â el ángulo de derrape, dado por $b = arcsen(Vby/V) \quad (2.15)$

Las componentes de sustentación y de resistencia sobre los ejes-cuerpo del avión son:

$$Fax = L\,sena - D\cos b\ \cos a$$

$$Fay = D\,senb$$

$$Faz = -L\cos a - D\cos b\ sena$$

$$Max = (1/2)(r\,CrSbV^2)$$

$$May = (1/2)(r\,CpSbV^2)$$

$$Maz = (1/2)(r\,CySbV^2)$$

siendo Cr, Cp y Cy los coeficientes (adimensionales) de alabeo (“roll”), de cabeceo (“pitch”) y de guiñada (“yaw”) respectivamente; S es la superficie alar y b la envergadura del ala y c la cuerda media aerodinámica.

Las componentes de las fuerzas y momentos propulsores pueden determinarse con ayuda de la figura 2.12.

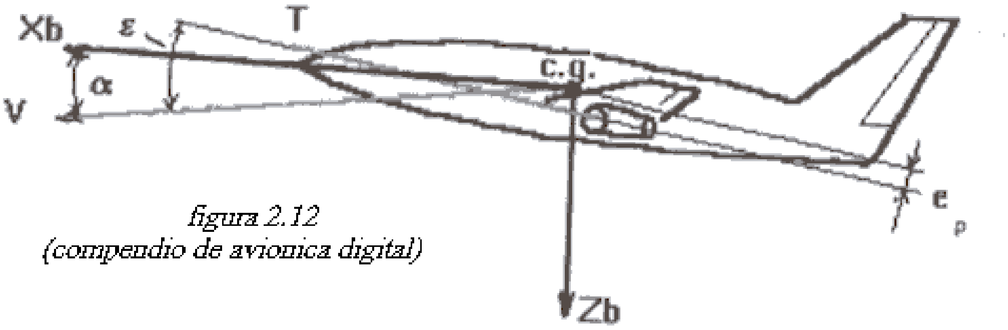


figura 2.12
(compendio de avionica digital)

å es el ángulo, que forma la línea del empuje con respecto a la velocidad V del avión; e_p es la excentricidad del empuje (por convenio positiva cuando es hacia abajo), y T es la fuerza de empuje o fuerza propulsora. Las ecuaciones propulsoras son:

$$\begin{aligned} Fpx &= T\cos(e - a) \\ Fpy &= T\,sen(e - a) \\ Mpy &= T\,e_p \end{aligned} \qquad (2.16)$$

Teniendo en cuenta que el empuje es solamente una función de la velocidad del avión, de la densidad del aire y de la posición de la palanca de control del empuje, es posible calcular las variaciones de las fuerzas, producidas por una perturbación cualquiera, con la ayuda de las derivadas parciales del empuje, con respecto a las variables mencionadas.

Sin embargo, el cálculo del momento perturbado (producido por la perturbación) no puede obtenerse por la simple diferenciación de las ecuaciones precedentes, debido a la influencia de los momentos de compensación aerodinámica de alabeo.

2.5.3 Forma convencional de las ecuaciones linealizadas del movimiento perturbado

Partiendo de las ecuaciones (2.10) y sustituyendo, en ellas, las variaciones de los términos gravitatorios, con la expresión de Fg se tiene:

$$\left.\begin{aligned} \Delta Fgx + \Delta Fax + \Delta Fpx &= m(\Delta V'x + Vsz\Delta w_y) \\ \Delta Fgy + \Delta Fay + \Delta Fpy &= m(\Delta V'y + Vsx\Delta w_z - Vsz\Delta w_x) \\ \Delta Fgz + \Delta Faz + \Delta Fpz &= m(\Delta V'z - Vsx\Delta w_y) \\ \Delta Max + \Delta Mpx &= Ix\,\Delta w'_x - Ixz\,\Delta w'_z \\ \Delta May + \Delta Mpy &= Iy\,\Delta w'_y \\ \Delta Maz + \Delta Mpz &= Iz\,\Delta w'_z - Ixz\,\Delta w'_x \end{aligned}\right\}(2.17)$$

Las ecuaciones (2.10) o (2.17) y las ecuaciones (2.11) son las ecuaciones linealizadas y simplificadas de una aeronave, compensada para realizar un vuelo rectilíneo, es decir en condiciones de vuelo estacionario, simétrico y nivelado (alabeo cero). Éstas son las condiciones hipotéticas, generalmente empleadas en el análisis de los Sistemas Automáticos de Control del Vuelo (SCAVs). Bajo estas condiciones de vuelo, las variaciones de velocidades lineales y angulares son cero.

Por tanto, los picados, subidas y derrapes estacionarios, sin aceleraciones longitudinales, se consideran condiciones de vuelo rectilíneo. Además, las subidas y picados con alas niveladas y las recogidas (“pull-ups”) sin derrape son condiciones simétricas de vuelo.

Obviamente, los movimientos de alabeo (“rolling”), de giro o virage (“turning”) y de derrape (“sideslipping”) son condiciones asimétricas típicas de vuelo.

Por otra parte, un vuelo compensado, rectilíneo implica tambíen que $ø_s = 0$ y $è_s = 0$. Para una condición de vuelo compensado particular, el avión tendrá unos ciertos valores de Vsx, Vsz y $è_s$.

Estos valores pueden ser cero, aunque en el caso de un avión convencional, la velocidad estacionaria, hacia delante Vsx debe ser mayor que su velocidad de pérdida (sin embargo en el caso de un helicóptero Vsx, Vsz y $è_s$ pueden ser nulas).

Las ecuaciones precedentes suelen ser escritas en una forma más sencilla y breve, mediante una serie de convenios y designaciones aplicados a las fuerzas, los momentos y los ángulos, tales como los establecidos en los documentos ARINC 407 Y 407-1 (“Synchro System Manual”) y que se indican a continuación:

Ejes		Fuerza	Momento			Angulo (rad)		Velocidades	
Designación	Símbolo	Símbolo	Desig.	Símb.	Direcc.+	Des.	Símb.	Lin.(m/s)	Ang.(rad/s)
Longitudinal	X	X	Roll	L	Y→X	Roll	ö	u	p
Lateral	Y	Y	Pitch	M	Z→X	Pitch	è	v	q
Normal	Z	Z	Yaw	N	X→Y	Yaw	ø	w	r

Con estos convenios y sustituyendo los términos incrementales por sus símbolos sencillos, indicados a continuación

Vx → u → u; Mx → L → L (no confundir con fuerza sustentación)
Vy → v → v; My → M → M;
Vz → w → w; Mz → N → N;
$ù_x$ → p → p; ø’ → ø’;
$ù_y$ → q → q; ö’ → ö’;
$ù_z$ → r → r; è’ → è’;
Fx → X → X; etc.
Fy → Y → Y;
Fz → Z → Z;

las ecuaciones (2.17) se transforman en las siguientes:

$$\left.\begin{aligned}
&\Delta Fgx + \Delta Fax + \Delta Fpx = m(u' + Wq)\\
&\Delta Fgy + \Delta Fay + \Delta Fpy = m(v' + Ur - Wp)\\
&\Delta Fgz + \Delta Faz + \Delta Fpz = m(w' - Uq)\\
&\Delta Max + \Delta Mpx = Ix\, p' - Ixz\, r'\\
&\Delta May + \Delta Mpy = Iy\, q'\\
&\Delta Maz + \Delta Mpz = Iz\, r' - Ixz\, p'
\end{aligned}\right\} \quad (2.18)$$

en las que U, V, W representan las componentes de las velocidad lineal, estacionaria a lo largo de los ejes x, y, z, esto es Vsx, Vsy, Vsz respectivamente.

Y las ecuaciones (2.11) pueden escribirse como sigue:

$$\left.\begin{aligned} p &= \varphi' - \psi' \operatorname{sen}\theta_s \\ q &= \theta' \\ r &= \psi' \cos\theta_s \end{aligned}\right\} \quad (2.19)$$

Analizando las ecuaciones (2.17) o (2.18) se observa que, no solamente se ha conseguido simplificar estas ecuaciones, sino que además pueden tratarse separadamente como dos grupos independientes (desacoplados). Uno que representa el movimiento longitudinal del avión y el otro el movimiento lateral.

En la realidad, sin embargo, existe un cierto grado de acoplamiento, entre estos dos movimientos, debido a los efectos de las fuerzas aerodinámicas y propulsoras, incluidas en la parte izquierda de estas ecuaciones.

La separación afecta, por tanto, a las fuerzas gravitatorias y a las inerciales y es consecuencia de las condiciones hipotéticas de equilibrado del avión.

Los dos grupos separados de ecuaciones son:

$$\left.\begin{aligned} \Delta Fgx + \Delta Fax + \Delta Fpx &= m(u' + Wq) \\ \Delta Fgz + \Delta Faz + \Delta Fpz &= m(w' - Uq) \\ \Delta May + \Delta Mpy &= Iy\, q' \end{aligned}\right\} \textit{movimiento longitudinal} \quad (2.20)$$

$$\left.\begin{aligned} \Delta Fgy + \Delta Fay + \Delta Fpy &= m(v' + Ur - Wp) \\ \Delta Max + \Delta Mpx &= Ix\, p' - Ixz\, r' \\ \Delta Maz + \Delta Mpz &= Iz\, r' - Ixz\, p' \end{aligned}\right\} \textit{movimiento lateral} \quad (2.21)$$

Se observa que todas las variables, en el primer grupo de ecuaciones (2.20), se desarrollan en el plano vertical de simetría, y todas las variables, del segundo grupo (2.21), lo hacen fuera del plano de simetría. Solamente hay una excepción, a saber Vs, que es la velocidad del avión, antes de la perturbación, o antes del cambio de control, o de ambos.

Sustituyendo los términos gravitatorios (Fg_{*}) por sus valores correspondientes, especificados en las ecuaciones (2.10), las ecuaciones anteriores pueden ser expresadas como sigue.

$$\left.\begin{aligned} \Delta Fax + \Delta Fpx &= \sum \Delta X = m(u' + Wq + g\theta \cos\theta_s) \\ \Delta Faz + \Delta Fpz &= \sum \Delta Z = m(w' - Uq + g\theta \operatorname{sen}\theta_s) \\ \Delta May + \Delta Mpy &= \sum \Delta M = Iy\, q' \end{aligned}\right\} \textit{movimiento longitudinal} \quad (2.22)$$

$$\left.\begin{aligned} \Delta Fay + \Delta Fpy &= \sum \Delta Y = m(v' + Ur - g\varphi \cos\theta_s - Wp) \\ \Delta Max + \Delta Mpx &= \sum \Delta L = Ix\, p' - Ixz\, r' \\ \Delta Maz + \Delta Mpz &= \sum \Delta N = Iz\, r' - Ixz\, p' \end{aligned}\right\} \textit{movimiento lateral} \quad (2.23)$$

2.5.4 Introducción de los desplazamientos de los controles en las ecuaciones precedentes

El control del avión se efectúa mediante el desplazamiento de sus superficies de mando primarias es decir, alerones, timones de dirección y de profundidad. Las superficies de control secundarias como flaps, spoilers (aerofrenos) ... no serán consideradas, pues estas superficies sólo se despliegan de forma auxiliar en maniobras a baja velocidad (como son el despegue y aterrizaje).

El desplazamiento de las superficies de control produce fuerzas y momentos aerodinámicos, que generan finalmente el cambio de control deseado. Por consiguiente, las fuerzas y los momentos mencionados son función de esos desplazamientos. Es decir:

$$\begin{array}{ll} Fa = f(\ddot{a}_e, \ddot{a}_a, \ddot{a}_r) & \left\{ \begin{array}{l} \ddot{a}_e \rightarrow \text{desplaz. timón profundidad (elevator)} \\ \ddot{a}_r \rightarrow \text{desplaz. timón dirección (rudder)} \\ \ddot{a}_a \rightarrow \text{desplaz. alerones (ailerons)} \end{array} \right. \\ Ma = F(\ddot{a}_e, \ddot{a}_a, \ddot{a}_r) & \end{array}$$

Aplicando, ahora, el desarrollo de Taylor (ecuación (2.12)) a las ecuaciones del movimiento (2.22) y (2.23); por ejemplo aplicándolo a la primera ecuación del grupo (2.22), se puede escribir:

$$\begin{aligned} &\sum \Delta X = (\P X / \P u)u + (\P X / \P u')u' + (\P X / \P w)w + (\P X / \P w')w' + \\ &+(\P X / \P q)q + (\P X / \P q')q' + (\P X / \P d_e)d_e + (\P X / \P d_e')d_e' \qquad (2.24) \end{aligned}$$

La ecuación (2.24) supone que la fuerza Fax = X ha sido causada por el desplazamiento del timón de profundidad $\ddot{a}_e$ solamente.

Si se aplican otros cambios (perturbaciones o controles) también, por ejemplo cuando se hace uso de cambios de empuje (T), o deflexiones de los flaps y de los “spoilers” simétricos, la ecuación precedente debe ser completada con los términos adicionales respectivos, tales como

$$(\P X / \P d_T)d_T, \quad (\P X / \P d_f)d_f, \quad (\P X / \P d_{sp})d_{sp}, \quad etc.$$

Si se olvidan las variaciones en el empuje y en las posiciones de superficies de control secundarios se puede escribir:

$$\left.\begin{aligned}
&(\partial X/\partial u)u + (\partial X/\partial u')u' + (\partial X/\partial w)w + (\partial X/\partial w')w' + \\
&+(\partial X/\partial q)q + (\partial X/\partial q')q' + (\partial X/\partial \delta_e)\delta_e + (\partial X/\partial \delta_e')\delta_e' = \\
&= m(u' + Wq + g\theta\cos\theta_s) \\[1ex]
&(\partial Y/\partial v)v + (\partial Y/\partial v')v' + (\partial Y/\partial r)r + (\partial Y/\partial r')r' + \\
&+(\partial Y/\partial p)p + (\partial Y/\partial p')p' + (\partial Y/\partial \delta_a)\delta_a + (\partial Y/\partial \delta_a')\delta_a' = \\
&= m(v' + Ur - g\varphi\cos\theta_s - Wp) \\[1ex]
&(\partial Z/\partial u)u + (\partial Z/\partial u')u' + (\partial Z/\partial w)w + (\partial Z/\partial w')w' + \\
&+(\partial Z/\partial q)q + (\partial Z/\partial q')q' + (\partial Z/\partial \delta_e)\delta_e + (\partial Z/\partial \delta_e')\delta_e' = \\
&= m(w' - Uq + g\theta\, sen\theta_s) \\[1ex]
&(\partial L/\partial v)v + (\partial L/\partial v')v' + (\partial L/\partial r)r + (\partial L/\partial r')r' + \\
&+(\partial L/\partial p)p + (\partial L/\partial p')p' + (\partial L/\partial \delta_a)\delta_a + (\partial L/\partial \delta_r)\delta_r = \\
&= Ix\, p' - Ixz\, r' \\[1ex]
&(\partial M/\partial u)u + (\partial M/\partial u')u' + (\partial M/\partial w)w + (\partial M/\partial w')w' + \\
&+(\partial M/\partial q)q + (\partial M/\partial q')q' + (\partial M/\partial \delta_e)\delta_e + (\partial M/\partial \delta_e')\delta_e' = \\
&= Iy\, q' \\[1ex]
&(\partial N/\partial v)v + (\partial N/\partial v')v' + (\partial N/\partial r)r + (\partial N/\partial r')r' + \\
&+(\partial N/\partial p)p + (\partial N/\partial p')p' + (\partial N/\partial \delta_a)\delta_a + (\partial N/\partial \delta_r)\delta_r = \\
&= Iz\, r' - Ixz\, p'
\end{aligned}\right\} \qquad (2.25)$$

Continuando con el proceso de simplificación, se tendrá en cuenta que algunas de las derivadas son muy pequeñas y pueden despreciarse de acuerdo con la experiencia acumulada, acerca de los datos aerodinámicos de muchos aviones. Por ello, y suponiendo que se estudia a un avión de tipo convencional, no se tendrán en cuenta a los siguientes términos:

$$\begin{aligned}
&(\partial X/\partial u'),\ (\partial X/\partial q),\ (\partial X/\partial q'),\ (\partial X/\partial w'),\ (\partial X/\partial \delta_e'), \\
&(\partial Y/\partial v'),\ (\partial Y/\partial r),\ (\partial Y/\partial r'),\ (\partial Y/\partial p),\ (\partial Y/\partial p'),\ (\partial Y/\partial \delta_a), \\
&(\partial Z/\partial u'),\ (\partial Z/\partial w'),\ (\partial Z/\partial \delta_e'), \\
&(\partial L/\partial v'),\ (\partial L/\partial r'),\ (\partial L/\partial p'),\ (\partial M/\partial u'),\ (\partial M/\partial \delta_e'), \\
&(\partial N/\partial v'),\ (\partial N/\partial r').
\end{aligned}$$

2.5.5 Definición e introducción de las Derivadas de Estabilidad

Para simplificar más la notación se hace uso de las sustituciones siguientes y se definen las “derivadas de estabilidad”.

$$Xu = (1/m)(\partial X/\partial u); \quad Xw = (1/m)(\partial X/\partial w); \quad X_{\delta e} = (1/m)(\partial X/\partial \delta_e)$$
$$Yv = (1/m)(\partial Y/\partial v); \quad Y_{\delta_r} = (1/m)(\partial Y/\partial \delta_r)$$
$$Zu = (1/m)(\partial Z/\partial u); \quad Zw = (1/m)(\partial Z/\partial w); \quad Z_{\delta_r} = (1/m)(\partial Z/\partial \delta_r);$$
$$Lv = (1/Ix)(\partial L/\partial v); \quad Lp = (1/Ix)(\partial L/\partial p); \quad Lr = (1/Ix)(\partial L/\partial r);$$
$$L_{\delta_r} = (1/Ix)(\partial L/\partial \delta_r); \quad L_{\delta a} = (1/Ix)(\partial L/\partial \delta_a);$$
$$Mu = (1/Iy)(\partial M/\partial u); \quad Mw = (1/Iy)(\partial M/\partial w); \quad Mw' = (1/Iy)(\partial M/\partial w');$$
$$Mq = (1/Iy)(\partial M/\partial q); \quad M_{\delta e} = (1/Iy)(\partial M/\partial \delta_e);$$
$$Nv = (1/Iz)(\partial N/\partial v); \quad Np = (1/Iz)(\partial N/\partial p); \quad Np' = (1/Iz)(\partial N/\partial p');$$
$$N_{\delta a} = (1/Iz)(\partial N/\partial \delta_a); \quad N_{\delta_r} = (1/Iz)(\partial N/\partial \delta_r);$$

Nota: las derivadas de estabilidad, así definidas son unos términos dimensionales (también se definen, a veces, derivadas de estabilidad adimensionales). Todas las derivadas de estabilidad son importantes, pero algunas son más importantes que otras, para el control del vuelo. Las derivadas de estabilidad dependen no sólo del avión particular en consideración, sino también de las condiciones específicas de vuelo. En consecuencia antes de despreciar, en los cálculos, a algunas de ellas, deberán comprobarse con cuidado los datos aerodinámicos aplicables.

El valor de la derivada $\partial Z/\partial q$ puede ser importante, aunque suele despreciarse cuando la velocidad compensada U es elevada. Cuando se trata del movimiento (sin avance) de un helicóptero $\partial Z/\partial q$ no puede ignorarse. Además $\partial Y/\partial r$ puede ser también importante, cuando U es pequeña.

Bajo todas estas condiciones y supuestos, las ecuaciones (2.25) pueden expresarse como sigue:

$$\left.\begin{array}{l}
(\partial X/\partial u)u + (\partial X/\partial w)w + (\partial X/\partial \delta_e)\delta_e = m(u' + Wq + gq\cos\theta_s) \\
(\partial Y/\partial v)v + (\partial Y/\partial \delta_r)\delta_r == m(v' + Ur - g\varphi\cos\theta_s - Wp) \\
(\partial Z/\partial u)u + (\partial Z/\partial w)w + (\partial Z/\partial \delta_e)\delta_e = m(w' - Uq + gq\, sen\theta_s) \\
\\
(\partial L/\partial v)v + (\partial L/\partial r)r + (\partial L/\partial p)p + (\partial L/\partial \delta_a)\delta_a + (\partial L/\partial \delta_r)\delta_r = Ix\,p' - Ixz\,r' \\
(\partial M/\partial u)u + (\partial M/\partial w)w + (\partial M/\partial w')w' + (\partial M/\partial q)q + (\partial M/\partial \delta_e)\delta_e = Iy\,q' \\
(\partial N/\partial v)v + (\partial N/\partial r)r + (\partial N/\partial p)p + (\partial N/\partial \delta_a)\delta_a + (\partial N/\partial \delta_r)\delta_r = Iz\,r' - Ixz\,p'
\end{array}\right\} \quad (2.26)$$

Y haciendo uso de la notación de las derivadas de estabilidad dimensionales, las ecuaciones precedentes se transforman en las siguientes:

$$\left.\begin{array}{l}
Xu\,u + Xw\,w + X_{\delta e}\delta_e = (u' + Wq + gq\cos\theta_s) \\
Yv\,v + Y_{\delta_r}\delta_r = v' + Ur - g\varphi\cos\theta_s - Wp \\
Zu\,u + Zw\,w + Z_{\delta e}\delta_e = w' - Uq + gq\, sen\theta_s \\
Lv\,v + Lp\,p + Lr\,r + L_{\delta a}\delta_a + L_{\delta r}\,\delta_r = p' - (Ixz/Ix)\,r' \\
Mu\,u + Mw\,w + Mw'\,w' + Mq\,q + M_{\delta e}\delta_e = q' \\
Nv\,v + Nr\,r + Np\,p + N_{\delta a}\delta_a + N_{\delta_r}\delta_r = r' - (Ixz/Iz)\,p'
\end{array}\right\} \quad (2.27)$$

o también

$$\left.\begin{array}{l} u' = Xu\,u + Xw\,w + X_{¶e}¶_e \quad - Wq - gq\,\cos q_s \\ v' = Yv\,v + Y_{d_r} d_r - Ur + gj\,\cos q_s + Wp \\ w' = Zu\,u + Zw\,w + Z_{de} d_e + Uq - gq\,senq_s \\ p' = Lv\,v + Lp\,p + Lr\,r + L_{da} d_a + L_{dr}\,d_r + (Ixz\,/\,Ix)\,r' \\ q' = Mu\,u + Mw\,w + Mw'\,w' + Mq\,q + M_{de} d_e \\ r' = Nv\,v + Nr\,r + Np\,p + N_{da} d_a + N_{d_r} d_r + (Ixz\,/\,Iz)\,p' \end{array}\right\} \quad (2.28)$$

Este conjunto de ecuaciones puede ser desglosado en dos grupos, que corresponden al movimiento longitudinal perturbado y al movimiento lateral/direccional del avión. Esto es:

MOVIMIENTO LONGITUDINAL

$$\left.\begin{array}{l} u' = Xu\,u + Xw\,w + X_{¶e}¶_e \quad - Wq - gq\,\cos q_s \\ w' = Zu\,u + Zw\,w + Z_{de} d_e + Uq - gq\,senq_s \\ q' = Mu\,u + Mw\,w + Mw'\,w' + Mq\,q + M_{de} d_e \end{array}\right\} \quad (2.29)$$

completando este grupo de ecuaciones con

$q' = q$ (tomada de las ecuaciones (2.19))

MOVIMIENTO LATERAL

$$\left.\begin{array}{l} v' = Yv\,v + Y_{d_r} d_r - Ur + gj\,\cos q_s + Wp \\ p' = Lv\,v + Lp\,p + Lr\,r + L_{da} d_a + L_{dr}\,d_r + (Ixz\,/\,Ix)\,r' \\ r' = Nv\,v + Nr\,r + Np\,p + N_{da} d_a + N_{d_r} d_r + (Ixz\,/\,Iz)\,p' \end{array}\right\} \quad (2.30)$$

completando este grupo de ecuaciones con

$p = j' - y'\,senq_s$

$r = y'\cos q_s$ (también tomadas de las ecuaciones (2.19))

estas dos ecuaciones se pueden escribir también como:

$j' = p + r\tan q_s$

$y' = r\sec q_s$

Los términos del lado izquierdo de estos grupos de ecuaciones tienen dimensiones de aceleración. En consecuencia, los términos de las derivadas de estabilidad, que figuran en el lado derecho son también dimensionales, según se había ya indicado.

2.5.6 Ecuaciones del movimiento del avión en el sistema de ejes de estabilidad

Las componentes de la velocidad a lo largo de los ejes-cuerpo del avión son las indicadas en las ecuaciones (2.13).

La posición del eje-cuerpo Xb, en condiciones estacionarias, apunta en la dirección del viento relativo. Es decir, el vector velocidad V y el eje Xb están alineados.

Esta condición implica que, inicialmente, el sistema de ejes-estabilidad puede estar inclinado, un cierto ángulo de "trayectoria" o de senda de vuelo ("flight path"), respecto al horizonte. A este ángulo se denominará ã$_S$, de forma que, como muestra la figura 2.13 se verifica $g = q - a$.

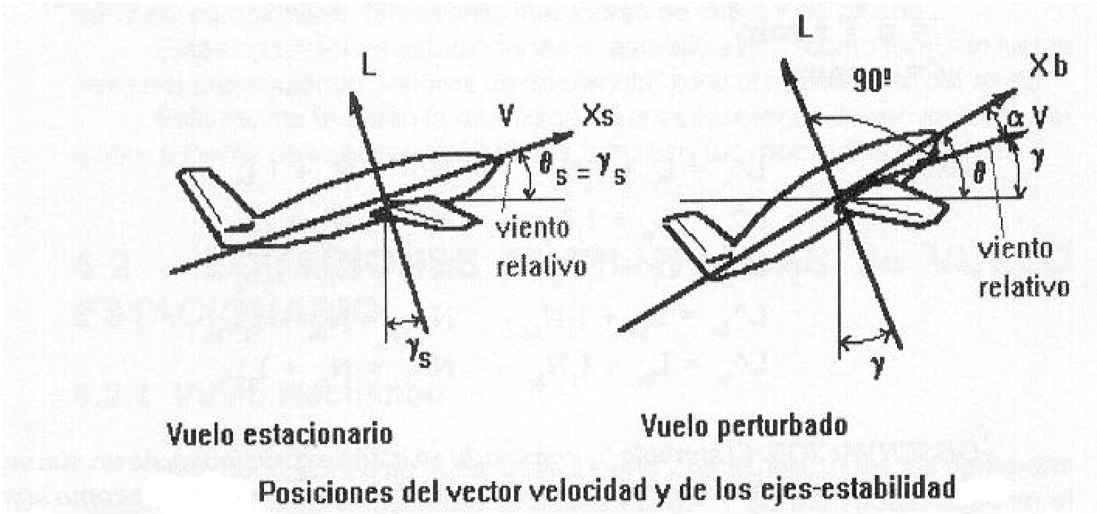


figura 2.13 (compendio de avionica digital)

Inicialmente (condición estacionaria) el eje Xb está orientado de forma que W=Ws=0; y entonces á$_S$ = 0 y è$_S$ = ã$_S$

Teniendo en cuenta esto, se pueden rescribir las ecuaciones (2.29) y (2.30) como sigue:

MOVIMIENTO LONGITUDINAL

$$\left.\begin{aligned} ' &= Xu\,u + \quad w + X_{¶e}¶_e \quad - \quad q\cos g \\ w' &= Zu \quad + Zw\,w + \quad {}_{de}d_e + Uq - gq \qquad g \\ q' &= Mu \quad + Mw\,w + \quad w'\,w' + \qquad q + M_{de}d_e \\ q\,' &= \end{aligned}\right\} \qquad .31)$$

MOVIMIENTO LATERAL

$$\left.\begin{aligned} v' &= Yv\,v + Y_{d_r}d_r - Ur + gj\,\cos g_s \\ p' &= Lv\,v + Lp\,p + Lr\,r + L_{da}d_a + L_{dr}\,d_r + (Ixz/Ix)\,r' \\ r' &= Nv\,v + Nr\,r + Np\,p + N_{da}d_a + N_{d_r}d_r + (Ixz/Iz)\,p' \\ j\,' &= p + r\tan g_s \\ y\,' &= r\sec g_s \end{aligned}\right\} \qquad (2.32)$$

los términos de inercia Ixz/Ix e Ixz/Iz , de estas ecuaciones del movimiento lateral, se expresarán como I_1 e I_2 respectivamente. Ahora, mediante operaciones matemáticas simples y despreciando los términos de inercia de segundo grado (I_1I_2), se pueden expresar las ecuaciones del movimiento lateral como sigue:

$$\left.\begin{aligned}
v' &= Yv\, v + Y_{d_r} d_r - Ur + gj \cos g_s \\
p' &= L\hat{}v\, v + L\hat{}p\, p + L\hat{}r\, r + L\hat{}_{da} d_a + L\hat{}_{dr}\, d_r + (Ixz / Ix)\, r' \\
r' &= N\hat{}v\, v + N\hat{}r\, r + N\hat{}p\, p + N\hat{}_{da} d_a + N\hat{}_{d_r} d_r + (Ixz / Iz)\, p' \\
j' &= p + r \tan g_s \\
y' &= r \sec g_s
\end{aligned}\right\} \quad (2.32\ bis)$$

$$\begin{aligned}
\textit{siendo}\quad & L\hat{}v = Lv + I_1 Nv & N\hat{}v = Nv + I_2 Lv \\
& L\hat{}p = Lp + I_1 Np & N\hat{}p = Np + I_2 Lp \\
& L\hat{}r = Lr + I_1 Nr & N\hat{}r = Nr + I_2 Lr \\
& L\hat{}_{da} = L_{da} + I_1 N_{da} & N\hat{}_{da} = N_{da} + I_2 L_{da} \\
& L\hat{}_{d_r} = L_{d_r} + I_1 N_{d_r} & N\hat{}_{d_r} = N_{d_r} + I_2 L_{d_r}
\end{aligned}$$

2.6 DINÁMICA DEL AVIÓN EN EL ESPACIO DE ESTADOS

2.6.1 Introducción al espacio de estados

Si se tiene un sistema lineal, o se linealizan las ecuaciones que representan nuestro sistema alrededor del estado de operación, se podrán escribir las siguientes ecuaciones de estado y de salida linealizadas:

$$\begin{aligned}
\dot{x}(t) &= A(t) \cdot x(t) + B(t) \cdot u(t) \\
y(t) &= C(t) \cdot x(t) + D(t) \cdot u(t)
\end{aligned}$$

donde

A(t) matriz de estado
B(t) matriz de entrada
C(t) matriz de salida
D(t) matriz de transmisión directa

Si las matrices A, B, C y D no involucran el tiempo t explícitamente, el sistema se denomina sistema invariante con el tiempo. En este caso las ecuaciones son:

$$\begin{aligned}
\dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \\
y(t) &= C \cdot x(t) + D \cdot u(t)
\end{aligned}$$

En la figura 2.14 se representa el diagrama de bloques del sistema de control lineal en tiempo continuo representado en el espacio de estados.

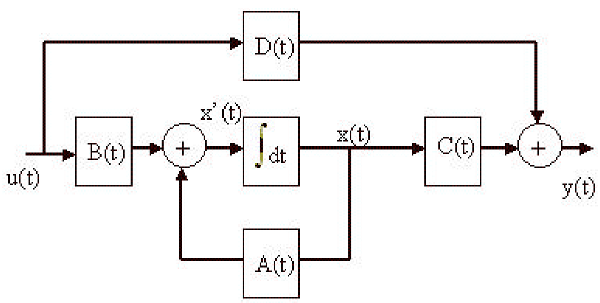


figura 2.14

2.6.2 Movimiento longitudinal del avión

Teniendo presentes las ecuaciones (2.31), el “vector de estado” puede escribirse:

$$\mathbf{X} = \left[u\ w\ q\ q\right]^T$$

donde: u es la velocidad lineal (m/s) en el eje longitudinal, w es la velocidad lineal (m/s) en el eje normal, q es la velocidad angular (rad/s) en el eje lateral, y è es el ángulo (rad) de cabeceo.

Suponiendo que al avión se le aplica, solamente, un cambio de control por medio del timón de profundidad $ä_e$, el “vector de actuación” puede expresarse por **u**=[$ä_e$].

Asimismo, mediante las ecuaciones (2.31), pueden determinarse las matrices A y B, a saber:

$$A = \begin{bmatrix} Xu & Xw & 0 & -g\cos g_s \\ Zu & Zw & U & -g\,seng_s \\ M\hat{\ }u & M\hat{\ }w & M\hat{\ }q & M\hat{\ }q \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} X_{de} \\ Z_{de} \\ M\hat{\ }_{de} \\ 0 \end{bmatrix} \qquad (2.33)$$

$$\begin{aligned} \textit{siendo} \qquad & M\hat{\ }u = (Mu + Mw'\,Zu) \\ & M\hat{\ }w = (Mw + Mw'\,Zw) \\ & M\hat{\ }q = (Mq + UMw') \\ & M\hat{\ }q = (-gMw'\,seng_s) \\ & M\hat{\ }_{de} = (M_{de} + Mw'\,Z_{de}) \end{aligned}$$

En consecuencia, la “ecuación de estado” del movimiento longitudinal del avión, bajo las condiciones especificadas previamente, será:

$$\mathbf{X}' = \begin{bmatrix} Xu & Xw & 0 & -g\cos g_s \\ Zu & Zw & U & -g\,seng_s \\ M\hat{\ }u & M\hat{\ }w & M\hat{\ }q & M\hat{\ }q \\ 0 & 0 & 1 & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} X_{de} \\ Z_{de} \\ M\hat{\ }_{de} \\ 0 \end{bmatrix} .\mathbf{u}$$

Nótese que en el caso de existir entradas adicionales de control, la matriz B (“matriz controladora”) necesita ser ampliada apropiadamente.

2.6.3 Movimiento lateral del avión

Teniendo presentes las ecuaciones (2.32 bis), el “vector de estado” puede escribirse:

$$\mathbf{X} = \left[v\ r\ p\ j\ \ y\right]^T$$

donde: v es la velocidad lineal (m/s) en el eje lateral, r es la velocidad angular (rad/s) en el eje normal, p es la velocidad angular (rad/s) en el eje longitudinal, ö es el ángulo (rad) de alabeo, y ø es el ángulo (rad) de guiñada.

Si se supone, ahora, que el avión es controlado por medio de un cambio de los alerones y del timón de dirección, $ä_a$ y $ä_r$ respectivamente, el “vector de actuaciones” será:

$$\mathbf{u} = \left[d_a \;\; d_r \right]^T$$

De las ecuaciones (2.32 bis) se pueden deducir las matrices A y B:

$$A = \begin{bmatrix} Yv & 0 & -U & g\cos g_s & 0 \\ L^\wedge v & L^\wedge p & L^\wedge r & 0 & 0 \\ N^\wedge v & N^\wedge p & N^\wedge r & 0 & 0 \\ 0 & 1 & \tan g_s & 0 & 0 \\ 0 & 0 & \sec g_s & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & Y_{d_r} \\ L^\wedge{}_{da} & L^\wedge{}_{d_r} \\ N^\wedge{}_{da} & N^\wedge{}_{d_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (2.34)$$

por tanto

$$\mathbf{X}' = \begin{bmatrix} Yv & 0 & -U & g\cos g_s & 0 \\ L^\wedge v & L^\wedge p & L^\wedge r & 0 & 0 \\ N^\wedge v & N^\wedge p & N^\wedge r & 0 & 0 \\ 0 & 1 & \tan g_s & 0 & 0 \\ 0 & 0 & \sec g_s & 0 & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} 0 & Y_{d_r} \\ L^\wedge{}_{da} & L^\wedge{}_{d_r} \\ N^\wedge{}_{da} & N^\wedge{}_{d_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} .\mathbf{u}$$

Es importante observar que, bajo las hipótesis hechas, todos los términos de la quinta columna de la matriz A son cero. Esto implica que en esta matriz existe una columna que es combinación lineal de las cuatro restantes. De hecho, si se examinan detalladamente, el conjunto de ecuaciones (2.32 bis), se observa, por ejemplo, que ø’ está relacionada directamente con la variable r, y en consecuencia estas cinco ecuaciones pueden reducirse a cuatro.

Por tanto, la ecuación de estado del movimiento lateral del avión puede ser expresada, bajo las hipótesis hechas, por la expresión modificada siguiente

$$\mathbf{X}' = \begin{bmatrix} Yv & 0 & -U & g\cos g_s & 0 \\ L^\wedge v & L^\wedge p & L^\wedge r & 0 & 0 \\ N^\wedge v & N^\wedge p & N^\wedge r & 0 & 0 \\ 0 & 1 & \tan g_s & 0 & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} 0 & Y_{d_r} \\ L^\wedge{}_{da} & L^\wedge{}_{d_r} \\ N^\wedge{}_{da} & N^\wedge{}_{d_r} \\ 0 & 0 \end{bmatrix} .\mathbf{u}$$

donde $\mathbf{X} = \left[v \; r \;\; p \; j \;\; \right]^T$

Con frecuencia, se emplea al ángulo de derrape, â, como una variable de estado, en vez de la velocidad v de derrape del avión. Con ayuda de las ecuaciones (2.13) se puede escribir, para ángulos pequeños:

$$v = U\, senb \;\cong\; Ub$$

Reemplazando este valor de v, en las ecuaciones (2.32 bis), se tiene:

$$b' = Yv\; b + (Y_{d_r}\; / U) d_r - r + (j \;\; g / U) \cos g_s$$

Con esta ecuación, las matrices A y B dadas en (2.34) se transforman en:

$$A=\begin{bmatrix} Yv & 0 & -1 & (g/U)\cos\gamma_s & 0 \\ L'_\beta & L'p & L'r & 0 & 0 \\ N'_\beta & N'p & N'r & 0 & 0 \\ 0 & 1 & \tan\gamma_s & 0 & 0 \\ 0 & 0 & \sec\gamma_s & 0 & 0 \end{bmatrix} \qquad B=\begin{bmatrix} 0 & Y_{\delta_r}/U \\ L'_{\delta a} & L'_{\delta_r} \\ N'_{\delta a} & N'_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad (2.35)$$

donde $L'_\beta = U\, L'v$; $N'_\beta = U\, N'v$

Lógicamente el vector de estados se ha modificado, pudiéndose escribir:

$$\mathbf{X}' = \begin{bmatrix} Yv & 0 & -1 & (g/U)\cos\gamma_s & 0 \\ L'_\beta & L'p & L'r & 0 & 0 \\ N'_\beta & N'p & N'r & 0 & 0 \\ 0 & 1 & \tan\gamma_s & 0 & 0 \\ 0 & 0 & \sec\gamma_s & 0 & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} 0 & Y_{\delta_r}/U \\ L'_{\delta a} & L'_{\delta_r} \\ N'_{\delta a} & N'_{\delta_r} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} .\mathbf{u}$$

siendo $\mathbf{X} = \begin{bmatrix}\beta & r & p & \varphi & \psi\end{bmatrix}^T$

O simplificando como ya se hizo anteriormente:

$$\mathbf{X}' = \begin{bmatrix} Yv & 0 & -1 & (g/U)\cos\gamma_s \\ L'_\beta & L'p & L'r & 0 \\ N'_\beta & N'p & N'r & 0 \\ 0 & 1 & \tan\gamma_s & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} 0 & Y_{\delta_r}/U \\ L'_{\delta a} & L'_{\delta_r} \\ N'_{\delta a} & N'_{\delta_r} \\ 0 & 0 \end{bmatrix} .\mathbf{u} \qquad (2.36)$$

siendo $\mathbf{X} = \begin{bmatrix}\beta & r & p & \varphi\end{bmatrix}^T$

2.6.4 Matrices de estado con datos de un avión real

A continuación se analiza un caso real de un avión comercial de pasajeros de cuatro motores reactores, determinando sus matrices A y B tanto para el movimiento longitudinal como lateral. Estas matrices serán las utilizadas en los estudios posteriores llevados a cabo en este proyecto.

Los parámetros generales correspondientes a este avión son [3]:

	Fase de aproxim.	Otras fases de vuelo
Peso (kg)	250000	290000
Inercias (Kg m^2)		
Ix	$18.6\ 10^6$	$24.6\ 10^6$
Iy	$41.35\ 10^6$	$45\ 10^6$
Iz	$58\ 10^6$	$67.5\ 10^6$
Ixz	$1.2\ 10^6$	$1.32\ 10^6$
Superficie alar (m^2)	510	

“Aspect ratio”	7.0
Cuerda c(m)	8.3
Total Empuje (kN)	900
Centro de Gravedad	0.25c
Ubicación del piloto (m) respecto a C.G	
Ixp	26.2
Izp	-3.05

Condiciones de vuelo:

Parámetros	Fase crucero	Nivel del mar
Altura (m)	12200	S.L (Sea Level)
Mach No.	0.8	0.198
U_0 (m/seg)	250	67
$\bar{q}$ =rU^2/2 (M/m^2)	9911	2810
á_s (grados)	4.6	8.5
ã_s	0	0

Derivadas dimensionales de Estabilidad: (Fase crucero):

Movimiento Longitudinal		Movimiento Lateral	
Xu	0.0002	Yv	-0.056
Xw	0.039	$Y_{\text{är}}$/U	0.012
$X_{\text{äe}}$	0.44	$L\text{^}_{\text{â}}$	-1.05
Zu	-0.07	L^p	-0.47
Zw	-0.317	L^r	0.39
Zq	-1.57	$L\text{^}_{\text{äa}}$	0.14
$Z_{\text{äe}}$	-5.46	$L\text{^}_{\text{är}}$	0.15
Mu	0.00006	$N\text{^}_{\text{â}}$	0.6
Mw	-0.003	N^p	-0.032
Mw’	-0.0004	N^r	-0.115
Mq	-0.339	$N\text{^}_{\text{äa}}$	0.008
$M_{\text{äe}}$	-1.16	$N\text{^}_{\text{är}}$	-0.48

MOVIMIENTO LONGITUDINAL

Con estos datos de la fase crucero y acudiendo a las ecuación (2.33) se deduce para el movimiento longitudinal:

$$\begin{bmatrix} u \\ w \\ q \\ q \end{bmatrix}' = \begin{bmatrix} 0.0002 & 0.039 & 0 & -9.81 \\ -0.07 & -0.317 & 250 & 0 \\ 0.000088 & -0.0028732 & -0.439 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} u \\ w \\ q \\ q \end{bmatrix} + \begin{bmatrix} 0.44 \\ -5.46 \\ -1.14362 \\ 0 \end{bmatrix} [d_e] \quad (2.37)$$

MOVIMIENTO LATERAL

Con los datos de la fase crucero, acudiendo a las ecuaciones (2.36) se deduce para el movimiento lateral:

$$\begin{bmatrix} b \\ r \\ p \\ j \end{bmatrix}' = \begin{bmatrix} -0.056 & 0 & -1 & 0.03924 \\ -1.05 & -0.47 & 0.39 & 0 \\ 0.6 & -0.032 & -0.115 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} b \\ r \\ p \\ j \end{bmatrix} + \begin{bmatrix} 0 & 0.012 \\ 0.14 & 0.15 \\ 0.008 & -0.48 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} d_a \\ d_r \end{bmatrix} \quad (2.38)$$

2.7 REPRESENTACIÓN EXTERNA DE LA DINÁMICA DEL AVIÓN

2.7.1 Funciones de transferencia en tiempo continuo

A partir de la representación interna (en el espacio de estados) de la dinámica del avión se puede obtener una representación externa (en funciones de transferencia) equivalente. Para ello se emplean las siguientes funciones de Matlab en la ventana de trabajo de dicho programa tras haber introducido los valores de las matrices A, B, C y D tanto para la dinámica longitudinal como lateral:

```
sistema_ss=ss(A,B,C,D);
sistema_tf=tf(sistema_ss);
```

Con esto se obtiene:

DINÁMICA LONGITUDINAL

Función de transferencia entre el ángulo de cabeceo del avión y la deflexión del timón de profundidad:

$$G(S) = \frac{q}{d_e} = \frac{-1.144S^2 - 0.3466S - 0.002971}{S^4 + 0.7558S^3 + 0.86S^2 + 0.001032S + 0.002247} \quad (2.39)$$

Atendiendo a la función de transferencia (2.39) y usando el comando “pole” de Matlab, se obtienen los siguiente modos longitudinales:

Corto periodo: -0.378453 ± 0.845597i → $ù_{sp}$=0.926424rad/seg $î_{sp}$=0.408509
Fugoide : 0.000553 ± 0.051160i → $ù_{ph}$=0.051164rad/seg $î_{ph}$=0.010807

DINÁMICA LATERAL

Función de transferencia entre el ángulo de derrape del avión y la deflexión de los alerones:

$$G_{11}(S) = \frac{b}{d_a} = \frac{-0.008S^2 + 0.006214S + 0.0007542}{S^4 + 0.641S^3 + 0.6993S^2 + 0.3605S - 0.004444} \quad (2.40)$$

Función de transferencia entre el ángulo de alabeo del avión y la deflexión de los alerones:

$$G_{21}(S) = \frac{j}{d_a} = \frac{0.14S^2 + 0.02706S + 0.09348}{S^4 + 0.641S^3 + 0.6993S^2 + 0.3605S - 0.004444} \quad (2.41)$$

Función de transferencia entre el ángulo de derrape del avión y la deflexión del timón de dirección:

$$G_{12}(S) = \frac{b}{d_r} = \frac{0.012S^3 + 0.487S^2 + 0.2371S - 0.006669}{S^4 + 0.641S^3 + 0.6993S^2 + 0.3605S - 0.004444} \quad (2.42)$$

Función de transferencia entre el ángulo de alabeo del avión y la deflexión del timón de dirección:

$$G_{22}(S) = \frac{j}{d_r} = \frac{0.15S^2 - 0.1742S - 0.4222}{S^4 + 0.641S^3 + 0.6993S^2 + 0.3605S - 0.004444} \quad (2.43)$$

Atendiendo al denominador de cualquiera de las cuatro funciones de transferencia anteriores y usando el comando “pole” de Matlab se obtienen los siguientes modos laterales:

Apaciguamiento de alabeo: -0.56241
Espiral: 0.01204 (>0 divergente)
Balanceo del holandés:-0.04532 ± 0.80878 → $ù_{dr}$= 0.81005rad/seg $î_{dr}$= 0.05594

2.7.2 Funciones de transferencia en tiempo discreto

Para un posterior análisis es más útil la representación interna, en tiempo discreto del sistema. Para ello se procede a la discretización de las funciones de transferencia anteriores usando un mantenedor de orden cero “sampling and hold”. Esto se consigue con el comando “c2d” de Matlab, obteniendo para un tiempo de muestreo de 0.008 segundos (suficientemente pequeño como se explicará en el subapartado 3.2.1.2) las siguientes funciones de transferencia discretas:

DINÁMICA LONGITUDINAL

$$G(z) = \frac{q}{d_e} = \frac{(-3.655z^3 + 3.651z^2 + 3.646z - 3.642) * 10^{-5}}{z^4 - 3.994z^3 + 5.982z^2 - 3.982z + 0.994} \quad (2.44)$$

DINÁMICA LATERAL

Función de transferencia entre el ángulo de derrape del avión y la deflexión de los alerones:

$$G_{11}(z) = \frac{b}{d_a} = \frac{(-2.55z^3 + 2.576z^2 + 2.531z - 2.557) * 10^{-7}}{z^4 - 3.995z^3 + 5.985z^2 - 3.985z + 0.9949} \quad (2.45)$$

Función de transferencia entre el ángulo de alabeo del avión y la deflexión de los alerones:

$$G_{21}(z) = \frac{b}{d_a} = \frac{(-4.475z^3 - 4.473z^2 - 4.464z + 4.462) * 10^{-6}}{z^4 - 3.995z^3 + 5.985z^2 - 3.985z + 0.9949} \quad (2.46)$$

Función de transferencia entre el ángulo de derrape del avión y la deflexión del timón de dirección:

$$G_{12}(z) = \frac{b}{d_r} = \frac{(11.13z^3 - 30.28z^2 + 27.17z - 8.024) * 10^{-5}}{z^4 - 3.995z^3 + 5.985z^2 - 3.985z + 0.9949} \quad (2.47)]$$

Función de transferencia entre el ángulo de alabeo del avión y la deflexión del timón de dirección:

$$G_{22}(z) = \frac{b}{d_r} = \frac{(4.777z^3 - 4.845z^2 - 4.732z + 4.798) * 10^{-6}}{z^4 - 3.995z^3 + 5.985z^2 - 3.985z + 0.9949} \quad (2.48)$$

Posteriormente para cada tipo de control analizado se elegirá justificadamente un tiempo de muestreo apropiado, distinto por lo general al usado aquí.

Capítulo 3

Control LQR de una aeronave en fase crucero

3.1 INTRODUCCIÓN AL CONTROL ÓPTIMO

Es sabido que los métodos de control basados en realimentación de estados son más versátiles y potentes que los correspondientes a la teoría del control clásico (lugar de las raíces, Nyquist, Bode ...). En particular, permiten la asignación de los autovalores de lazo cerrado a voluntad. También se sabe que estos métodos pueden ser complementados con técnicas de observación de estados para sobrellevar limitaciones (económicas y/o físicas) en la medida de los mismos. Sin embargo, ni los métodos de diseño de control clásico, ni los de realimentación de estados para la asignación de autovalores, se pone explícitamente de manifiesto el compromiso que existe entre la satisfacción de las especificaciones dinámicas y el bajo "costo" para poder cumplirlas. Por ejemplo, entre la velocidad deseada de respuesta de la variable controlada y una acción de control no demasiado agresiva. Surge entonces el concepto de control óptimo.

En términos de control se dice que un control es óptimo si minimiza un funcional de costo en el que claramente se manifiesta un compromiso entre distintas especificaciones y restricciones. A este funcional se le llamará índice de comportamiento [5] y normalmente se identifica con la letra J. Obviamente, el mismo controlador evaluado con otro índice de comportamiento J_1, no será óptimo si lo era con J.

El problema conocido como LQR (Linear Quadratic Regulator) es un problema de regulación, que se resuelve a partir de la optimización de un índice de rendimiento cuadrático, dando lugar a una ley de control lineal. El índice de rendimiento es:

$$J = x^T(tf)P(0)x(tf) + \int_{t0}^{tf} (x^T Qx + u^T Ru)dt$$

donde x es el vector de estados del sistema elegidos como errores con respecto a los valores de estado estacionario. Las matrices de peso Q, R y P(0) son matrices reales, simétricas, constantes y definidas positivas.

En muchos problemas se considera tf = , es decir horizonte infinito. En este caso, para que J sea finito necesariamente x(tf) = 0, por lo que el índice de comportamiento se reduce a:

$$J = \int_{t0}^{\infty} (x^T Qx + u^T Ru)dt$$

Para el caso de sistemas discretos el índice de comportamiento es :

$$J = x^T(N)P(0)x(N) + \sum_{k=0}^{N-1} (x^T(k)Qx(k) + u^T(k)Ru(k))$$

Y para el caso de horizonte infinito (tf →):

$$J = \sum_{k=0}^{\infty} (x^T(k)Qx(k) + u^T(k)Ru(k))$$

Para el caso continuo, la solución del regulador óptimo de horizonte infinito se puede resumir en una realimentación del vector de estados con ganancia K constante:

$$\begin{aligned} &u(t) = -Kx(t) \quad \text{donde} \quad K = R^{-1}B^TP \quad \text{siendo P la solución de la ec. de Riccati} \\ &\qquad Q - PBR^{-1}B^TP + PA + A^TP = 0 \qquad (3.1) \end{aligned}$$

donde A y B son las matrices que definen la dinámica del sistema $\dot{x}(t) = Ax(t) + Bu(t)$

Para el caso discreto y horizonte infinito la solución es también una realimentación del estado con ganancia constante:

$$\begin{aligned} &u(k) = -Kx(k) \\ &K = [R + B^TPB]^{-1}B^TPA \qquad (3.2) \\ &P = [A + BK]^T P[A + BK] + K^TRK + Q \quad \text{(Ec. de Ricatti en tiempo discreto)} \end{aligned}$$

donde A y B son las matrices que definen la dinámica del sistema en tiempo discreto:

$$x(k+1) = Ax(k) + Bu(k)$$

3.2 APLICACIONES DEL CONTROL LQR EN AERONONÁUTICA

Muchos de los aviones en activo usan leyes de control desarrolladas bajo el uso de técnicas clásicas de respuesta en frecuencia con lazos simples, apoyadas por el diseño mediante el lugar de las raíces.

En los últimos 20 años se han propuesto nuevas técnicas de análisis y control multivariable. Estas técnicas pueden tratar los lazos de control múltiples de manera sistemática. La primera aplicación significante del control multivariable en aeronáutica comenzó en Boeing en 1978. Dicho trabajo se debe a la NASA bajo la dirección del proyecto "Integrated Application of Active Controls (IAAC) Technology to an Advanced Subsonic Transport". Los resultados de dicho proyecto pusieron en evidencia las ventajas del control multivariable frente a las técnicas de control clásicas. Fundamentalmente se observaban un aumento de estabilidad y robustez mediante el uso de técnicas de control LQR.

Motivado por el éxito inicial, muchos controladores multivariables se han desarrollado en los últimos 15 años para aplicaciones en Boeing. Por ejemplo en [6] se puede encontrar el desarrollo de una ley de control (con técnica LQR) para la dinámica lateral del avión comercial Boeing 767. Las técnicas LQR también se han usado para el control longitudinal de aviones de transportes con reducida estabilidad longitudinal [7]

Las aplicaciones del control LQR en aeronáutica se extienden también al mundo militar, gran variedad de controladores han sido desarrollados en torno al avión F-16 [8].

También alcanza el control LQR la tarea de mantener satélites en órbitas [9].

Finalmente como caso ilustrativo de aplicación LQR a la aviación se remite al lector al tutorial interactivo de Matlab [10] donde se muestra de manera sencilla y simplificada un control LQR sobre el ángulo de cabeceo de un avión.

3.3 CONTROL LQR CON DINÁMICA LONGITUDINAL

En este punto se partirá de la dinámica longitudinal del avión expresada en la ecuación (2.37) para realizar un control óptimo LQR sobre el ángulo de cabeceo (pitch) del avión. Como se puede observar en la ecuación (2.37) dicho ángulo corresponde a la cuarta componente del vector de estado, y se supondrá en todo caso que dicho estado es accesible y medible (si no, sería necesario el diseño de un observador previa comprobación de la observabilidad del sistema).

Esta sección está dividida en apartados a fin de comprobar los efectos de introducir unas dinámicas de actuadores y piloto sobre los resultados de los análisis. Así, primero se analizará la respuesta del sistema con control LQR considerando unos actuadores ideales (de respuesta instantánea e ilimitada) y sin considerar la acción del piloto. En un segundo apartado se estudiará el efecto de introducir un actuador con saturación y una respuesta con constante de tiempo finita. Finalmente se introducirá la dinámica del piloto manteniendo los actuadores del apartado anterior.

Como punto de partida las especificaciones sobre la respuesta en cabeceo del avión deseada serán:

Referencia cabeceo 0.2 rad (11.46º)
Sobreoscilación < 10%
Tiempo subida < 2 seg. (3.3)
Error en permanente < 2%
Tiempo establecimiento < 10 seg.

Dichos valores se han tomado del Control Tutorial for Matlab (The University of Michigan) [10].

3.3.1 Sistema simple

3.3.1.1 Control LQR en tiempo continuo

Aprovechando los comandos de Matlab existentes y la herramienta Simulink, el control LQR sobre el cabeceo del avión es sencillo de realizar. Eso sí, hay un detalle que conviene resaltar: se está tratando un problema de seguimiento con referencia no nula. Para conseguir que el sistema siga una referencia no nula es necesario escalar convenientemente la señal de referencia como se muestra en el siguiente esquema de Simulink (ver figura 3.1).

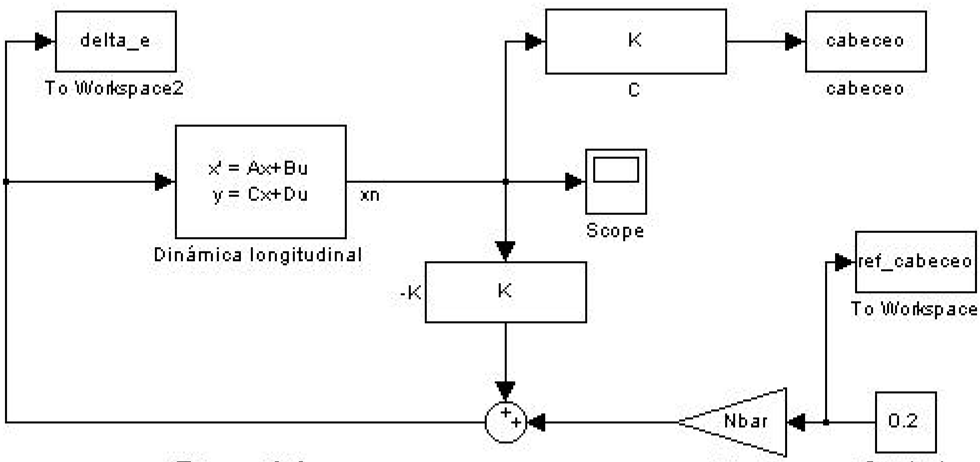


Figura 3.1

Se observa que al no tratarse de un problema de regulación, la solución de control difiere de la hallada para el problema de horizonte infinito en la ecuación (3.1). La señal de control en este caso es una combinación de la referencia (adecuadamente escalada) y de la realimentación con vector de ganancia constante del problema de regulación LQR:

$$u(t) = Nbar * r - Kx(t) \qquad r \equiv \text{referencia}$$

Para conseguir un escalado adecuado de la referencia se usa la función "rscale" creada por Janyie Sun. Dicha función calcula de forma óptima, es decir con mínimo costo computacional, el factor de escala a aplicar, basándose en que, en régimen permanente, la derivada del vector estado se anula. Esto es, si se denota por r a la referencia, el factor de escala Nbar se halla resolviendo:

$$\left.\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) = 0 \\ y(t) &= Cx(t) + Du(t) = Cx(t) = r \\ u(t) &= Nbar * r - Kx(t) \end{aligned}\right\} \quad Ax + BNbarCx - BKx = 0$$

En estos momentos no se detalla su resolución, pues para casos posteriores se tendrá que diseñar detalladamente funciones de escalado ante la inexistencia de éstas. En el apéndice A.1.1 se muestra el código Matlab del archivo "rscale.m".

Para la simulación del control LQR se ha creado el archivo de Matlab LQR_long.m, incluido en el apéndice A.1.2. En la figura 3.2 se presenta la sucesión de tareas que se realizan en dicho archivo.

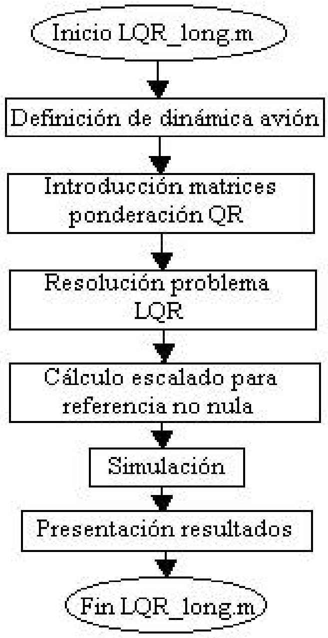


Figura 3.2

La figura 3.3 muestra el resultado de la simulación tomando como matrices de ponderación

$$R = 5 \qquad Q = \begin{bmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,80 \end{bmatrix} \qquad \text{(control cabeceo} \equiv 4^{a} \text{ componente del vector de estados)}$$

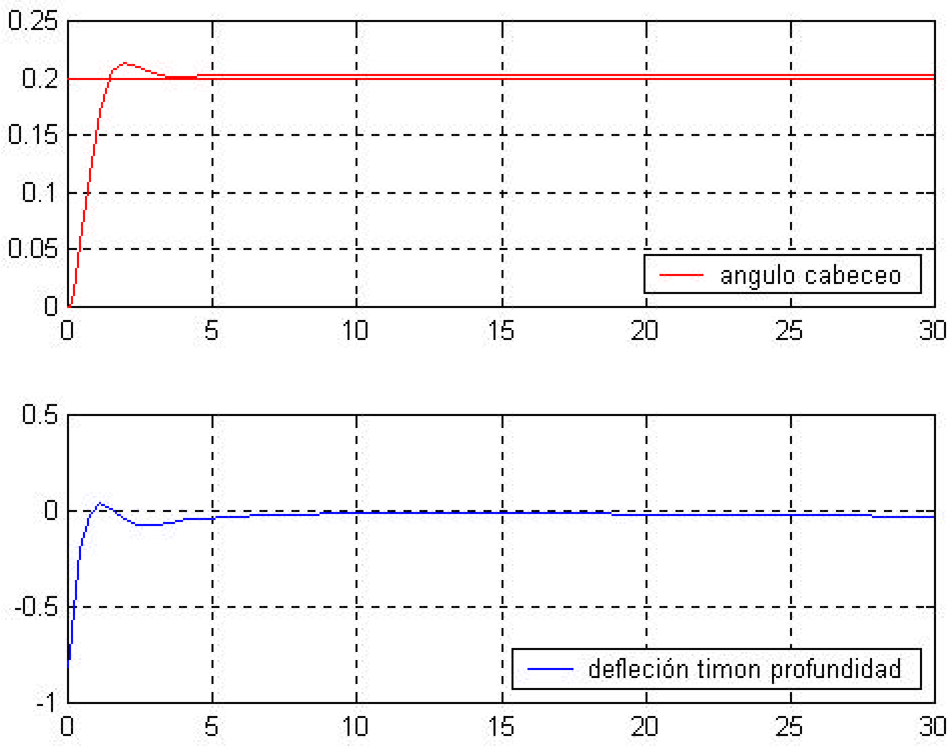


Figura 3.3

Si se realiza la misma simulación para un tiempo mayor de 20 segundos se observa que el ángulo de cabeceo sigue aproximándose más a la referencia. Esto no significa que a los 20 segundos todavía no se ha alcanzado el régimen permanente, sino que éste se alcanza para valores de tiempo muy altos (LQR horizonte infinito). Se considerará como tiempo de establecimiento aquél a partir de cual la salida se estabiliza durante un tiempo prolongado, y que además satisfaga la especificación de error en régimen permanente (si es posible). En el caso tratado resulta un tiempo de subida (ts) de 1.45 segundos, y un tiempo de establecimiento (te) de 4.7 segundos con un error en régimen permanente (e.r.p.) del 1.55% ((0.2031-0.2)*100/0.2). La sobreoscilación (S.O.) en este caso es del 6.8% ((0.2136-0.2)*100/0.2) que también está dentro de las especificaciones mostradas en (3.3).

Es interesante observar también los polos del sistema original y los del sistema con realimentación del vector de estados con ganancia del LQR. Cómo se muestra a continuación, el sistema original tiene dos polos conjugados inestables (aquéllos con parte real positiva), que coinciden con los autovalores de la matriz A. Sin embargo el sistema en bucle cerrado es estable (cómo se observa en la simulación anterior), pues todos sus polos presentan parte real negativa.

-0.37845292570034 + 0.84559712018466i
-0.37845292570034 - 0.84559712018466i   Polos sistema original
0.00055292570034 + 0.05116065903975i
0.00055292570034 - 0.05116065903975i

-0.00899514778879
-0.28813607501543   Polos del lazo cerrado
-1.42301566391117 + 1.62510006305492i   con control LQR
-1.42301566391117 - 1.62510006305492i

La tabla 3.1 muestra los resultados de distintas simulaciones donde se ha ido variando el peso sobre el error en cabeceo en la matriz Q. Además de recoger los datos correspondientes a las especificaciones, se recogen también los valores máximos y mínimos de la señal de actuación (deflexión del timón de profundidad), pues será un dato útil para el siguiente apartado. Para cada parámetro se indica al lado si está dentro de las especificaciones o no. La última columna indica si el control se ajusta en su totalidad a las especificaciones o no.

peso	Ts (seg)		Te (seg)		e.r.p. (%)		S.O. (%)		Actuación	Espec.
1	2.05	No	230	No	1.55	Si	80	No	(-0.18 -0.026)	No
50	1.6	Si	6	Si	2.3	No	7.7	Si	(-0.65 0.013)	No
70	1.45	Si	5	Si	1.6	Si	6.9	Si	(-0.8 0.05)	Si
80	1.45	Si	4.7	Si	1.55	Si	6.8	Si	(-0.814 0.045)	Si
100	1.37	Si	4.7	Si	1	Si	6.5	Si	(-0.9 0.06)	Si
500	0.95	Si	3	Si	0.1	Si	5	Si	(-2 0.26)	Si
7000	0.53	Si	2	Si	0	Si	4	Si	(-7.5 3)	Si

Tabla 3.1

Se puede observar que, a medida que aumenta el peso en la matriz Q, mejora la respuesta en el cumplimiento de especificaciones. A su vez también se exige más a los actuadores, esto tendrá su repercusión en el apartado posterior. Al aumentar dicho peso mucho (por encima de 400) aparece una oscilación secundaria de pequeña amplitud. Por ejemplo para un peso de 500 se tiene la respuesta parcial de la figura 3.4.

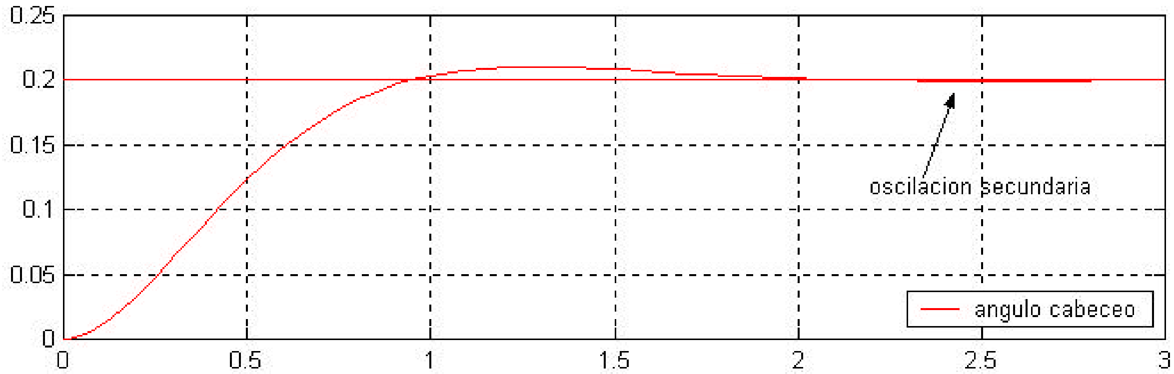


Figura 3.4

Para la representación de resultados se usa el archivo "Graficas.m", cuyo código aparece en el apéndice A.1.3.

3.3.1.2 Control LQR en tiempo discreto

En el apartado anterior se desarrolló un controlador en tiempo continuo. Sin embargo uno de los avances más importantes en aviación fue la introducción de la electrónica digital (microprocesadores) en los Sistemas de Control Automático de Vuelo (SCAV). La justificación del paso al tiempo discreto se encuentra con las múltiples ventajas de los sistemas de control digitales:

- Reducción de tamaño y precio (esto último no es cierto para sistemas sencillos con un reducido número de variables).
- No existe límite en la complejidad del algoritmo. Cosa que sí sucedía anteriormente con los sistemas analógicos.
- Facilidad de ajuste y cambio. Por el mismo motivo anterior un cambio en un control analógico implica, en el mejor de los casos, un cambio de componentes si no un cambio del controlador completo.
- Tendencia al control distribuido o jerárquico. Se ha pasado de la idea de usar un único controlador o computador para toda una planta a la de distribuir los dispositivos. Con esto se alcanza una mayor flexibilidad y fiabilidad, que se puede ver aumentada con la introducción de redundancia de equipos.
- Exactitud y estabilidad en el cálculo debido a que no existen derivas u otras fuentes de error.

Lógicamente los sistemas de control digitales también tienen sus inconvenientes. En la mayoría de los casos el proceso a controlar es continuo (caso tratado aquí), es decir se debe excitar con una señal continua y genera una salida continua. Esta señal, como en cualquier lazo de control es medida por algún dispositivo que a su vez entrega una señal continua proporcional a la magnitud medida. Por otra parte está el computador que sólo trabaja con valores discretos. Para compatibilizar ambos existen los convertidores digital-analógico (CDA) y analógico-digital (CAD) que realizan la conversión de magnitudes. No se profundizará más en el estudio de estos elementos, pues existe una amplia gama de documentos y textos sobre ellos.

Otro aspecto importante del trabajo en tiempo discreto con procesos continuos es la elección de un tiempo de muestreo adecuado y de una técnica de muestreo. En este caso se usará un mantenedor de orden cero, o lo que es lo mismo muestreo y retención “sampling and hold”. La elección del tiempo de muestreo se realiza atendiendo a distintas técnicas empíricas existentes que proporcionan un tiempo de muestreo suficientemente pequeño. Pero como regla fundamental, se aceptará que el tiempo de muestreo es adecuado cuando los resultados en tiempo discreto se ajusten al caso continuo (esta comparación no es siempre posible).

El archivo LQR_long_d de Matlab lleva a cabo un control LQR en tiempo discreto del cabeceo del avión equivalente al control LQR en tiempo continuo del apartado anterior, este archivo se incluye en el apéndice A.1.4. En la figura 3.5 se presenta la sucesión de tareas que se realizan en el archivo LQR_long_d.

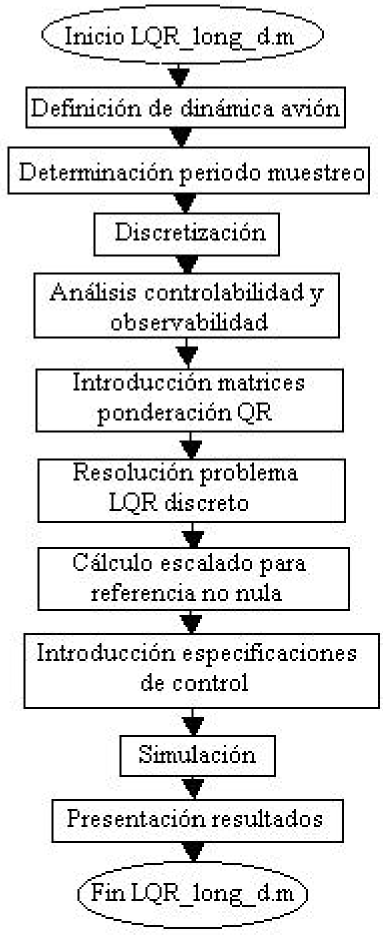


Figura 3.5

Para no confundir con los desarrollos en tiempo continuo se ha adoptado una nueva notación, de forma que las matrices que definen el sistema en tiempo discreto A, B, C y D pasan a denotarse F, G, H, J respectivamente.

En el apéndice A.1.4 se construyen las matrices de observabilidad y controlabilidad para comprobar que el sistema es completamente controlable (es posible llevarlo de cualquier estado inicial **x**(t0) a cualquier otro estado en un número finito de intervalos) y completamente observable (con el sistema en el estado x(t), es posible determinar dicho estado a partir de las mediciones de la salida con un retraso finito de tiempo). En el presente caso el sistema resulta observable y controlable, por lo que se puede pasar al diseño de un controlador discreto.

Para la elección del tiempo de muestreo se ha elegido la mitad del aconsejado por una regla empírica, que propone coger un tiempo de muestreo de 1/30 BW. Donde BW es la frecuencia en la que la respuesta en frecuencia del lazo cerrado (con realimentación unitaria) cae por debajo de –3dB. El cálculo de dicha frecuencia para el sistema concreto dinámica longitudinal del avión, lo realiza el archivo de Matlab mostrado en el apéndice A.1.5. Para el caso tratado aquí resulta un tiempo de muestreo de 0.01754 segundos.

En cuanto al escalado de la señal de referencia ya no se han encontrado funciones con tal cometido. Partiendo de que en régimen permanente x(k+1)=x(k) y que la salida y(k) es igual a la referencia r, se puede plantear un sistema de ecuaciones suponiendo una señal de control u(k)=N*r-Kx(k). El sistema se plantea y resuelve en el archivo mirscale_discreto.m de Matlab, resultando independiente de la señal de referencia r tomada. Las variables del sistema de ecuaciones planteado son:

x, corresponde a x(1)/r (primera componente vector estados/referencia)
y, corresponde a x(2)/r
z, corresponde a x(3)/r
N, es el factor de escala buscado.
x(4) no es una variable, pues se supone x(4)=r, ya que y=x(4).

La función mirscale_discreto devuelve el factor de escala a aplicar para el control LQR en tiempo discreto (con vector ganancia K) del ángulo de cabeceo ante referencia no nula. A continuación se muestra el contenido del archivo mirscale_discreto, donde se ha considerado una dinámica en el espacio discreto del avión dada por: x(k+1)=Fx(k)+Gu(k); y(k)=[0 0 0 1]x(k).

```
function salida=mirscale_discreto(F,G,K)

S=solve('F(1,1)*x+F(1,2)*y+F(1,3)*z+F(1,4)+G(1)*N=x+G(1)*(K(1)*x+K(2)*
                                                   y+K(3)*z+K(4))',...
         'F(2,1)*x+F(2,2)*y+F(2,3)*z+F(2,4)+G(2)*N=y+G(2)*(K(1)*x+K(2)*
                                                   y+K(3)*z+K(4))',...
         'F(3,1)*x+F(3,2)*y+F(3,3)*z+F(3,4)+G(3)*N=z+G(3)*(K(1)*x+K(2)*
                                                   y+K(3)*z+K(4))',...
         'F(4,1)*x+F(4,2)*y+F(4,3)*z+F(4,4)+G(4)*N=1+G(4)*(K(1)*x+K(2)*
                                          y+K(3)*z+K(4))','N,x,y,z');
salida=eval(S.N);
```

El esquema del sistema en Simulink es similar al caso continuo, salvo que se sustituye el bloque Dinámica longitudinal por un bloque en el espacio de estados discreto. Dicho bloque se inicializa como muestra la figura 3.6.

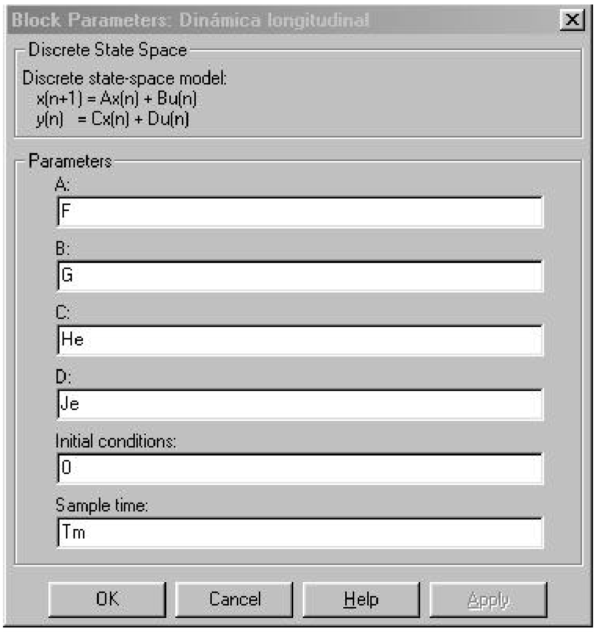


Figura 3.6

El archivo que genera las gráficas y los resultados de la simulación es idéntico al caso continuo.

3.3.2 Sistema con actuador real

En este apartado se incluirá la dinámica del actuador sobre el timón de profundidad. Esta dinámica consistirá en un sencillo sistema de primer orden (3.4), obtenido de un informe de la Universidad de Budapest [11], que habrá que discretizar. Además para completar la dinámica del actuador se tienen que añadir bloques de saturación, que hacen el sistema no lineal, dicha saturación limitará la deflexión del timón de profundidad entre –20 y +40 grados (-0.349 0.698 rads). Para la discretización del sistema de primer orden bastan estas líneas en Matlab:

```
sisc=tf([37],[1 37]);                    %f.t actuadores en t.continuo
sisd=c2d(sisc,Tm);                       %f.t actuadores en t.discreto
[num_act,den_act]=tfdata(sisd,'v');
```

$$\frac{37}{s+37} \qquad \text{(dinámica actuador)} \qquad (3.4)$$

El modelo en Simulink queda modificado con respecto al caso discreto del apartado anterior como se muestra en la figura 3.7.

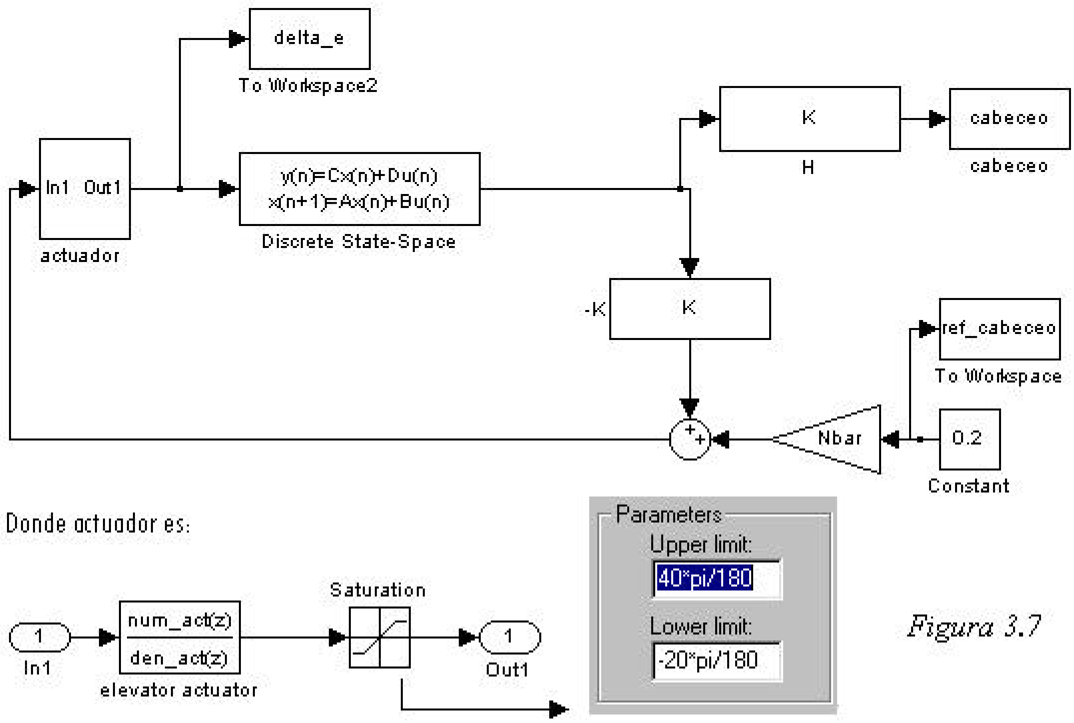


Figura 3.7

Lógicamente los resultados de la simulación del control LQR cambian respecto al apartado anterior. Por una parte, la deflexión del timón de profundidad no puede sobrepasar sus límites. Por otra parte el cambio del timón no es instantáneo, por lo que en el instante inicial éste tendrá un deflexión nula y evolucionará a otros valores con la dinámica de primer orden anteriormente expuesta (bastante rápida, ô = 1/37 s).

La figura 3.8 muestra la simulación para el caso de R=5 y $Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80 \end{bmatrix}$:

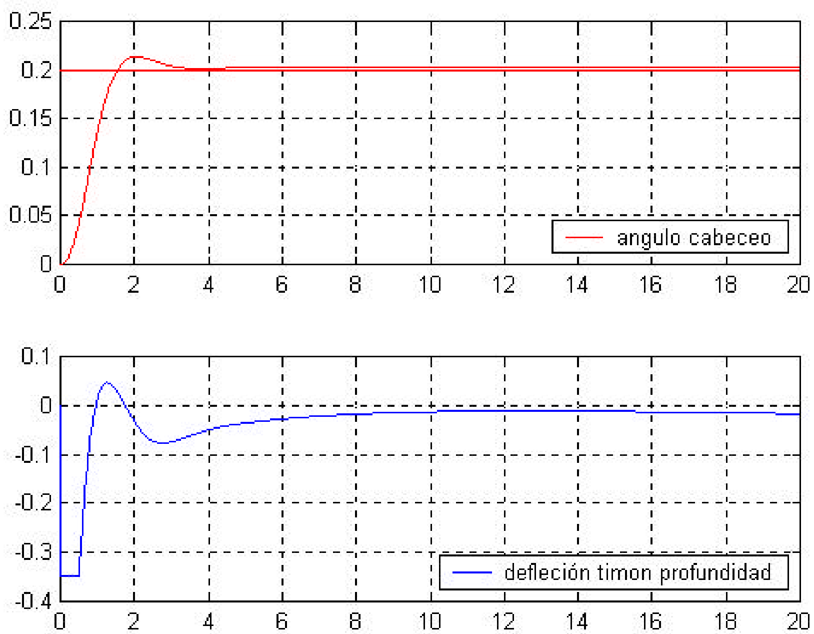


Figura 3.8

El comportamiento, en cuanto al error en régimen permanente y tiempo de establecimiento, prácticamente no cambia con respecto al caso con actuadores ideales. Sin embargo el tiempo de subida ha aumentado de 1.45 a 1.524 s debido al freno inicial impuesto a la caída en el ángulo de deflexión del timón. Por su parte la sobreoscilación disminuye de 6.8% a 6.7%. Esta disminución de la sobreoscilación se ve más aguzada para un peso en Q de 500, pues en este caso desaparece la oscilación secundaria perceptible en el apartado anterior, y la oscilación principal disminuye del 5% al 4%. Eso sí, para un peso de 500, el tiempo de subida ha aumentado de 0.95 a 1.36 s. con respecto al caso sin actuadores.

Para terminar este apartado es conveniente comentar que la mejora en características de oscilación del caso con actuador real respecto al caso con actuador ideal no es infinita. Si se aumenta el peso en la matriz Q lo suficiente para conseguir la doble saturación del actuador (superior e inferior), aparecerán más de una oscilación secundaria en la respuesta, cosa que no sucedía en el caso con actuador real. Además la amplitud de las oscilaciones aumenta de forma considerable. Así por ejemplo para un peso de 100.000 (poco práctico dado que con valores menores la respuesta se ajusta perfectamente a las especificaciones) se obtiene la respuesta de la figura 3.9.

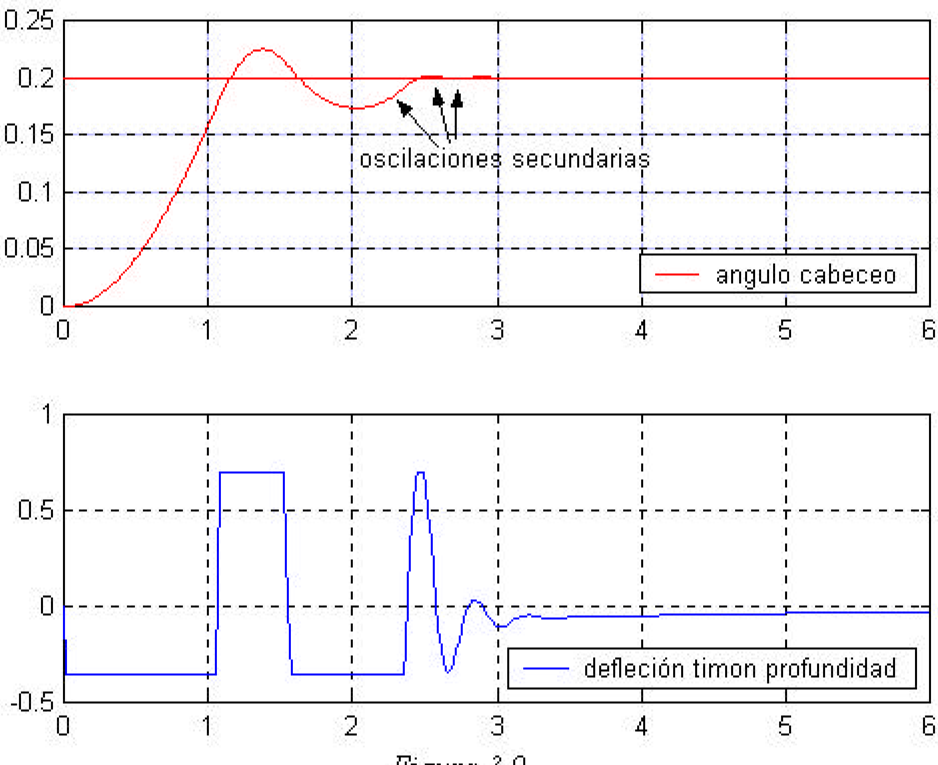


Figura 3.9

3.3.3 Sistema completo con actuador real y piloto

Ya se ha introducido la dinámica del actuador en el apartado anterior. Ahora se introducirá la dinámica del piloto. El comportamiento del piloto ante cambios longitudinales se puede expresar como muestra la ecuación (3.5), obtenida del Instituto Politécnico de Virginia [12]. Para conseguir la función de transferencia discreta equivalente se tiene que usar la transformación bilineal (Tustin) por tratarse de un sistema no causal:

$$s = \frac{2}{Tm}\frac{z-1}{z+1}$$

$$0.74(0.27S+1)e^{-0.3S} \qquad (3.5)$$

```
%Definición función transferencia piloto
Tp=0.3;             %retardo introducido por piloto en segundos
Kp=0.74;
s=tf('s');
aux=Kp*(0.27*s+1);
z=tf('z',Tm);
rp=Tp/Tm-mod(Tp/Tm,1);                   %retardo  en  tiempos  de  muestreo
piloto=c2d(aux,Tm,'tustin')*z^(-rp);     %f.t discreta: acción piloto
[num_pil,den_pil]=tfdata(piloto,'v');    %numdor/dendor para Simulink
```

Si se introduce esta dinámica en el modelo Simulink del apartado anterior se obtiene la respuesta de la figura 3.10 para un peso 80 de la matriz Q del LQR:

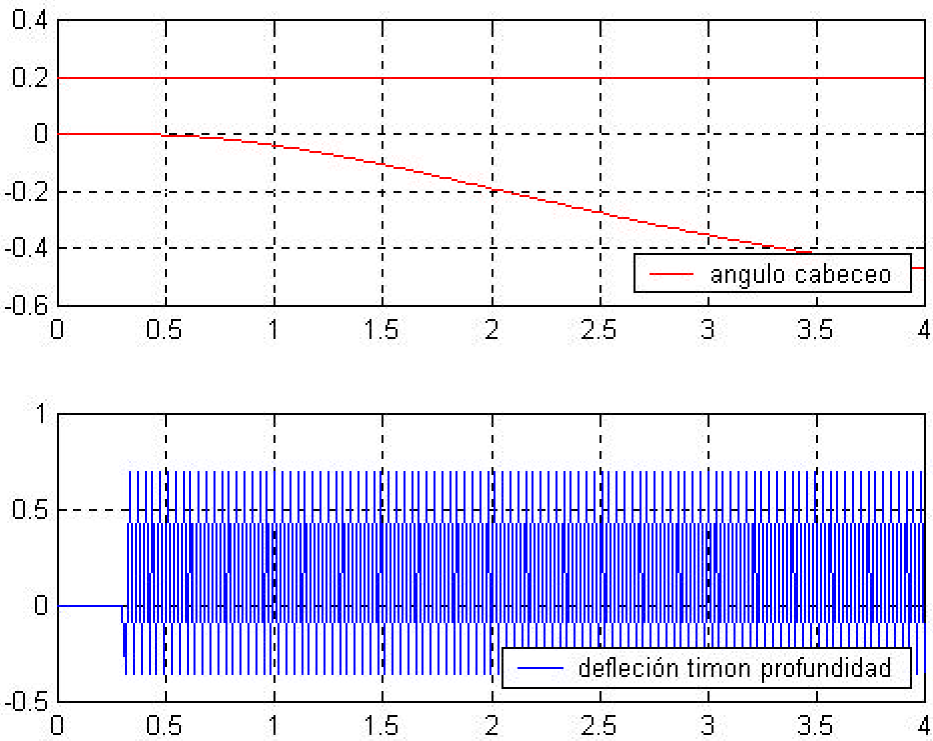


Figura 3.10

Se puede observar que el sistema se ha inestabilizado. Para conseguir estabilizar el sistema se propone cancelar la acción del cero introducido por el piloto, para ello se introduce un controlador con un polo en -1/0.27 (cero de transmisión en piloto). Además para compensar el efecto de la ganancia introducida por el piloto, una vez calculada la ganancia de realimenación K y el escalado de referencia para dicho valor K se deben dividir dichos valores por la ganancia del piloto Kp. Con todo esto el modelo Simulink queda como muestra la figura 3.11:

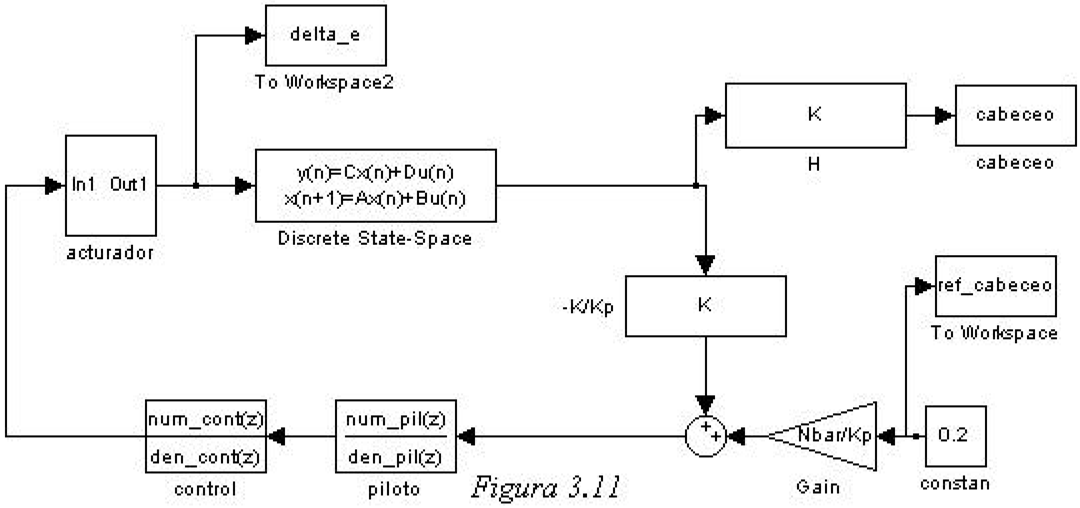


Figura 3.11

```
control=c2d(1/(0.27*s+1),Tm,'tustin');
[num_cont,den_cont]=tfdata(control,'v');
```

Con este modelo y un peso de 80 para Q se obtiene una respuesta que no se ajusta a las especificaciones debido a la aparición de múltiples oscilaciones. Estas oscilaciones resultan de la interacción entre el piloto y la dinámica del avión, debido a las limitaciones de rango y velocidad de los actuadores. Estas oscilaciones son conocidas como PIO (pilot-induced oscillation) y su reducción pasa por intercalar un filtro de compensación de fases entre las dinámicas del piloto y el actuador [13]. La respuesta obtenida es la de la figura 3.12.

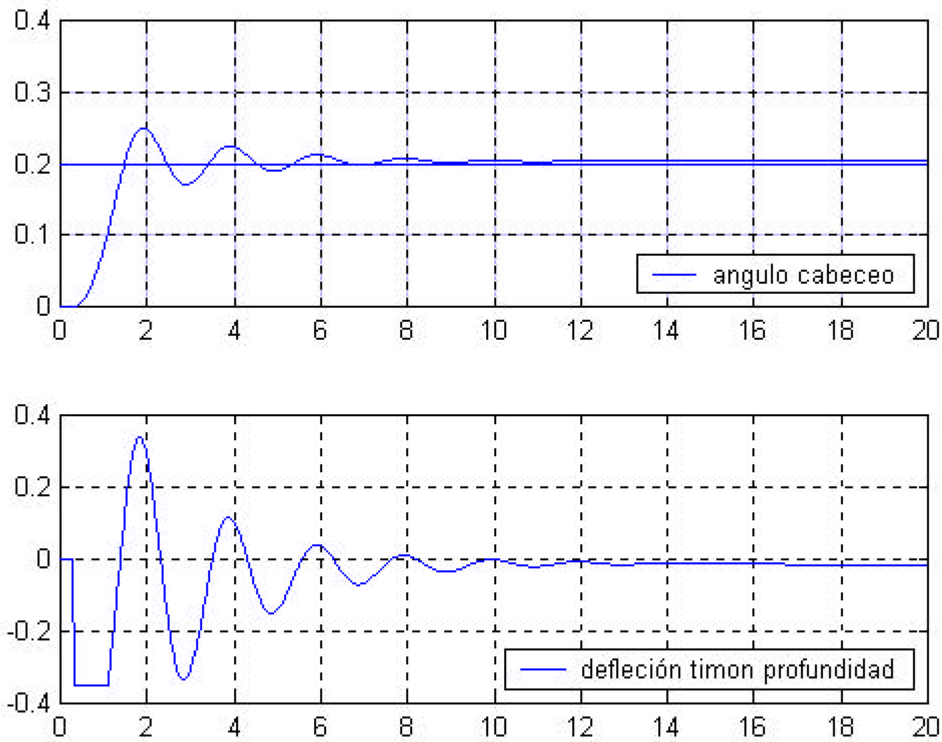


Figura 3.12

Esta respuesta no se ajusta a las especificaciones. Si se prueba con otros pesos para Q no se mejora en ningún aspecto.

Llegado este punto se propone una última opción para disminuir las oscilaciones introducidas por el piloto y cumplir todas las especificaciones. Esta opción consiste en aprovechar el cero introducido por el piloto (-1/0.27) para formar una red de avance con un polo introducido en el controlador. Para que este polo cumpla su función, debe ser más rápido que el cero que forma con él la red de avance. En la figura 3.13 se observa como mejora la respuesta si se sustituye el bloque control anterior por `c2d(1/(0.1*s+1),Tm,'tustin')`:

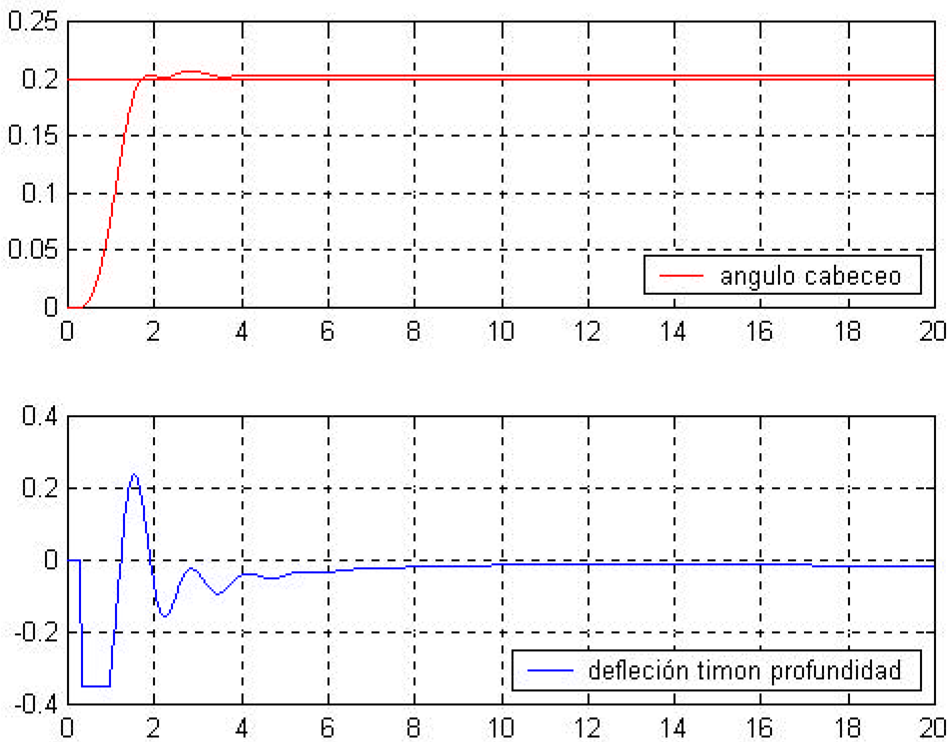


Figura 3.13

3.4 CONTROL LQR CON DINÁMICA LATERAL

En este punto se partirá de la dinámica lateral del avión expresada en la ecuación (2.38) para realizar un control óptimo LQR sobre el ángulo de derrape (sideslip, â) y el ángulo de alabeo (roll, ö) del avión. Como se puede observar en la ecuación (2.38) el ángulo de derrape corresponde a la primera componente del vector de estados, y el ángulo de alabeo corresponde a la cuarta. Al igual que con el control LQR longitudinal, se supondrá en todo caso que dicho estado es accesible y medible. En otro caso sería necesario el diseño de un observador previa comprobación de la observabilidad del sistema.

$$\begin{bmatrix} b \\ r \\ p \\ j \end{bmatrix}' = \begin{bmatrix} -0.056 & 0 & -1 & 0.03924 \\ -1.05 & -0.47 & 0.39 & 0 \\ 0.6 & -0.032 & -0.115 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} b \\ r \\ p \\ j \end{bmatrix} + \begin{bmatrix} 0 & 0.012 \\ 0.14 & 0.15 \\ 0.008 & -0.48 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} d_a \\ d_r \end{bmatrix} \qquad (2.38)$$

En este apartado se ha hecho la misma subdivisión de apartados que en el caso del control LQR longitudinal con el mismo fin. Se intentará seguir una referencia de 0.05 rad (3º) para el ángulo de derrape, y de 0.2 rad (13º) para el ángulo de alabeo. Al tratarse de un control múltiple con dinámicas cruzadas será complicado ajustar las respuestas a unas especificaciones dadas, así que se intentará obtener siempre una respuesta satisfactoria manteniendo eso sí un error del 2% a los 10 segundos.

3.4.1 Sistema simple

Se realizará un control LQR discreto donde la matriz Q será fijada con pesos no nulos para las componentes de ángulo de derrape (primera) y ángulo de alabeo (cuarta). La matriz R variará según el peso que se le quiera dar a la actuación de los alerones y del timón de dirección:

$$Q = \begin{bmatrix} 80 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 80 \end{bmatrix} \qquad R = \begin{bmatrix} peso\_a & 0 \\ 0 & peso\_r \end{bmatrix}$$

En este caso se trata una referencia doble (derrape y alabeo), por lo que el escalado ante referencia no nula se complica. Existen cuatro tipos de escalados adecuados a cuatros casos distintos: las referencias en alabeo y derrape son ambas nulas, la referencia en alabeo es nula, la referencia en derrape es nula, las referencias en alabeo y derrape son ambas no nulas. En todos estos casos el escalado lo realizará una matriz de escalado 2x2 que actúa directamente sobre el vector referencia, pero la distribución de los elementos nulos dentro de dicha matriz de escalado es distinta según el caso. Para conseguir una matriz de escalado adecuada se han planteado numerosos sistemas de ecuaciones hasta dar con uno compatible determinado. En el apartado anterior el escalado era independiente del valor de referencia, posteriormente se verá que en este caso el escalado depende del valor relativo entre referencias (cociente).

El esquema Simulink para el control LQR es el mostrado en la figura 3.14.

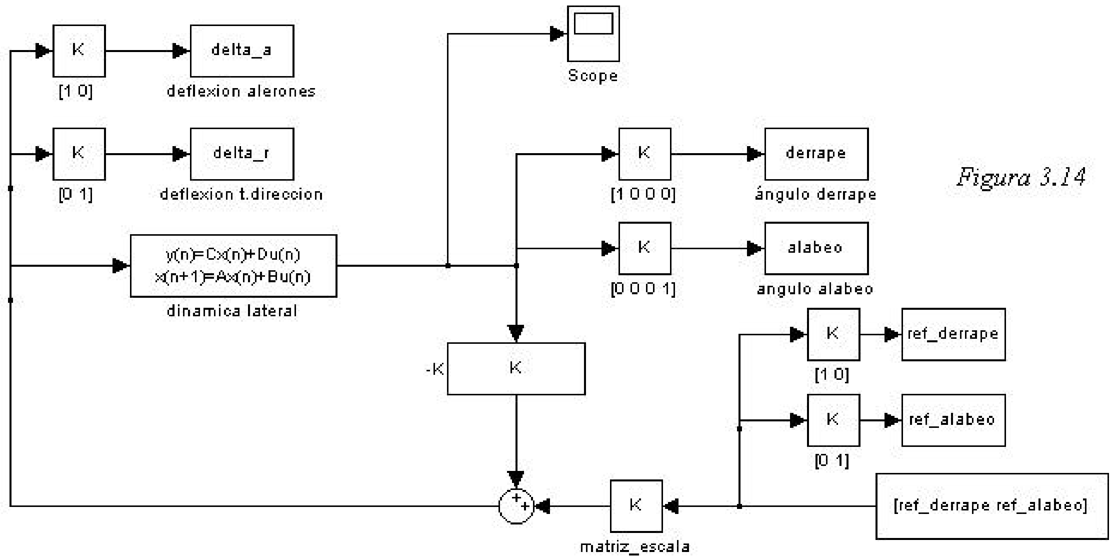


Figura 3.14

El archivo en Matlab para el control LQR aparece en el apéndice A.1.6. Siendo la sucesión de tareas realizadas en el las mostradas en la figura 3.2 (ver punto 3.3.1.2).

representación de resultados.

Para la determinación del tiempo de muestreo se ha usado la misma regla empírica que

de banda por observación de los diagramas de Bode de las cuatro funciones de transferencias definidas a partir de la representación en espacio de estados del sistema tratado (salidas alabeo y derrape, frente a entradas alerones y timón dirección). Para la representación de dichos diagramas de Bode se ha usado la herramienta ltiview en el código que se comenta a continuación.

```
sist_ss=ss(A,B,C,D);
sist_tf=tf(sist_ss);

sist_cloop=sist_tf;   %para tener objeto lti model.

sist_cloop(1,1)=feedback(sist_tf(1,1),1);
sist_cloop(1,2)=feedback(sist_tf(1,2),1);
sist_cloop(2,1)=feedback(sist_tf(2,1),1);
sist_cloop(2,2)=feedback(sist_tf(2,2),1);

ltiview('bode',sist_cloop)
```

Los polos de las cuatro funciones de transferencia definidas en sist_tf coinciden, pues el denominador de estas funciones de transferencia es común. Al cerrar los lazos independientemente, estos polos se desplazan un poco dependiendo de la señal de entrada, auque se mantienen cercanos. Las gráficas obtenidas al ejecutar este archivo indican que la elección de BW = 2 rad/seg es más que suficiente (figura 3.15).

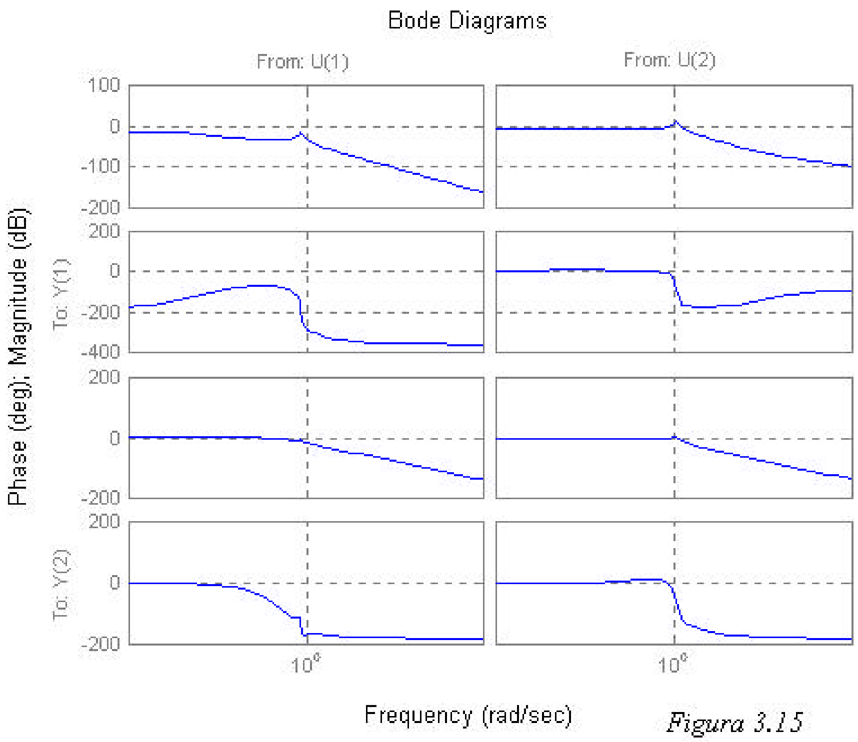


Figura 3.15

cálculo de la matriz de escalado para referencia no nula. Este escalado se determina igual a x(k). El archivo mirscale_lateral.m contiene el código Matlab para el cálculo de la riz de escalado, este archivo se incluye en el apéndice A.1.8. Al tratarse de un sistema MIMO (Multi Input Multi Output) en el planteamiento del sistema de ecuaciones se deben e derrape y con ref2 a la referencia para el ángulo de alabeo, la sucesión de tareas realizadas en el archivo mirscale_lateral.m se pueden resumir en la figura 3.16.

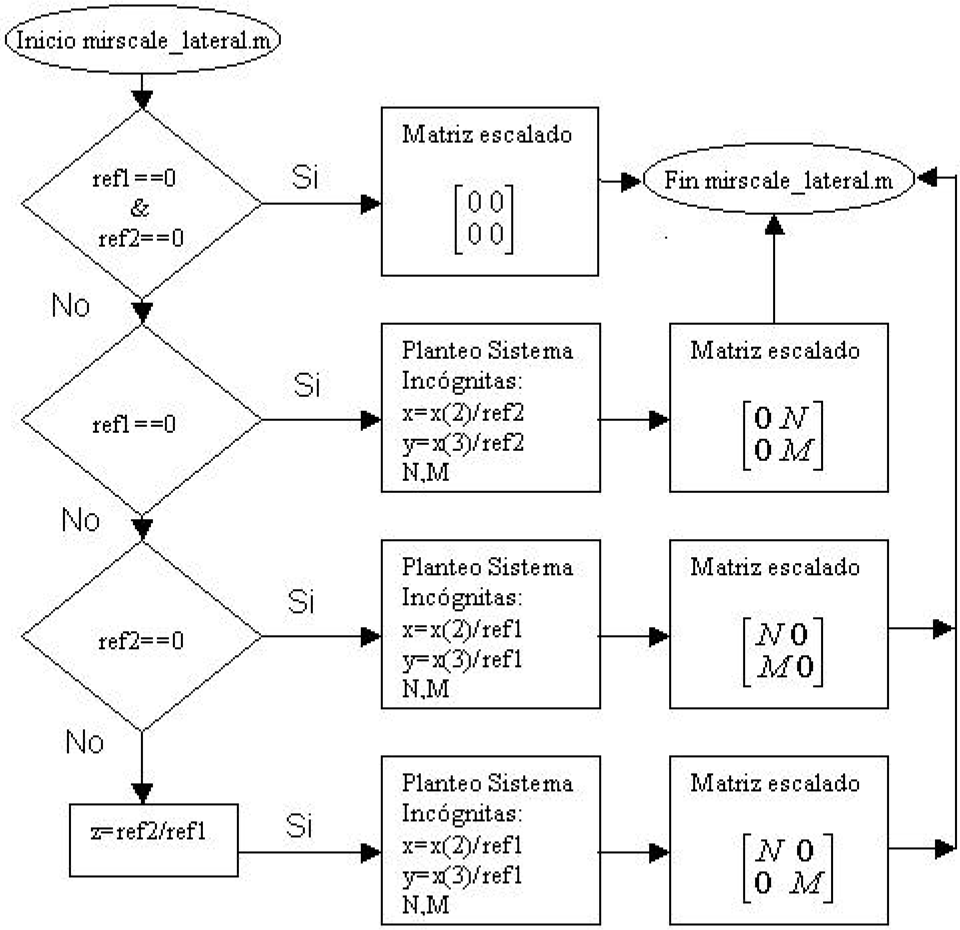


Figura 3.16

cociente entre referencias. Del mismo modo parece que en los casos de una sola referencia no

referencia no nula, se ve que el escalado permanece invariable. Esto es así, porque los

la referencia no nula (escalar por 1/referencia las ecuaciones).

El resultado de la simulación para una matriz $R = \begin{bmatrix} 5 & \\ 0 & 5 \end{bmatrix}$ en la figura 3.17.

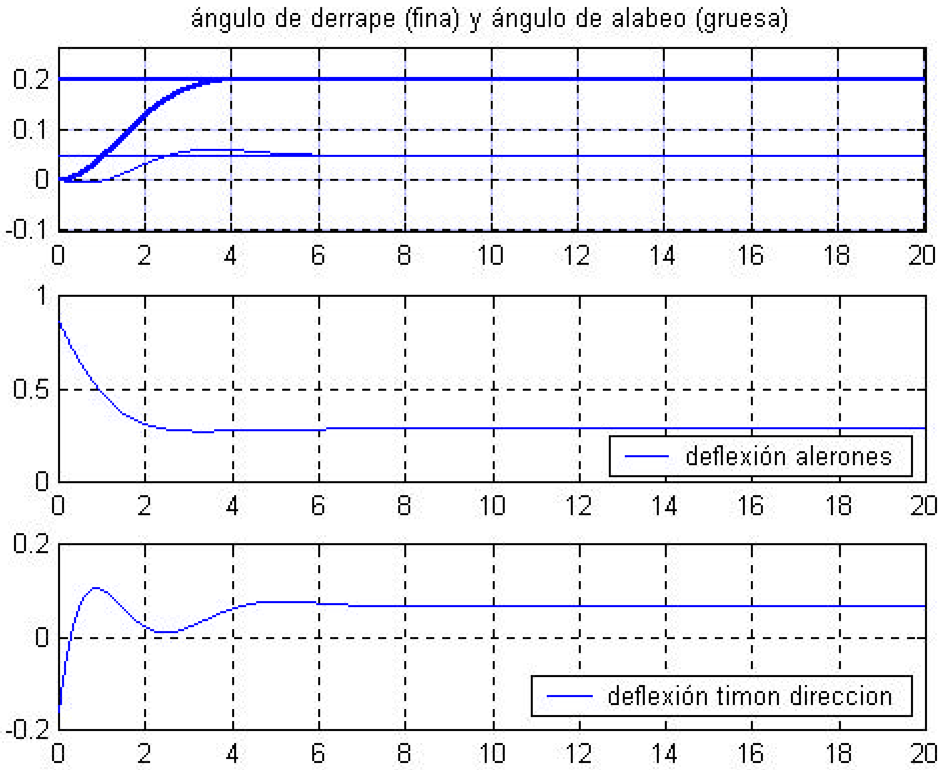


Figura 3.17

En esta respuesta se puede observar que inicialmente el ángulo de derrape disminuye en vez de intentar alcanzar directamente su valor de referencia (0.05 rad). Esto se debe al control simultaneo de dos variables y a la dinámica cruzada existente entre actuaciones y ángulos a controlar. A pesar de todo esta respuesta es aceptable, se observa que no existe sobreoscilación en el ángulo de alabeo, y que los errores en permanente son muy pequeños. El tiempo de subida (ts) en la respuesta en alabeo es de 3.5 seg.

Se puede comprobar que el control LQR estabiliza el sistema observando que los polos del lazo cerrado discreto tienen todos módulos menor que la unidad:

Polo		Módulo
0.9931 + 0.0105i	→	0.9932
0.9931 - 0.0105i	→	0.9932
0.9932 + 0.0039i	→	0.9932
0.9932 - 0.0039i	→	0.9932

Para analizar el efecto de la ponderación de pesos en la matriz R, es interesante calcular la actuación total (integral continua o suma discreta) de las superficies de control. Esto se realiza en Matlab con la función norm llamada con los parámetros delta_a.signals.values y delta_r.signals.values.

En la tabla 3.2 se presentan los resultados de distintas simulaciones, donde para distintos pesos sobre las actuaciones, se calcula y aplica el control LQR correspondiente midiendo la carga de los actuadores, el ts de la respuesta en alabeo, y el pico de

sobreoscilación (S.O.) para las respuestas en alabeo y derrape. En dicha tabla se distinguen también dos tipos de comportamiento
inversión inicial en la respuesta en derrape y no presenta S.O. en alabeo; el tipo de respuesta B presenta S.O. en alabeo y no presenta inversión inicial (ver figura 3.18).

peso_a	peso_r	S.O. Alabeo (					
	5	0	23.6	3.5	15.7006	3.2128	A
50	50	0	22	5	14.4355	2.99	

Tabla 3.2

En esta tabla la variación de la matriz R se ha hecho manteniendo la relación delta_a/delta_r = 1. Se observa que a medida que aumentan los pesos de la matriz R del LQR se obtiene una respuesta que exige menos a las actuaciones disminuyendo la S.O, pero a cambio la se hace más lenta (aumentar R simétricamente equivale a disminuir Q). Finalmente decir que en todos estos casos el tiempo de subida de la respuesta en derrape se mantiene inferior al tiempo de subida de la respuesta en alabeo, y que el error se mantiene inferior al 2%.

Considero interesante analizar la respuesta del sistema ante una matriz R con relación de pesos no unitarios. La figura 3.18 muestra la respuesta del sistema con peso_a=0.05 y peso_r=5:

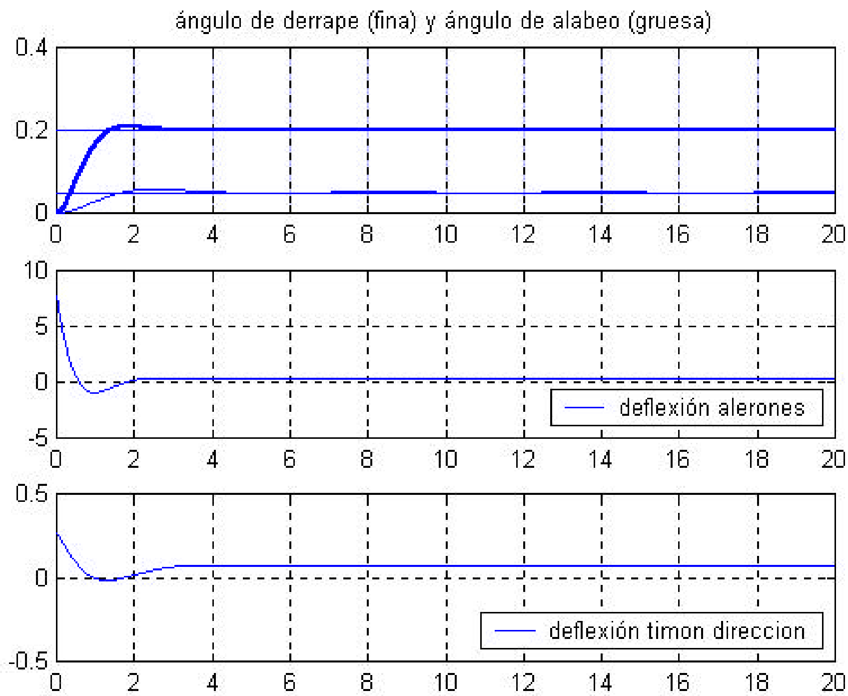


Figura 3.18

Al permitir una mayor carga de los alerones (disminuir peso_a) se prioriza la respuesta en alabeo, llegando a ser el ts en alabeo menor que el ts en derrape. En un posterior estudio de acoplamiento (ver apartado 6.3.2) se observará que la actuación de los alerones interactúa fuertemente con el ángulo de alabeo, mientras que el timón de dirección se relaciona sobre todo con la respuesta en derrape.

Si se intercambian ahora los pesos, es decir peso_a=5 y peso_r=0.05 se obtiene una respuesta (figura 3.19) donde se acentúa el valle inicial en la respuesta en derrape:

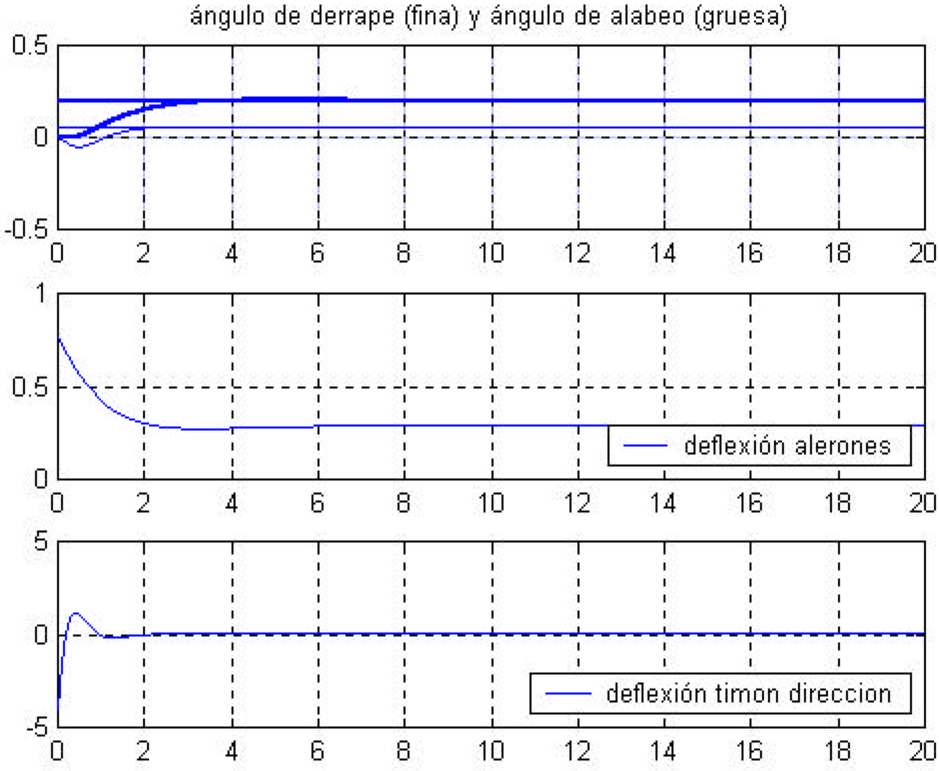


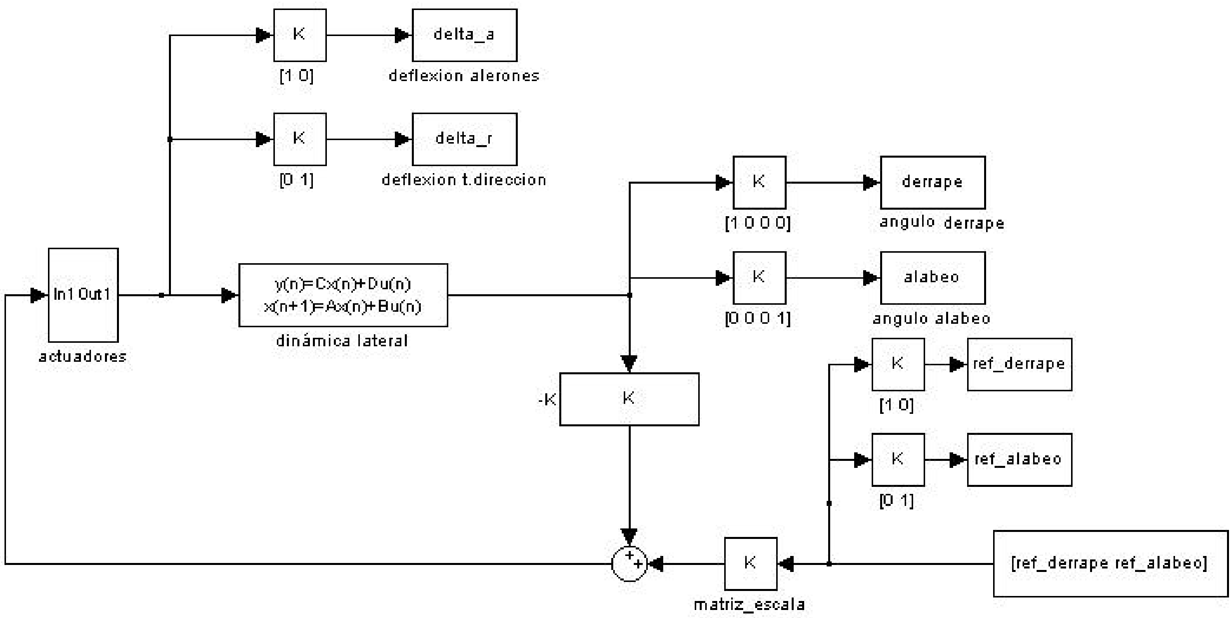
Figura 3.19

3.4.2 Sistema con actuador real

En este apartado se incluirá la dinámica de los actuadores sobre los alerones y el timón de dirección. Esta dinámica consistirá en un sistema de segundo orden $1/(0.0169S^2+0.121S+1)$ (Instituto Politécnico Nacional de México, Ing. Aeronáutica [14]) que habrá que discretizar. Además para completar la dinámica de los actuadores se tienen que añadir bloques de saturación (sistema no lineal), dicha saturación limitará la deflexión de los alerones entre –20 y +20 grados ([-0.35, 0.35] rads) y la del timón de profundidad entre –30 y +30 grados ([-0.52, 0.52] rads).

Para la discretización del sistema de primer orden bastan unas lineas en Matlab:

```
sisc=tf([1],[0.0169 0.121 1]);          %f.t actuadores en t.continuo
sisd=c2d(sisc,Tm);                      %f.t actuadores en t.discreto
[num_act,den_act]=tfdata(sisd,'v');
```



Donde actuadores es:

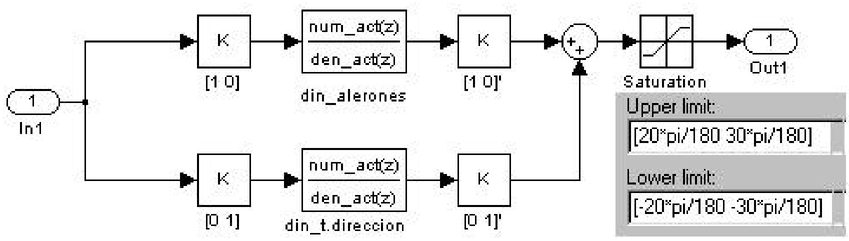


Figura 3.20

El efecto de la inclusión de la dinámica de los actuadores es similar al observado en el control LQR de la dinámica longitudinal. Al no poder conmutar instantáneamente los actuadores de su valor inicial nulo al valor inicial impuesto por el LQR, y al estar limitada la deflexión de las superficies de control, se ralentiza la respuesta del sistema (aumentando los tiempos de subida). A su vez, también disminuye la sobreoscilación.

La figura 3.21 muestra la respuesta del sistema para $R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$

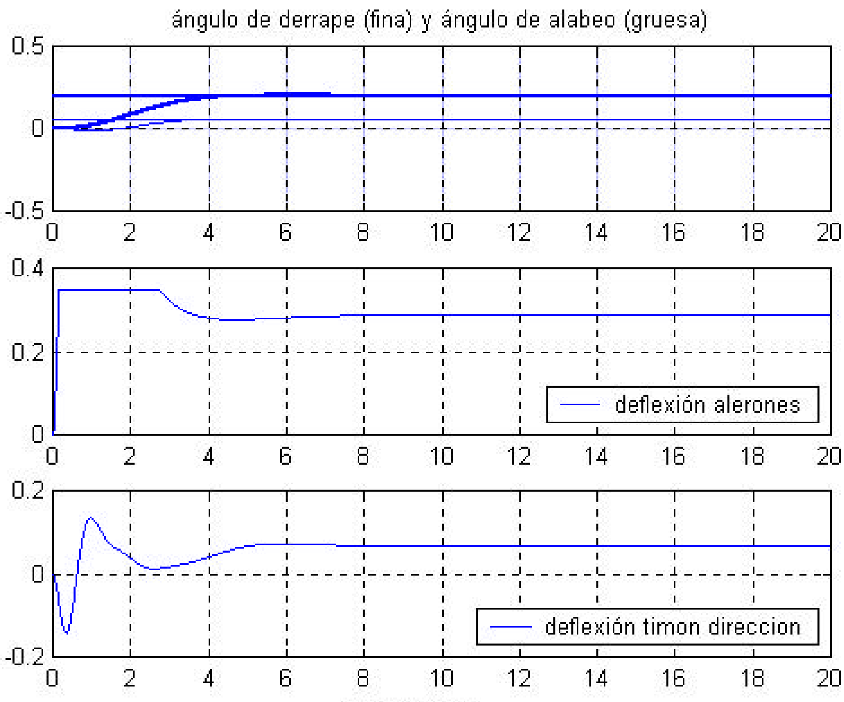


Figura 3.21

Aunque esta respuesta puede ser aceptable, no lo son así muchas de las que parecían correctas en el apartado anterior. La mayoría de controles propuestos anteriormente exigen demasiado a los actuadores, sobre todo en lo que se refiere a los alerones, llegándose a una situación de saturación de la que el sistema no es capaz de recuperarse. La figura 3.22 muestra por ejemplo la respuesta del sistema para el caso con

$$R = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$$

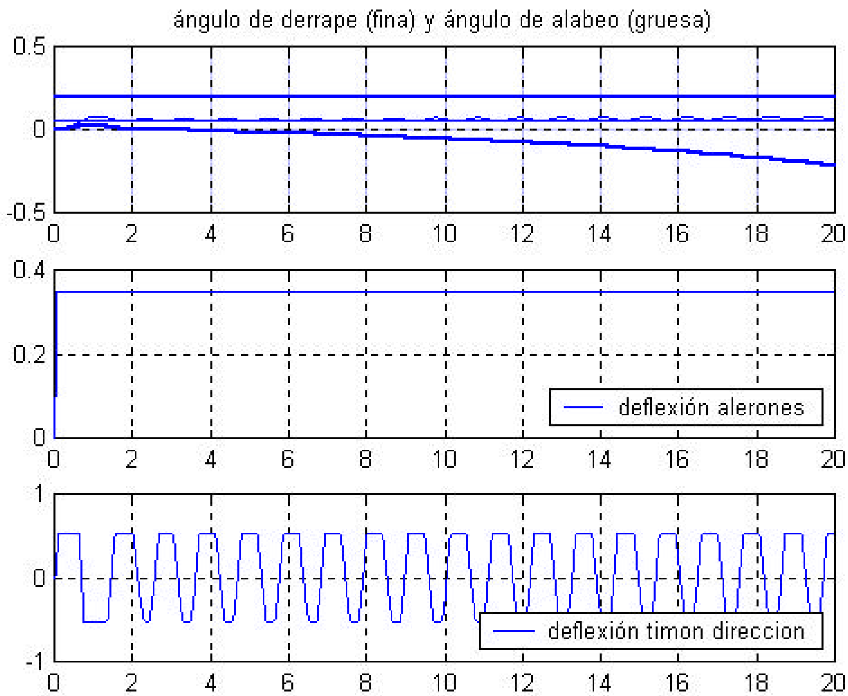


Figura 3.22

Para intentar evitar la saturación de alerones se puede escoger una opción de control que minimice en mayor parte la actuación de los alerones. Con este objeto se realizan 2 simulaciones: una con peso_a=5 y peso_r=0.5 (figura 3.23), satura el timón de profundidad pero el sistema se recupera: otra con peso_a=5 y peso_r=0.05 (figura 3.24), satura el timón de profundidad y el sistema se inestabiliza.

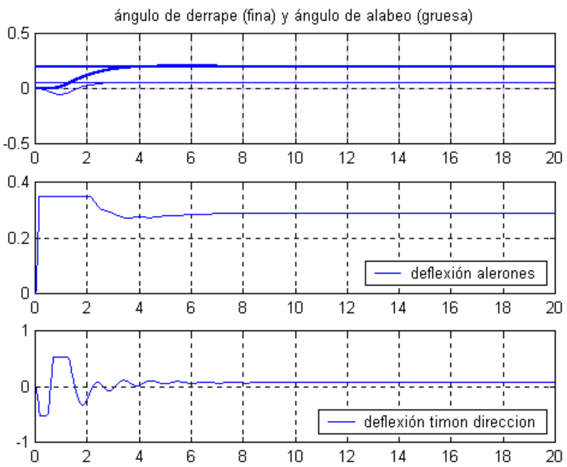


Figura 3.23

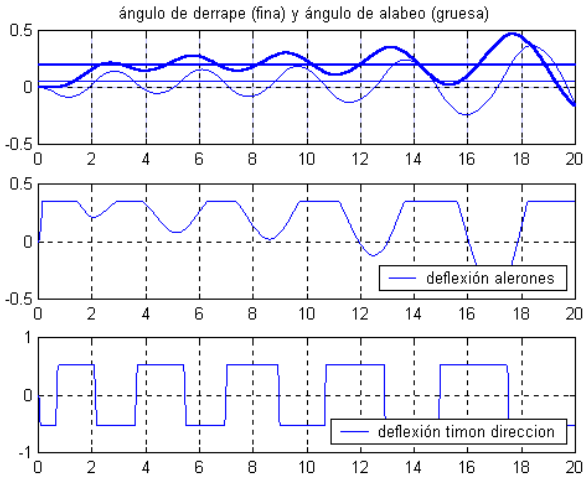


Figura 3.24

Como se ha podido comprobar hasta ahora en todas las respuestas (con o sin actuadores) la carga a los alerones es mucho mayor que la carga sobre el timón de dirección, siendo imposible evitar la saturación de los alerones. En un futuro estudio de acoplamiento (ver punto 6.3.2) se verá que esto parece lógico para el control del ángulo de alabeo, pero no así para el control del ángulo de derrape cuyo acoplo principal es con el timón de dirección. En el punto 6.3 se intentará un control PID sobre la dinámica lateral desacoplada.

3.4.3 Sistema completo con actuador real y piloto

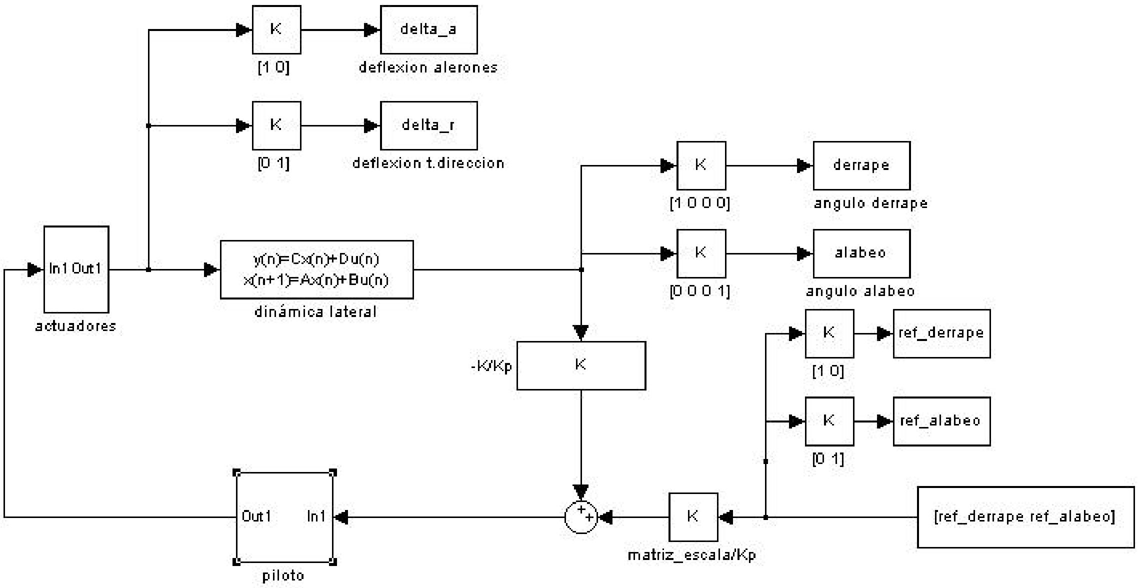
Ya se ha introducido la dinámica del actuador en el apartado anterior. Ahora se introducirá la dinámica del piloto. El comportamiento del piloto ante cambios transversales se puede expresar como (3.6). Esta dinámica se ha obtenido del Instituto Politécnico de Virginia [15].

$$1.5/(0.13S+1)e^{-0.15S} \qquad (3.6)$$

Para conseguir la función de transferencia discreta equivalente a (3.6) se introducen las siguientes líneas en Matlab:

```
%Definición función transferencia piloto
Tp=0.15;           %retardo introducido por piloto en segundos
Kp=1.5;
s=tf('s');
aux=Kp/(0.13*s+1);
z=tf('z',Tm);
rp=Tp/Tm-mod(Tp/Tm,1);                   %retardo  en  tiempos  de  muestreo
piloto=c2d(aux,Tm)*z^(-rp);              %f.t      discreta      de      piloto
[num_pil,den_pil]=tfdata(piloto,'v');    %para usar en Simulink
```

Para compensar el
ganancia de realimentación K y la matriz de escalado sobre la referencia para dicho valor K, se deben dividir dichos valores por la ganancia del piloto Kp en el modelo Simulink (figura .25).



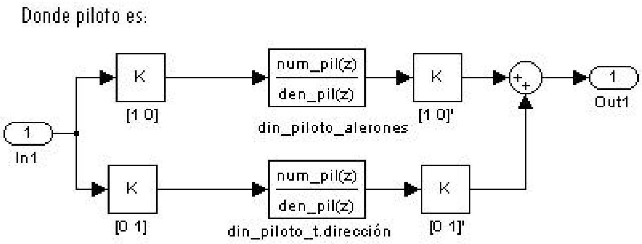


Figura 3.25

piloto aparecen unas oscilaciones PIO (Pilot-

$$= \begin{bmatrix} 5 & 0 \\ & 5 \end{bmatrix}:$$

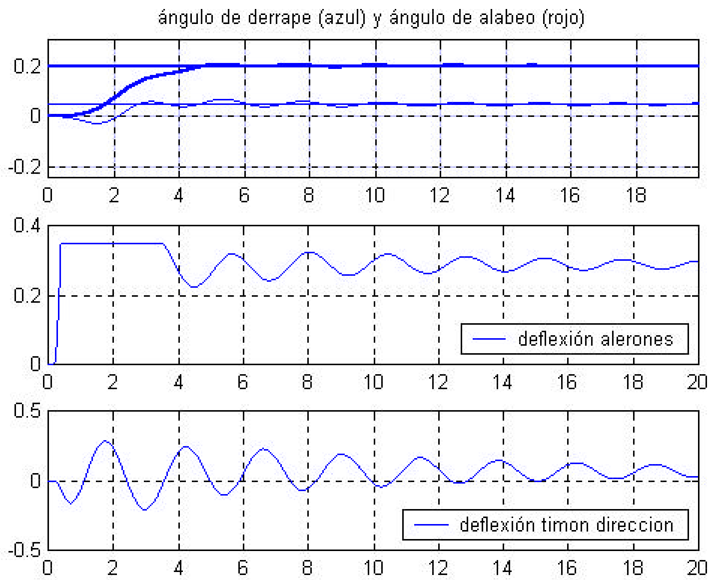


Figura 3.26

3.5 COMPORTAMIENTO ANTE PERTURBACIONES

En este apartado se analizará el comportamiento del sistema completo controlado ante perturbaciones en las actuaciones y en las variables reguladas. Se verá como el control LQR es capaz de recuperarse ante perturbaciones instantáneas, así como no puede hacerlo ante perturbaciones mantenidas.

Para realizar la simulación del comportamiento del sistema ante perturbaciones se añaden en Simulink dichas perturbaciones usando bloques de entrada de tipo escalón, sumadores y matrices de escalado (para las perturbaciones a la salida). La figura 3.27 muestra como añadir las perturbaciones a la entrada y salida para el caso longitudinal (esquemas superiores) y el caso lateral (esquemas inferiores).

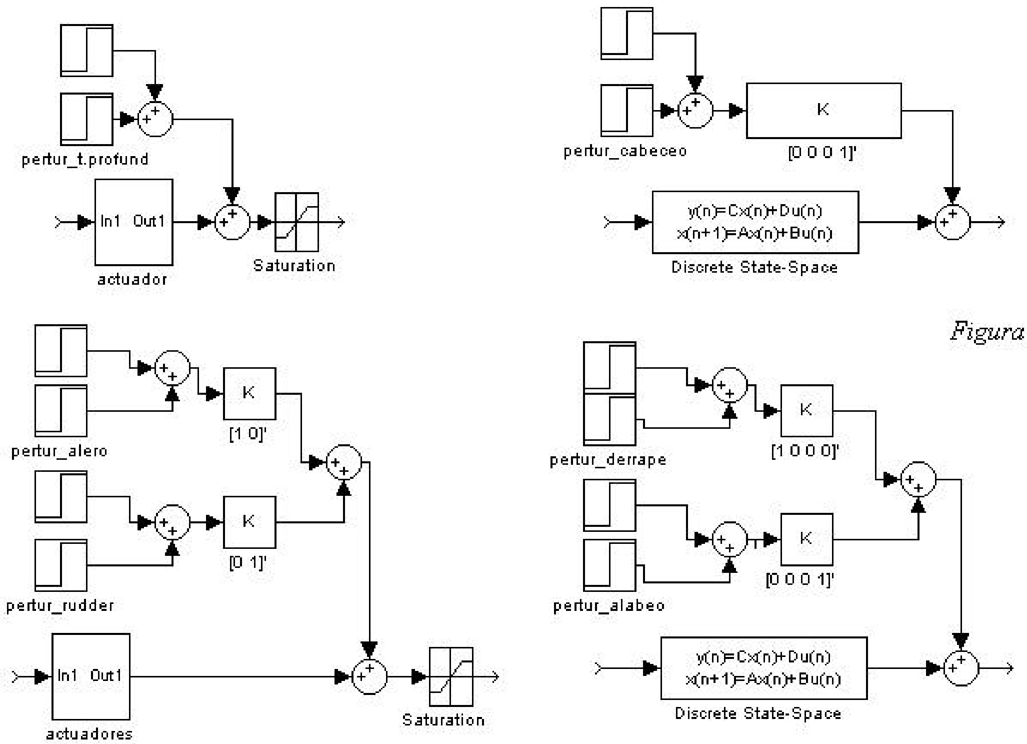


Figura 3.27

Los parámetros de los bloques de entrada de tipo escalón se ajustan según el instante

(hasta final de tiempo de simulación).

3.5.1 Perturbaciones a la salida

3.5.1.1 LQR con dinámica longitudinal

A continuación se introduce una

salida (ángulo de cabeceo) del 50% del valor de referencia (0.2*0.5=0.1rad) a los 12 segundos. En la figura 3.28 se observa un pequeño error entre el ángulo de cabeceo y la

rror se debe a que aún no se ha alcanzado el régimen permanente (ver sección 3.6), pudiéndose asegurar que el controlador rechaza la perturbación instantánea correctamente, de hecho el controlador es capaz de rechazar cualquier perturbación instantánea a la salida sea cual sea su amplitud.

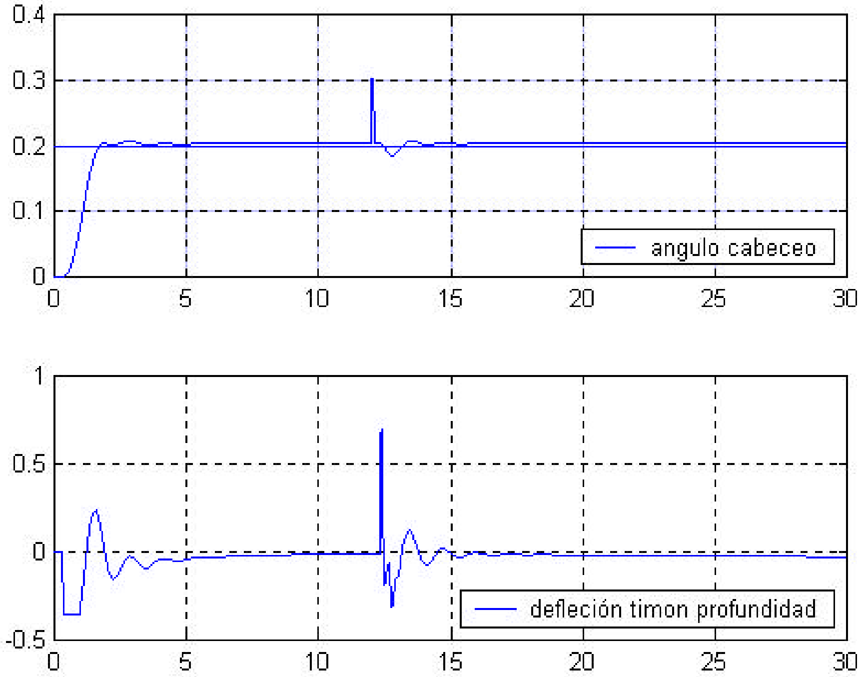


Figura 3.28

Si la perturbación introducida es mantenida (hasta fin de simulación) el controlador LQR no es capaz de rechazarla. No obstante en el caso del sistema estudiado y ante una perturbación del 50% del valor de referencia el ángulo de cabeceo en régimen permanente se aproxima mucho al valor de referencia (error < 2%) como se puede ver en la figura 3.29.

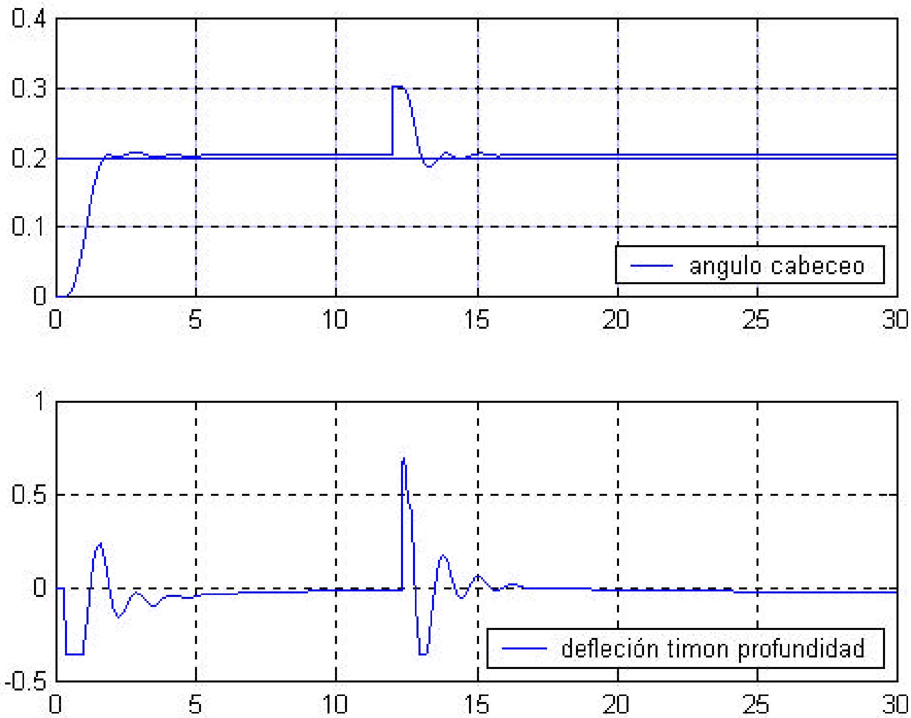


Figura 3.29

Sin embargo si la perturbación mantenida es de mayor amplitud el sistema no se recupera como sí ocurriera para perturbaciones instantáneas.

3.5.1.2 LQR con dinámica lateral

Al igual que en el caso longitudinal, el controlador LQR lateral es capaz de rechazar cualquier perturbación instantánea en las variables controladas (ángulo de alabeo y derrape), provocando cambios en las dos señales de actuación (alerones y timón dirección). A modo de ejemplo se muestra (figura 3.30) la respuesta del sistema en el caso de introducir una perturbación instantánea en t=15 en el ángulo de alabeo del 50% del valor de referencia (0.5*0.2=0.1), y otra perturbación instantánea en t=25 en el ángulo de derrape del 100% del valor de referencia (0.05):

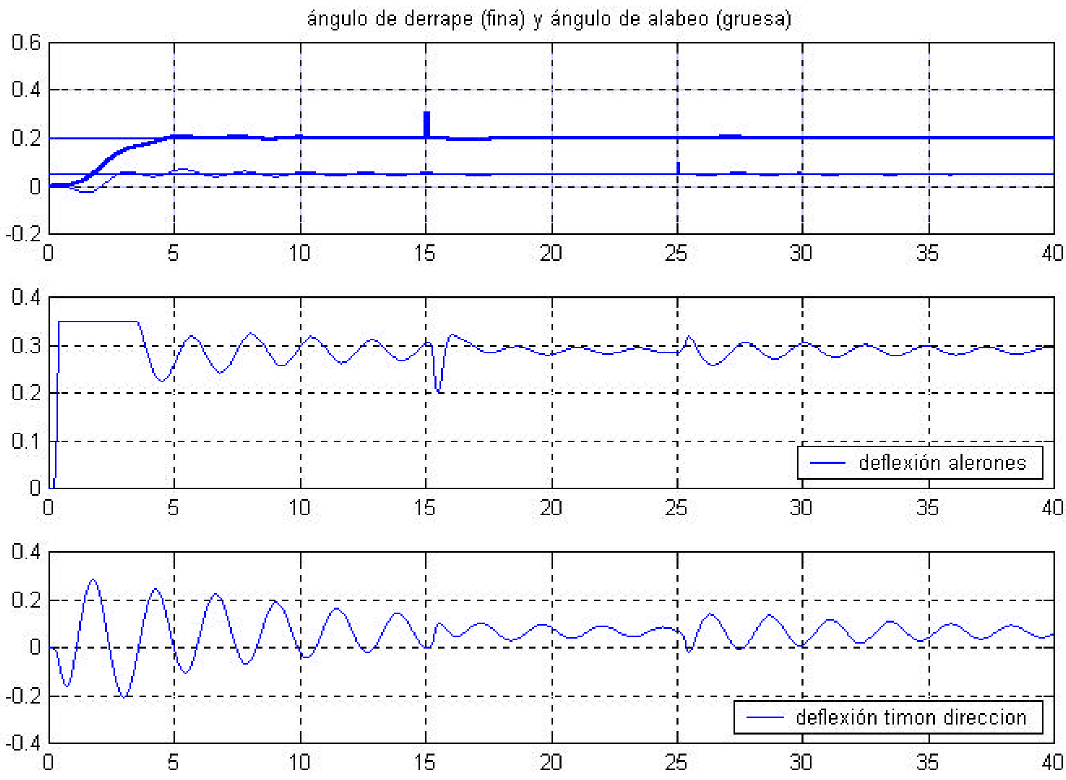


Figura 3.30

Si se aplican perturbaciones mantenidas el sistema no es capaz de recuperarse, viéndose afectado en mayor medida por las perturbaciones en el ángulo de derrape que provocan un cambio en el nivel medio de la deflexión de los alerones. Como ejemplo se puede ver en la figura 3.31, el comportamiento del sistema cuando se introducen perturbaciones del 100% en el ángulo de alabeo (t = 15 en adelante) y en el ángulo de derrape (t =30 en adelante). Las perturbaciones se han escogido de gran magnitud en este caso para que sean más fáciles de observar sus efectos gráficamente, siendo similar cualitativamente el comportamiento para perturbaciones de menor valor.

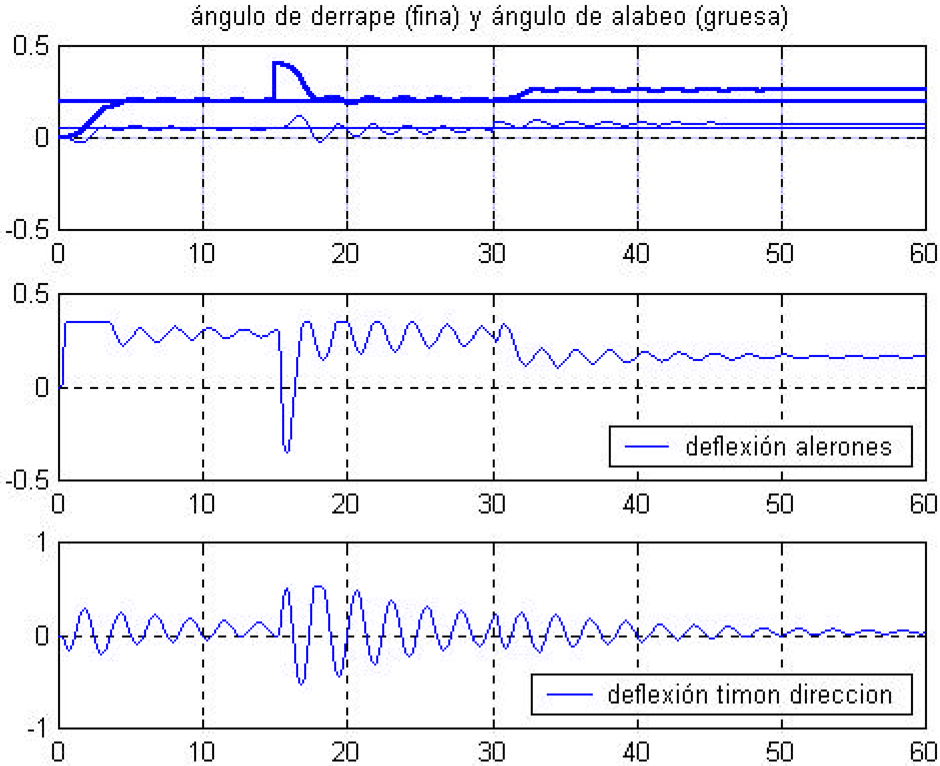


Figura 3.31

3.5.2 Perturbaciones a la entrada

3.5.2.1 LQR con dinámica longitudinal

Introduciendo una perturbación de 0.1 rad instantánea (de t =12 a t =12.1 seg) en la deflexión del timón de profundidad se observa en la figura 3.32 que el sistema se recupera rápidamente. Esta recuperación ocurre incluso si la perturbación es tan grande que satura el actuador. Al igual que para el caso con perturbaciones a la entrada, la figura 3.32 muestra un pequeño error en el seguimiento de la referencia. Este error es debido a que no ha pasado el suficiente tiempo entre la aparición de la perturbación y el fin de simulación como para alcanzar el régimen permanente (ver sección 3.6).

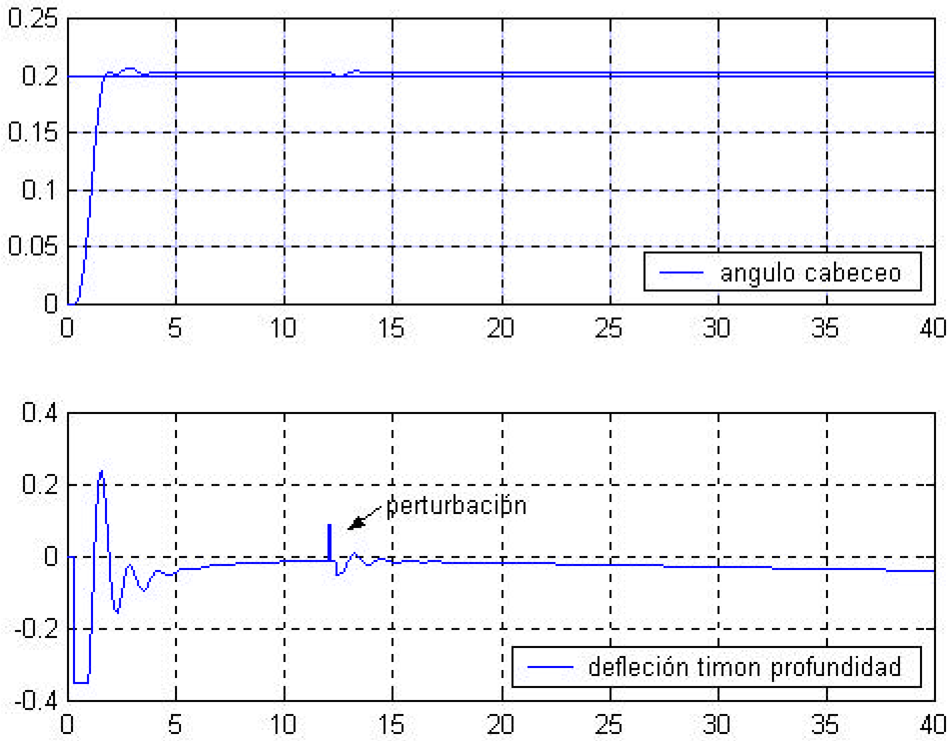


Figura 3.32

Sin embargo si la misma perturbación es mantenida la respuesta del sistema se aleja mucho de la deseada como muestra la figura 3.33.

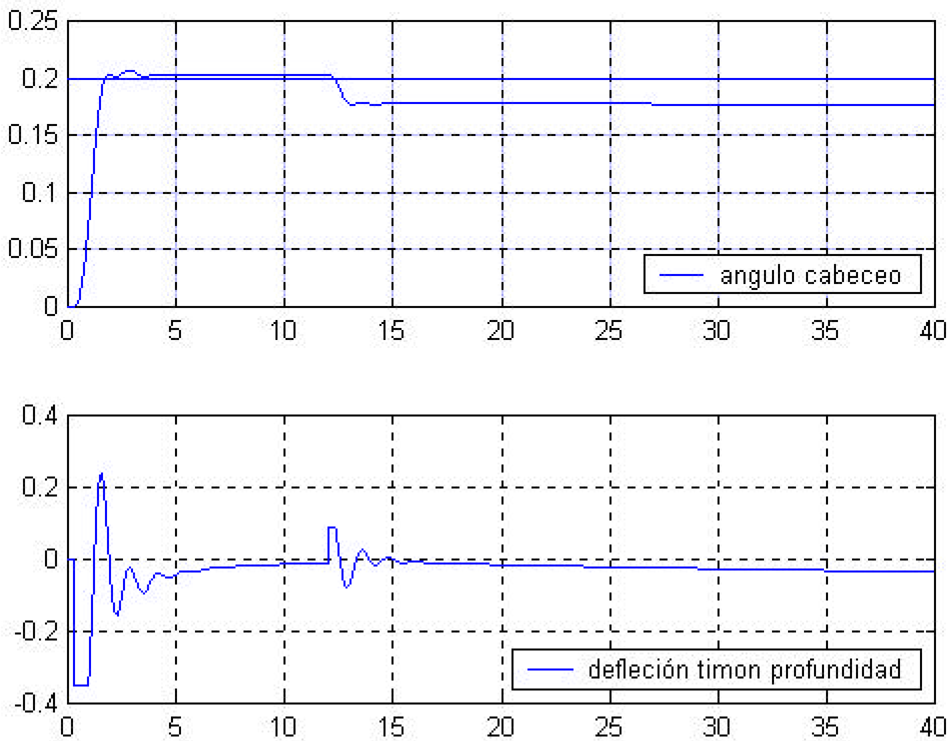


Figura 3.33

3.5.2.2 LQR con dinámica lateral

Al igual que con las perturbaciones a la salida, ante perturbaciones a la entrada el sistema se recuperará si estas son instantáneas, y no lo hará si son mantenidas. La figura 3.34 muestra la respuesta del sistema cuando se introducen perturbaciones instantáneas de 0.1rad en el timón de dirección (t =22 a t =22.1) y –0.1rad en los alerones (t =35 a t =35.1).

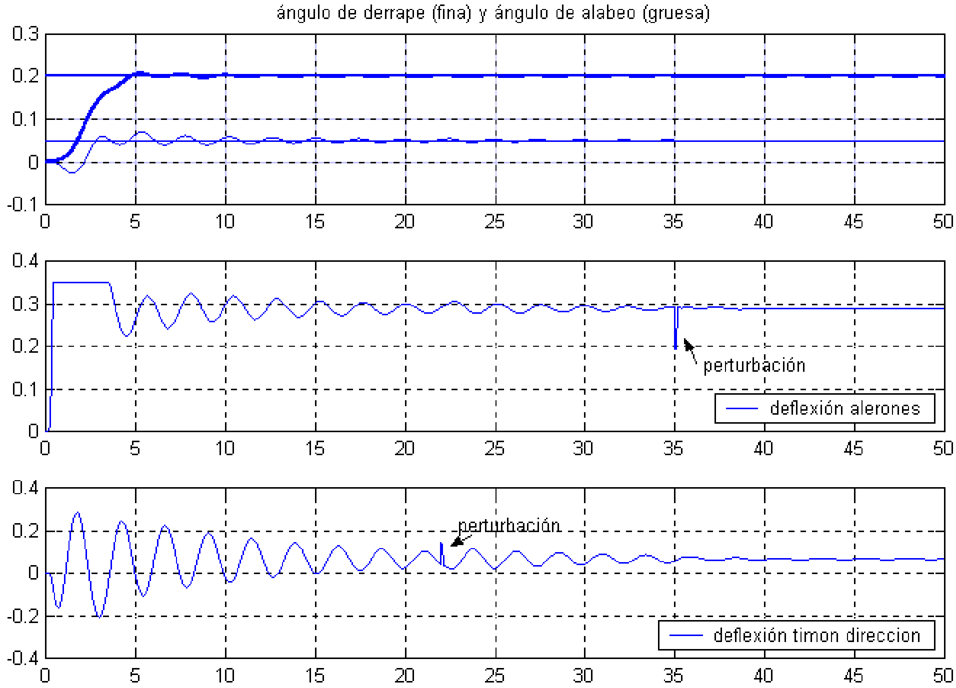


Figura 3.34

con las mismas perturbaciones mantenidas el sistema se aleja de la referencia (figura 3.35):

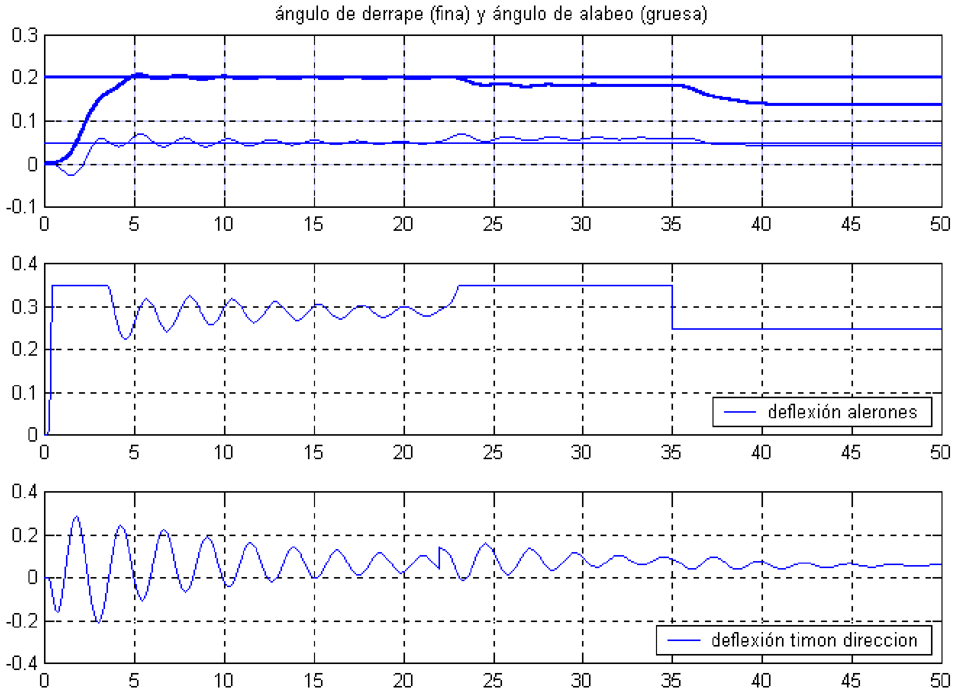


Figura 3.35

3.6 ANÁLISIS DEL RÉGIMEN PERMANENTE

Hasta ahora las simulaciones se han realizado hasta unas cuantas decenas de segundos. Sin embargo es interesante observar el valor al que tiende la señal de control en régimen permanente para los distintos casos. El análisis realizado en esta sección se puede extender a todos los controladores tratados en otros capítulos, pues el sistema a controlar es el mismo en todos los casos, por lo que también lo será su ganancia estática.

3.6.1 LQR con dinámica longitudinal

Si se realiza una simulación del control LQR sobre la dinámica longitudinal completa del avión para una referencia de 0.2 se puede observar el siguiente resultado para valores altos de tiempo:

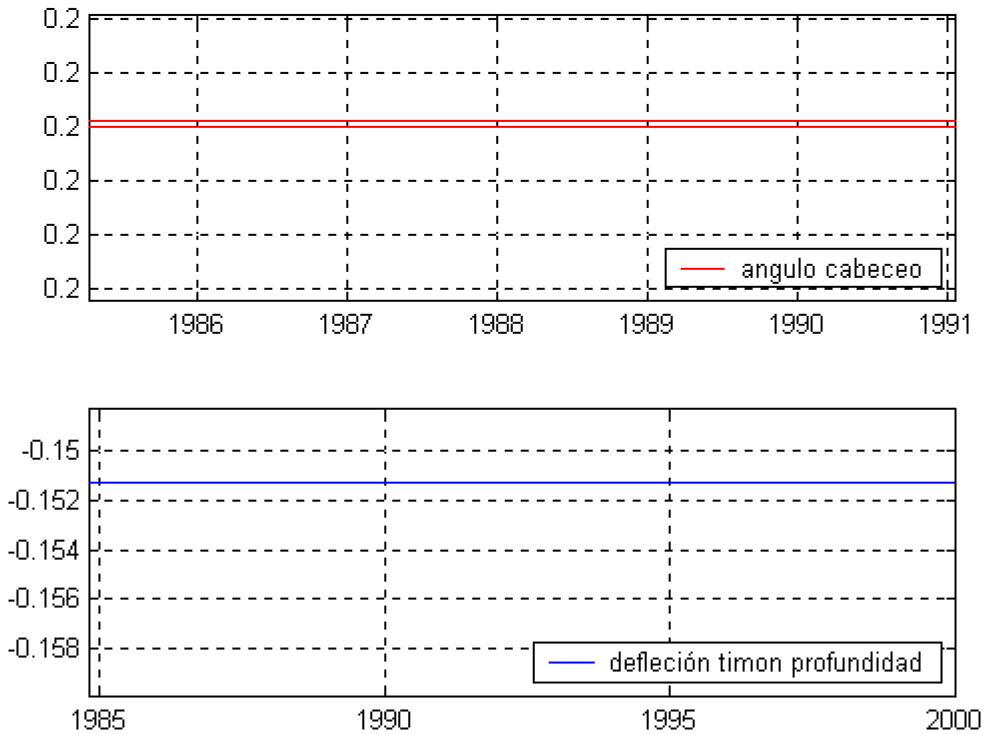


Figura 3.36

Se puede ver que la señal de control tiende a –0.151 rad en régimen permanente para una referencia en cabeceo de 0.2 rad. De aquí se deduce una ganancia estática del sistema de -0.151/0.2. Por su parte el timón de profundidad satura inferiormente a los –0.349 rad (-20º), esto implica que para que un valor de referencia pueda ser alcanzado, éste debe ser menor de:

$$\frac{0.2 * 0.349}{0.151} = 0.462$$

Efectivamente si se simula la respuesta del sistema con control LQR (Q=80 y R=5) para una referencia de 0.46rad el sistema la sigue (figura 3.37), mientras que si la referencia es 0.47 rad el timón satura permanentemente haciéndose oscilante la salida (figura 3.38).

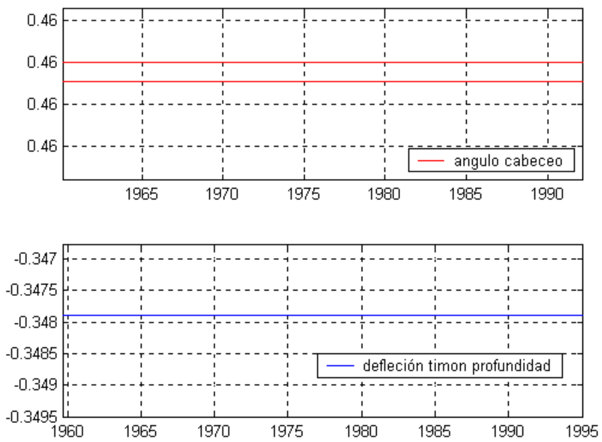


Figura 3.37

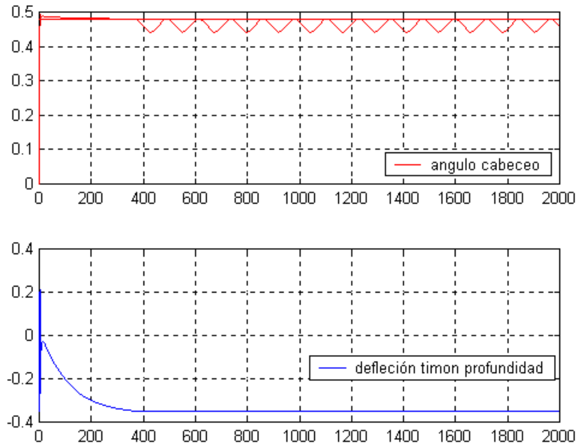


Figura 3.38

Sin embargo la respuesta del sistema ante referencia 0.48 rad no se hace oscilante hasta que no satura el actuador en el permanente, es decir pasados los 380s. Esta frontera temporal irá disminuyendo a medida que aumenta el valor de referencia en cabeceo. No obstante, en la mayoría de los casos cabe la posibilidad de desactivar el control LQR (pasando a control exclusivamente manual) una vez se haya estabilizado el ángulo de cabeceo y antes de que comiencen las oscilaciones.

3.6.2 LQR con dinámica lateral

En el caso de la dinámica lateral se tiene un sistema MIMO, con 4 funciones de transferencias inherentes. Por ello existen ganancias estáticas cruzadas que impiden realizar un análisis límite en régimen permanente como el realizado para la dinámica SISO correspondiente a la dinámica lateral. El análisis en régimen permanente de la dinámica lateral debería realizarse con empleo de controladores que partan del desacoplo del sistema, pero este no es el caso tratado en el control LQR lateral donde se ha tratado un controlador LQR multivariable.

Capítulo 4

Control GPC analítico sin restricciones

4.1 INTRODUCCIÓN AL CONTROL PREDICTIVO

4.1.1 Control predictivo basado en modelo

El Control Predictivo Basado en Modelo (MPC Model Based Predictive Control) es una estrategia de control que incluye un campo muy amplio de métodos de control con ideas comunes. Estos métodos de diseño conducen a controladores lineales que poseen prácticamente la misma estructura y presentan suficientes grados de libertad. Las ideas que aparecen en mayor o menor medida en toda la familia de controladores predictivos son básicamente tres:

- Uso explícito de un modelo para predecir la salida del proceso en futuros instantes de tiempo (horizonte).
- Cálculo de las señales de control minimizando una cierta función objetivo.
- Estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, lo que implica aplicar la primera señal de control en cada instante y desechar el resto, repitiendo el cálculo en cada instante de muestreo.

Las ventajas del control predictivo son muchas pero la que justifica principalmente su uso para el control de una aeronave es que posee intrínsecamente compensación del retardo. Otra gran ventaja del control predictivo es la fácil inclusión de restricciones (como la saturación de actuadores). Pero esta última ventaja lleva aparejada el inconveniente principal del control predictivo, una gran carga computacional. Efectivamente, la inclusión de restricciones en la función de coste a minimizar impide la existencia de una solución analítica, y es necesario el cálculo por algoritmos iterativos del vector de señales de control futuras para cada instante de muestreo.

4.1.2 Control Predictivo Generalizado. Formulación

Uno de los algoritmos de MPC es el Control Predictivo Generalizado (GPC). La idea básica del GPC es calcular una secuencia de futuras acciones de control de tal forma que minimice una función de coste multipaso. El índice a minimizar es una función cuadrática que mide por un lado la distancia entre la salida predicha del sistema y una cierta trayectoria de referencia hasta el horizonte de predicción, y por otro el esfuerzo de control necesario para obtener dicha salida. El GPC presenta como gran ventaja el poder trabajar con procesos inestables o de fase no mínima y proporcionar solución explícita en ausencia de restricciones.

La nomenclatura y ecuaciones que se presentan a continuación pueden consultarse en [16], a partir de la formulación propuesta por Clarke en 1987 [17][18].

El GPC emplea un modelo CARIMA (Controlled Auto-Regressive Integrated Moving Average) para la predicción de la salida:

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{e(t)}{D} \qquad (4.1)$$

siendo $D = 1 - z^{-1}$

$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{na} z^{-na}$ $\qquad B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{nb} z^{-nb}$

En (4.1) la perturbación viene dada por un ruido blanco coloreado por el polinomio $C(z^{-1})$ y d es el tiempo muerto del sistema. Como en la práctica es difícil encontrar el verdadero valor de este polinomio, se puede emplear como parámetro de diseño para rechazo de perturbaciones o mejora de la robustez.

El GPC usa una función de coste cuadrática de la forma:

$$J(N_1, N_2, Nu) = \sum_{j=N1}^{N2} d(j)\left|\hat{y}(t+j|t) - w(t+j)\right|^2 + \sum_{j=1}^{Nu} l(j)\left|Du(t+j-1)\right|^2 \qquad (4.2)$$

donde $\hat{y}(t+j|t)$ es la predicción óptima j pasos hacia delante de la salida del proceso con datos conocidos hasta el instante t, N_1 y N_2 son los horizontes mínimo y máximo de coste. Nu es es el horizonte de control y ä(j) y ë(j) son las secuencias de ponderación mientras que w(t+j) es la futura trayectoria de referencia. En adelante se considerará ä(j)=1 y ë(j) constante.

El objetivo es el cálculo de la futura secuencia de control u(t), u(t+1), ... de tal manera que la salida predicha del proceso $\hat{y}(t+j)$ permanezca próxima a w(t+j). Esto se logra minimizando $J(N_1,N_2,Nu)$.

Predicción óptima

Con la intención de minimizar la función de coste, se obtendrá previamente la predicción óptima de y(t+j) para j N_1 y j N_2. Considérese la siguiente ecuación diofántica:

$$1 = Ej(z^{-1})\tilde{A} + z^{-j}Fj(z^{-1}) \qquad \text{donde } \tilde{A} = DA \qquad (4.3)$$

Los polinomios Ej y Fj están definidos con grados j-1 y na respectivamente. Se pueden obtener dividiendo 1 entre $\tilde{A}(z^{-1})$ hasta que el resto pueda ser factorizado como $z^{-j}Fj(z^{-1})$. El cociente es entonces $Ej(z^{-1})$.

Si se multiplica la ecuación (4.1) por $Ej(z^{-1})z^{j}$Ä

$$\tilde{A}(z^{-1})Ej(z^{-1})\hat{y}(t+j) = Ej(z^{-1})B(z^{-1})D\,u(t+j-d-1) + Ej(z^{-1})e(t+j) \qquad (4.4)$$

Teniendo en cuenta (4.3), la ecuación (4.4) queda:

$$(1 - z^{-j}Fj(z^{-1}))\hat{y}(t+j) = Ej(z^{-1})B(z^{-1})\Delta\, u(t+j-d-1) + Ej(z^{-1})e(t+j)$$

La cual se puede escribir como

$$\hat{y}(t+j) = Fj(z^{-1})y(t) + Ej(z^{-1})B(z^{-1})\Delta\, u(t+j-d-1) + Ej(z^{-1})e(t+j) \quad (4.5)$$

Al ser el grado del polinomio $Ej(z^{-1})$ igual a j-1 los términos del ruido en la ecuación (4.5) están todos en el futuro. La mejor predicción de $\hat{y}(t+j)$ será por consiguiente:

$$\hat{y}(t+j/t) = Gj(z^{-1})\Delta\, u(t+j-d-1) + Fj(z^{-1})y(t)$$

con $Gj(z^{-1}) = Ej(z^{-1})B(z^{-1})$

Es posible demostrar que los polinomios Ej y Fj se pueden obtener recursivamente, de forma que los nuevos valores en el paso j+1 (Ej_{+1} y Fj_{+1}) sean función de los del paso j. La forma de obtener dichos polinomios recursivamente se puede resumir en:

1. Comenzar con $E_1=1$, $F_1=z(1-\tilde{A})$
2. Ir añadiendo nuevos términos a Ej con $e_{j+1,j}=f_{j,0}$
3. Calcular $f_{j+1,i} = f_{j,i+1} - f_{j,0}\,\tilde{a}_{i+1} \quad i = 0 \ldots na \quad siendo\ f_{j,na+1} = 0$

Para resolver el GPC es necesario obtener el conjunto de señales de control u(t), u(t+1), ...u(t+N) que minimizan la función de coste dada en la ecuación (4.2). Al tener el proceso un retardo de d períodos de muestreo, la salida sólo se verá influenciada por la señal u(t) después del instante d+1. Los valores N_1, N_2 y Nu que marcan los horizontes pueden ser definidos como $N_1=d+1$, $N_2=d+N$ y Nu=N. No tiene sentido hacer $N_1<d+1$ ya que los términos de (4.2) sólo dependerán de las señales de control pasadas. Por otro lado, haciendo $N_1>d+1$ los primeros puntos de la secuencia de salida, que serán los mejores estimados, no se tendrán en cuenta.

El conjunto de las j predicciones óptimas es

$$\begin{aligned} \hat{y}(t+d+1/t) &= G_{d+1}\Delta\, u(t) + F_{d+1}y(t) \\ \hat{y}(t+d+2/t) &= G_{d+2}\Delta\, u(t+1) + F_{d+2}y(t) \\ &\vdots \\ \hat{y}(t+d+N/t) &= G_{d+N}\Delta\, u(t+N-1) + F_{d+N}y(t) \end{aligned}$$

y puede ser escrito en forma matricial como:

$$y = G\,u + F(z^{-1})y(t) + G'(z^{-1})\Delta\, u(t-1) \qquad (4.6)$$

Donde

$$y = \begin{bmatrix} \hat{y}(t+d+1/t) \\ \hat{y}(t+d+2/t) \\ \vdots \\ \hat{y}(t+d+N/t) \end{bmatrix} \qquad u = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix} \qquad G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}$$

$$G'(z^{-1}) = \begin{bmatrix} z(G_{d+1}(z^{-1}) - g_0) \\ z^2(G_{d+2}(z^{-1}) - g_0 - g_1 z^{-1}) \\ \vdots \\ z^N(G_{d+N}(z^{-1}) - g_0 - g_1 z^{-1} - \cdots - g_{N-1} z^{-(N-1)}) \end{bmatrix} \qquad F(z^{-1}) = \begin{bmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{bmatrix}$$

Al depender los últimos términos de la ecuación (4.6) sólo del pasado, pueden agruparse en f, dando lugar a:

$$\mathrm{y} = \mathrm{G\,u} + \mathrm{f} \qquad (4.7)$$

Obtención de la ley de control

Entonces la ecuación (4.2) puede escribirse como:

$$\mathrm{J} = (\mathrm{G\,u} + \mathrm{f} - \mathrm{w})^{\mathrm{T}}(\mathrm{G\,u} + \mathrm{f} - \mathrm{w}) + l\,\mathrm{u}^{\mathrm{T}}\mathrm{u} \qquad (4.8)$$
$$\text{con} \quad \mathrm{w} = [\mathrm{w}(t+d+1) \quad \mathrm{w}(t+d+2) \quad \cdots \quad \mathrm{w}(t+d+N)]^{\mathrm{T}}$$

La ecuación (4.8) se puede poner como:

$$\mathrm{J} = \frac{1}{2}\mathrm{u}^{\mathrm{T}}\mathrm{H\,u} + \mathrm{b\,u} + \mathrm{f}_0 \qquad (4.9)$$
$$\text{donde} \quad \mathrm{H} = 2(\mathrm{G}^{\mathrm{T}}\mathrm{G} + l\,\mathrm{I})$$
$$\mathrm{b} = 2(\mathrm{f} - \mathrm{w})^{\mathrm{T}}\mathrm{G}$$
$$\mathrm{f}_0 = (\mathrm{f} - \mathrm{w})^{\mathrm{T}}(\mathrm{f} - \mathrm{w})$$

El mínimo de J, siempre que no existan restricciones en la señal de control, puede ser calculado igualando a cero el gradiente de J, lo cual conduce a:

$$\mathrm{u} = -\mathrm{H}^{-1}\mathrm{b}^{\mathrm{T}} \qquad (4.10)$$

Debido al uso de la estrategia deslizante, sólo se aplica realmente el primer elemento del vector u, repitiendo de nuevo el mismo procedimiento al siguiente instante de muestreo.

Implementación del algoritmo en Matlab

La programación del algoritmo del GPC es sencilla en MATLAB, ya que los cálculos matriciales, que son los que provocan la mayor complejidad numérica, pueden ser resueltos fácilmente. La subrutina gpc.m en MATLAB que realiza los cálculos de la predicción óptima se muestra en el apéndice A.2.1.

Para calcular la señal de control es necesario realizar un programa con una llamada a la subrutina gpc. La señal de control se obtiene a partir de los valores devueltos por dicha subrutina mediante los siguientes cálculos:

```
h=-inv(g'*g+lambda*eye(N))*g';
kr=-h(1,:);
ku=h*gp;
ku=ku(1,:);
ky=h*fb;
ky=ky(1,:);
```

Los vectores kr, ku y ky servirán para construir la señal de control en un modelo de Simulink teniendo en cuenta que:

kr: es un vector 1 x N.
ku: es un vector 1 x M. Siendo M = d + orden del numerador del modelo del sistema.
ky: es un vector 1 x Q. Siendo Q el orden del denominador del modelo del sistema

La señal de control se construye entonces según la siguiente ley:

$$\begin{aligned}\Delta u(t) = {} & ku(1)\Delta u(t-1) + ku(2)\Delta u(t-2) + \cdots + ku(M)\Delta u(t-M) + \\ & ky(1)y(t) + ky(2)y(t-1) + \cdots + ky(Q)y(t-M+1) + \\ & kr(1)w(t+1) + kr(2)w(t+2) + \cdots + kr(N)w(t+N)\end{aligned}$$

Si ahora se considera una referencia constante y se sustituye el operador Ä por $1\text{-}z^{-1}$, se llega a la siguiente ley de control:

$$\begin{aligned}u(t) = {} & (1+ku(1))u(t-1) + (ku(2)-ku(1))u(t-2) + \cdots \\ & + (ku(M)-ku(M-1))u(t-M) - ku(M)u(t-M-1) + \\ & ky(1)y(t) + ky(2)y(t-1) + \cdots + ky(Q)y(t-M+1) + \\ & sum(kr)w\end{aligned}$$

4.2 GPC CON DINÁMICA LONGITUDINAL

4.2.1 Especificaciones de control

En este punto se partirá de la dinámica longitudinal del avión expresada en la ecuación (2.37) o su equivalente en función de transferencia dada en (2.44). Con dicha dinámica se realizará un control GPC sobre el ángulo de cabeceo (pitch) del avión, actuando sobre el timón de profundidad.

Este apartado está dividido en subapartados a fin de comprobar los efectos sobre los resultados de los análisis al introducir unas dinámicas de actuadores y piloto. Así, primero se analizará la respuesta del sistema con control GPC considerando unos actuadores ideales (respuesta instantánea e ilimitada) y sin considerar la acción del piloto. En el siguiente apartado se estudiará el efecto de introducir un actuador con saturación y una respuesta con constante de tiempo finita. Finalmente se introducirá la dinámica del piloto manteniendo los actuadores del apartado anterior.

Como punto de partida las especificaciones sobre la respuesta en cabeceo del avión deseada serán las expuestas ya para el control LQR:

Referencia cabeceo 0.2 rad (11.46º)

Sobreoscilación < 10%
Tiempo subida < 2 seg.
Error en permanente < 2%
Tiempo establecimiento < 10 seg.

Dichos valores se han tomado del Control Tutorial for Matlab (The University of Michigan) [10].

4.2.2 Sistema simple

En este punto se simulará un control GPC sobre la dinámica longitudinal del sistema sin considerar las dinámicas de piloto y actuador. No se tratará de ajustar la respuesta a las especificaciones sino que se intentará ver los efectos sobre el control GPC relativos a cambios en sus parámetros. El ajuste de la respuesta a las especificaciones se intentará realizar con la dinámica completa en los siguientes apartados.

En el archivo principal.m incluido en el apéndice A.2.2 se calculan los parámetros para el control GPC mediante el uso de la subrutina gpc. La sucesión de tareas realizadas en dicho archivo se muestran en la figura 4.1.

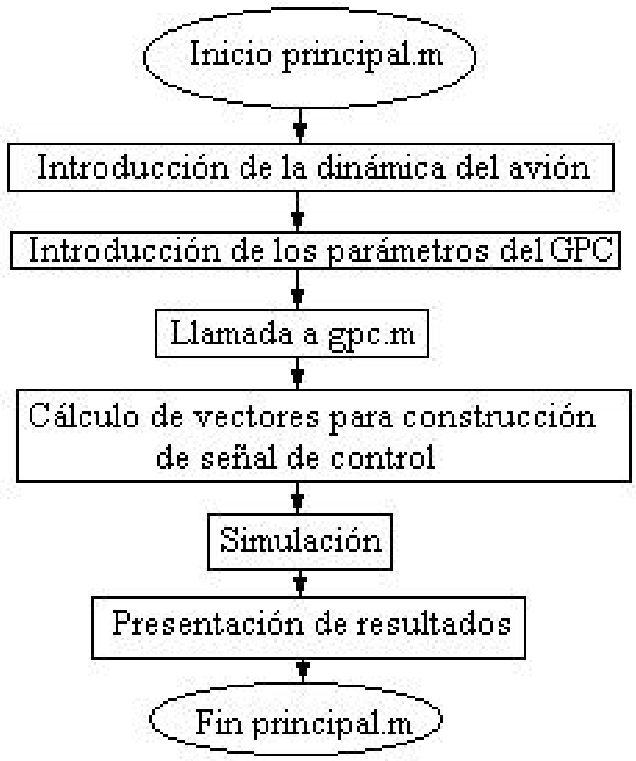


Figura 4.1

La figura 4.2 muestra el modelo Simulink donde se simula el control GPC con los vectores obtenidos tras la ejecución del programa anterior.

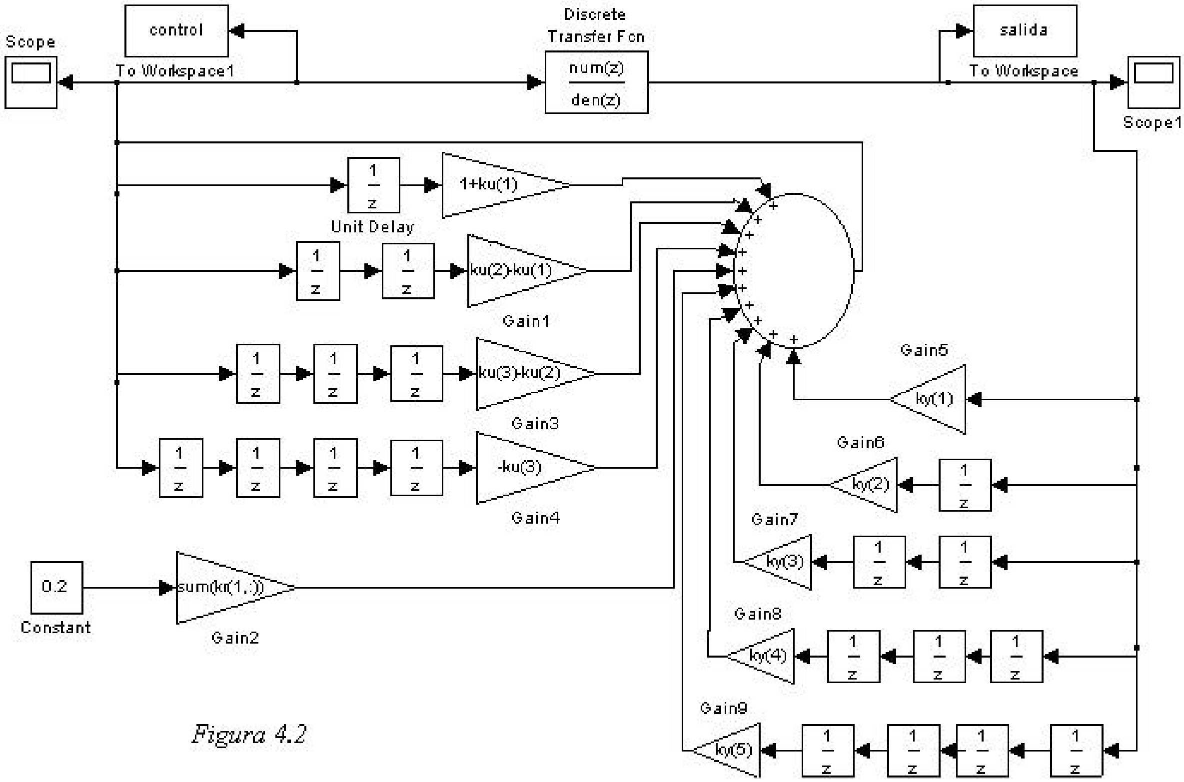


Figura 4.2

El tiempo de muestreo de los retrasos unitarios y de la función de transferencia discreta es 0.008 segundos. Pues, este periodo se usó en el apartado 2.7.2 para la obtención de la función de transferencia discreta dada en (2.44) y que se ha tomado como punto de partida en A.2.2.

La figura 4.3 muestra el resultado de la simulación para tres valores de N distintos y ë = 0.8. Se observa que a medida que aumenta N disminuye la sobreoscilación y el esfuerzo de control, mientras que el sistema se hace un poco más lento. Llega un momento en que el aumento de N no implica una mejora perceptible, este valor de N se puede fijar en 150. De la misma forma para valores de N inferiores a 45 el sistema no se estabiliza, esto se puede observar en la figura 4.4 correspondiente a N=40. En todo caso estable, el error en régimen permanente es prácticamente nulo.

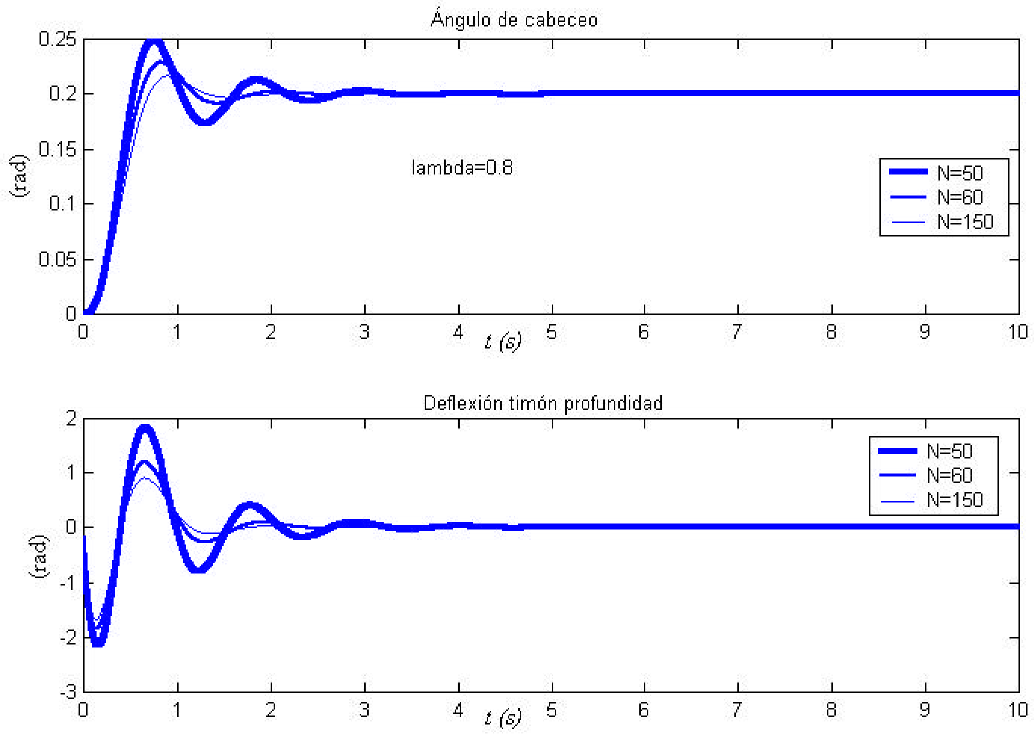


Figura 4.3

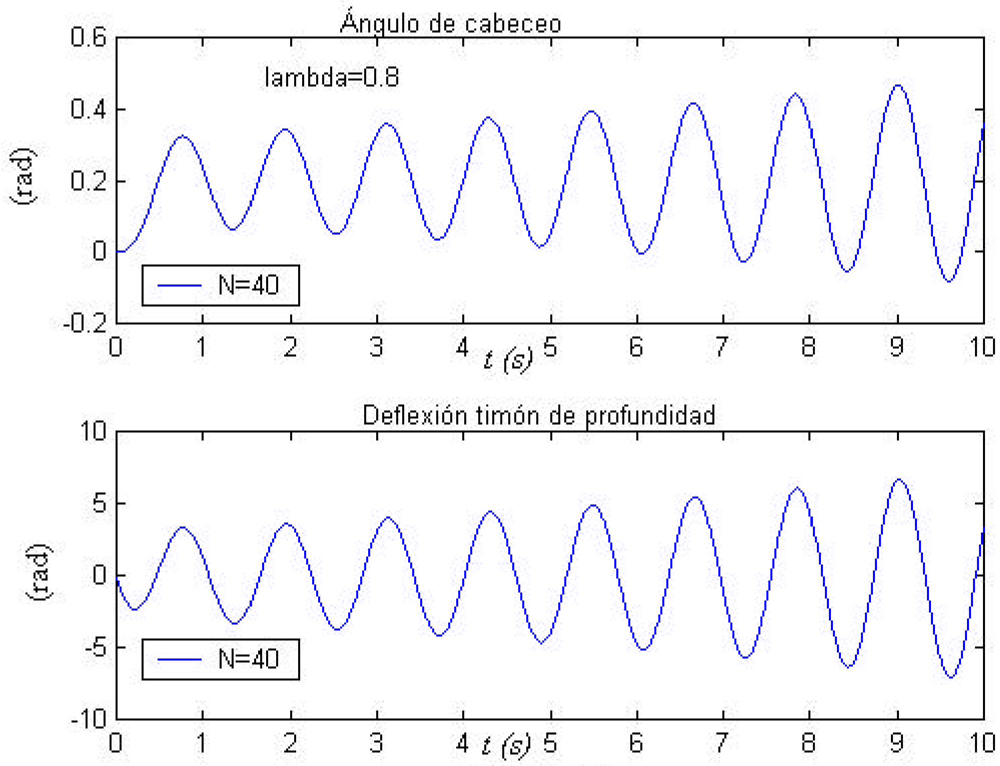


Figura 4.4

La figura 4.5 muestra el resultado de la simulación del control GPC para tres valores distintos de ë con N = 150. Se observa que a medida que aumenta ë disminuye considerablemente el esfuerzo de control, se mantiene la misma sobreoscilación, la respuesta se ralentiza y el error permanece prácticamente nulo en todo caso.

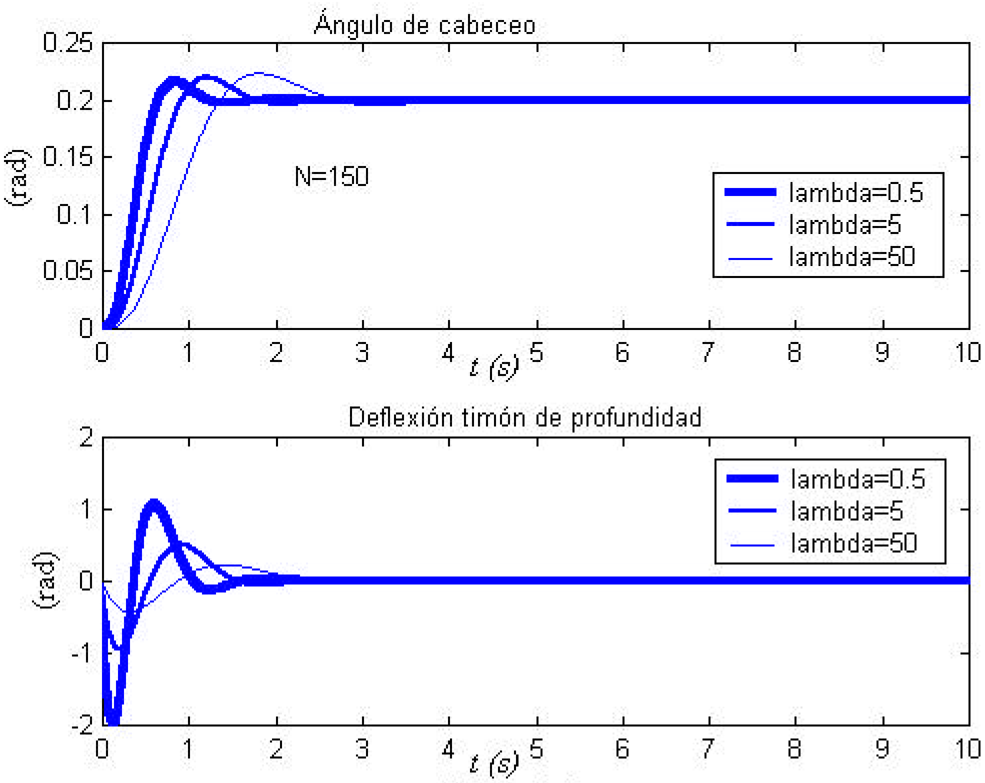


Figura 4.5

4.2.3 Sistema con actuador real

En este apartado se incluirá la dinámica del actuador sobre el timón de profundidad ya expuesta en el apartado 3.3.2 (LQR). Dicha dinámica consistirá en un sencillo sistema de primer orden (4.11) discretizado, junto con un bloque de saturación se la señal de actuación entre [-0.349, 0.698] rads.

$$\frac{37}{s+37} \quad \text{dinámica actuador} \quad (4.11)$$

El efecto de la inclusión del actuador en el control GPC tiene dos importantes diferencias con respecto al control LQR:

- Si el controlador satura no podrá recuperarse, a no ser que se incluyan restricciones en la resolución del algoritmo GPC. Pero en tal caso no se llegará a una ley explícita con el consiguiente aumento del coste computacional. En este punto se sintonizará el GPC para evitar la saturación del actuador.
- El efecto ralentizador de la dinámica del actuador es poco perceptible en el control GPC, pues ya de por sí el esfuerzo de control puede moderarse manipulando ë o suavizando la señal de referencia.

En el apéndice A.2.3 se muestra el archivo en Matlab para el control GPC con actuador. Las tareas realizadas en dicho archivo son las mismas mostradas en la figura 4.1, incluyéndose la definición de la dinámica del actuador junto a la del avión.

Al haber aumentado el orden del sistema también aumenta la dimensión de los vectores para el control GPC, por lo que el modelo Simulink mostrado en la figura 4.2 deja de ser válido y hay que sustituirlo por el mostrado en la figura 4.6. En este modelo también se ha incluido un filtro paso de baja (Butterworth con frecuencia corte 2.5 rad/seg) para el suavizado de la señal de referencia, con lo que se consigue disminuir el esfuerzo de control. Dicho filtro se puede sustituir por un sistema de primer orden (4.12).

$$\frac{1}{t\,.\,s+1} \quad \text{con} \quad t = 1/2.5 \qquad (4.12)$$

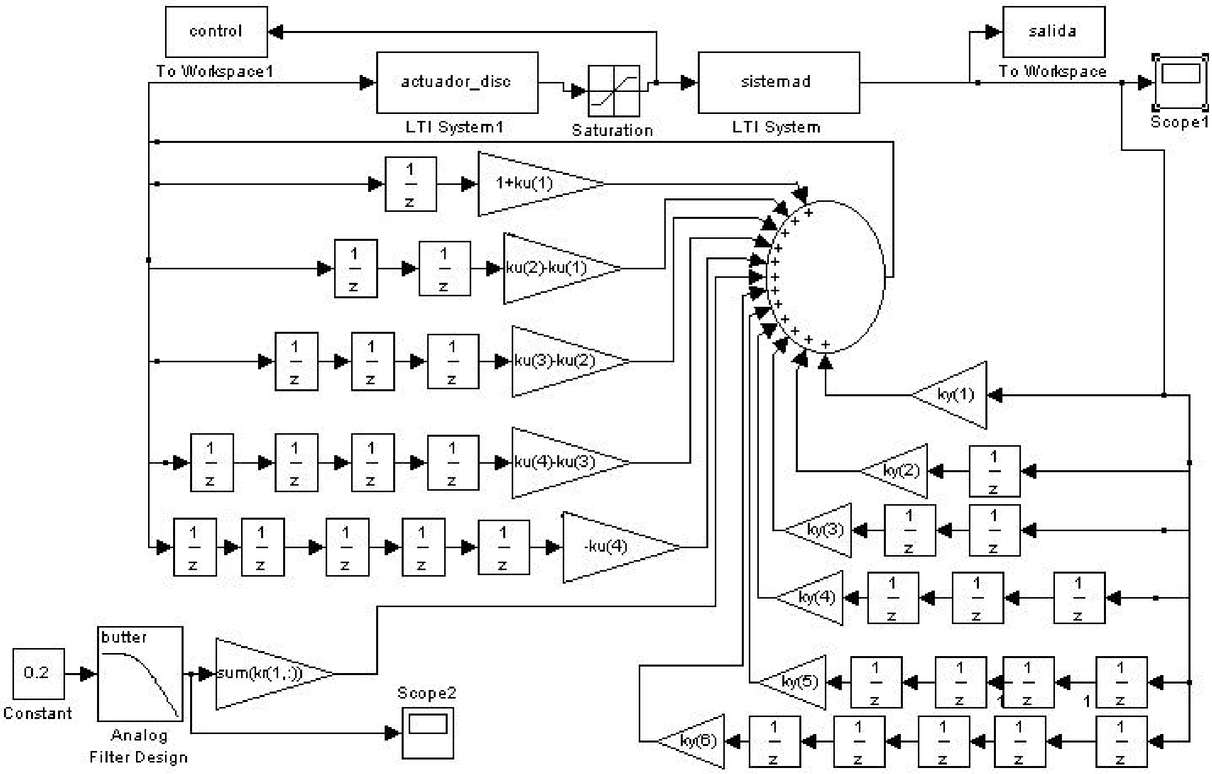


Figura 4.6

Antes de mostrar los resultados de la simulación es importante señalar dos modificaciones necesarias para la correcta simulación. La primera de ellas es que al haber introducido en el modelo Simulink un filtro analógico, es necesario configurar el paso inicial a 0.008 segundos (seleccionar en simulation – parameters –Initial Step size) para que todos los vectores de la representación de resultados tengan la misma dimensión. La segunda es reducir el vector numerador, cuyos elmentos [n_1 n_2 n_3 n_4 n_5 n_6] son los coeficientes del polinomio numerador ($n_1z^5+n_2z^4+n_3z^3+n_4z^2+n_2z+n_1$), eliminando $n_1=0$ para conseguir un vector ku de dimensión adecuada.

Variando los parámetros N y ë del GPC se obtiene para el caso con referencia suavizada (ref2) y sin suavizar (ref1) las respuestas mostradas en la figura 4.7. Esta respuesta es la mejor que se puede conseguir en ambos casos, pues la disminución en ë de una unidad lleva a la saturación del actuador sin que el controlador se recupere como se observa en la figura 4.8. La figura 4.7 muestra como la simulación con referencia suavizada es algo más lenta, pero a cambio presenta unas características mucho mejores en sobreoscilación.

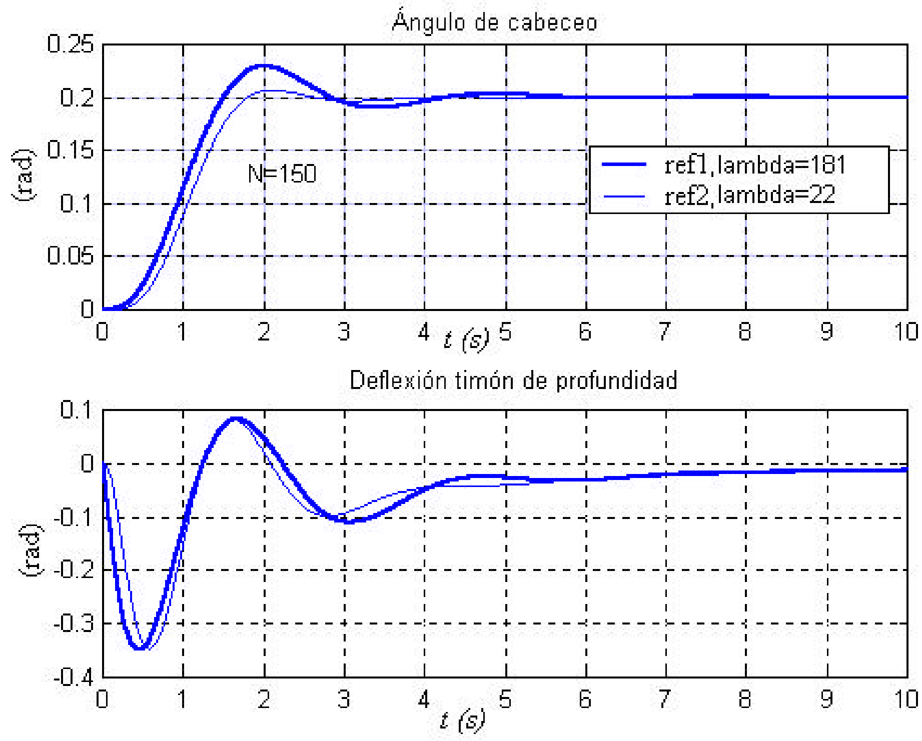


Figura 4.7

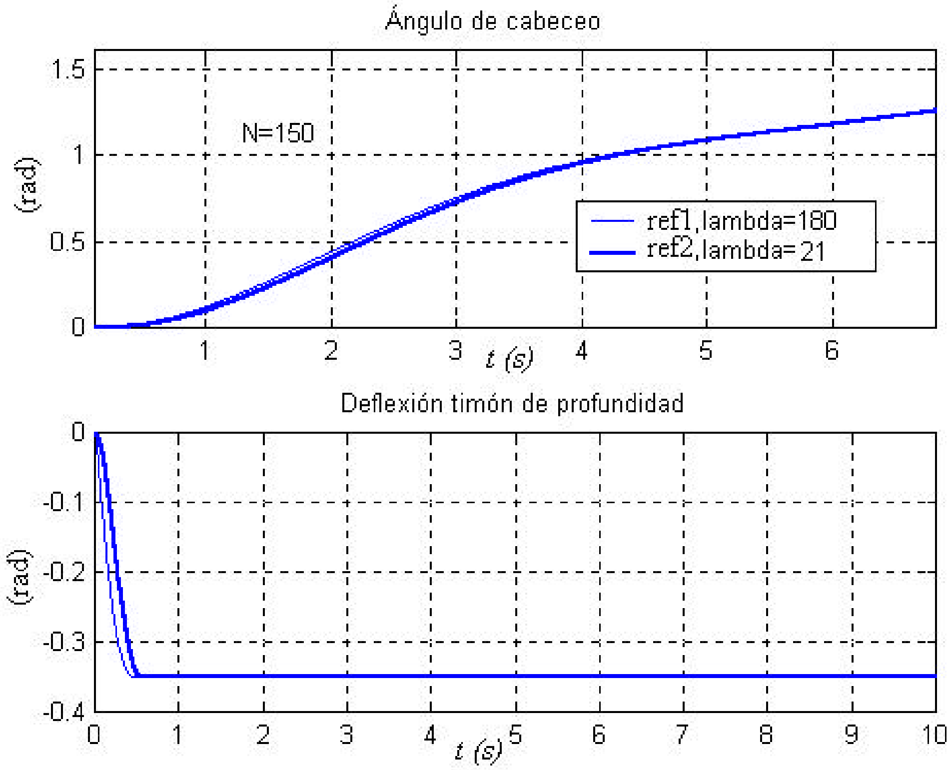


Figura 4.8

4.2.4 Sistema con actuador real y piloto

En este apartado se incluirá la dinámica del piloto ya expuesta en el apartado 3.3.3 (LQR). El comportamiento del piloto ante cambios longitudinales se puede expresar como (4.13) [19].

$$0.74(0.27S + 1)e^{-0.3S} \qquad (4.13)$$

Las modificaciones a introducir en el archivo de Matlab son:

```
%dinámica piloto sin retardo (se añade en Simulink)
s=tf('s');
piloto_cont=0.74*(0.27*s+1);
piloto_disc=c2d(piloto_cont,0.008,'tustin');

%dinámica total
sistema_total=sistemad*actuador_disc*piloto_disc;
[num,den]=tfdata(sistema_total,'v');
num=num(2:7);                  %eliminar un cero de la izquierda

%Retardo piloto
d=37;                       %Retardo piloto:37*0.008 0.3seg
```

Al introducir la dinámica del piloto aumenta el orden del sistema por lo que debe modificarse el esquema de control. Además al considerar el retardo introducido por el piloto (37 periodos de muestreo) el vector de realimentación ku pasa a tener 42 elementos, por lo que el esquema Simulink quedaría demasiado abultado. Para evitar esto se calcula la función de transferencia equivalente a la realimentación del vector de control con unas simples líneas en Matlab:

```
%cálculo de f.t equivalente a la realimentación de la señal de control
ku(1,d+6)=0;
z=tf('z',0.008);
control_u=z^-1*(1+ku(1,1));
for i=2:1:d+6
   control_u=control_u+z^-i*(ku(1,i)-ku(1,i-1));
end
```

Con todo esto el esquema de control en Simulink queda como muestra la figura 4.9.

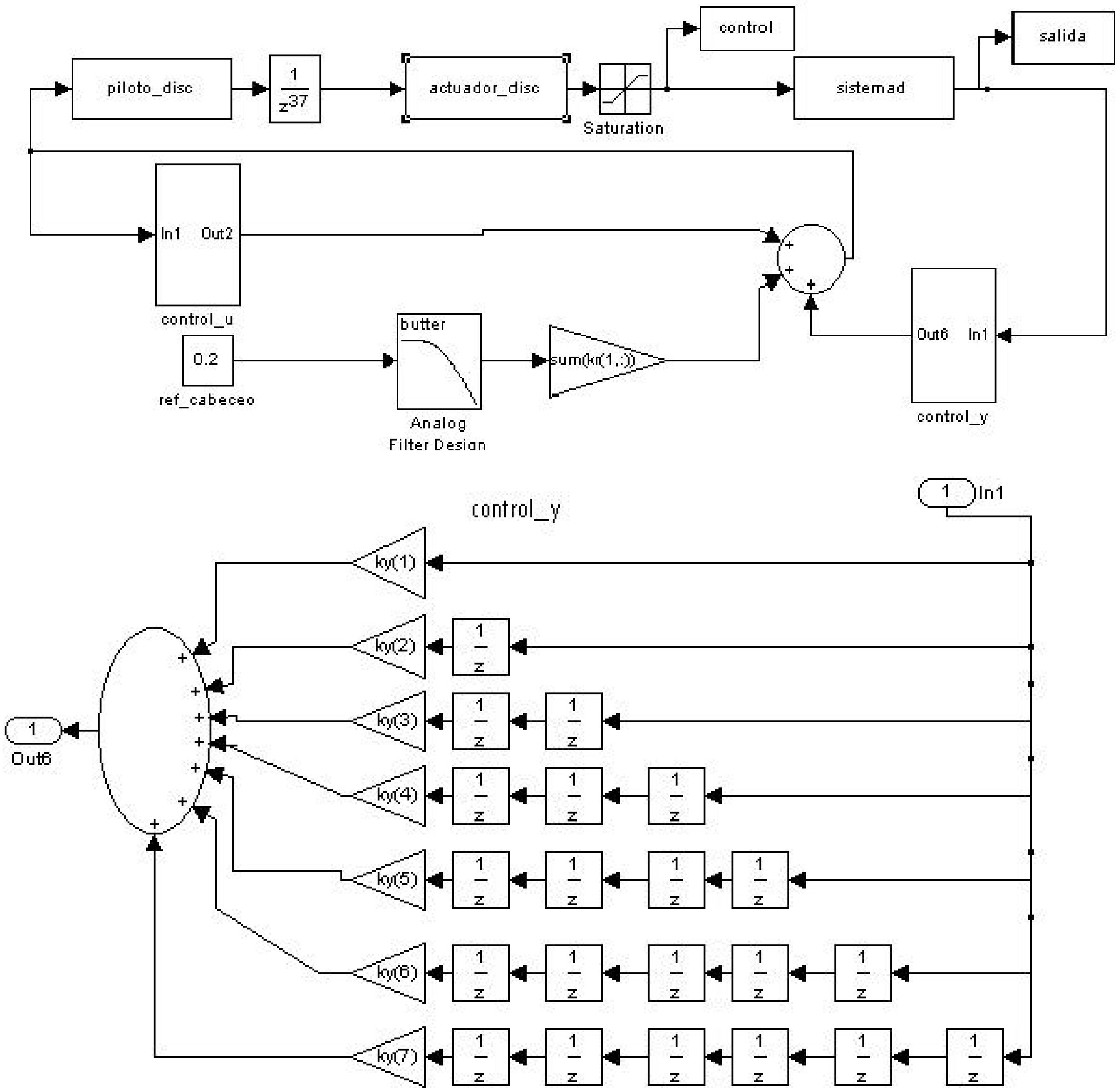


Figura 4.9

La figura 4.10 muestra la respuesta del sistema cuando se considera el retardo del piloto y cuando no. Se observa que el control GPC posee intrínsecamente compensación del retardo, pues la respuesta del sistema considerando el retardo es la respuesta sin considerar retardo desplazada en el tiempo. Atendiendo a la figura 4.10 se observa el cumplimiento de todas las especificaciones dadas en 4.2.1 excepto la relativa al tiempo de subida.

te =3s < 10s
e.r.p 0 < 0s
ts = 2.07s > 2s
S.O. = 3% < 10%

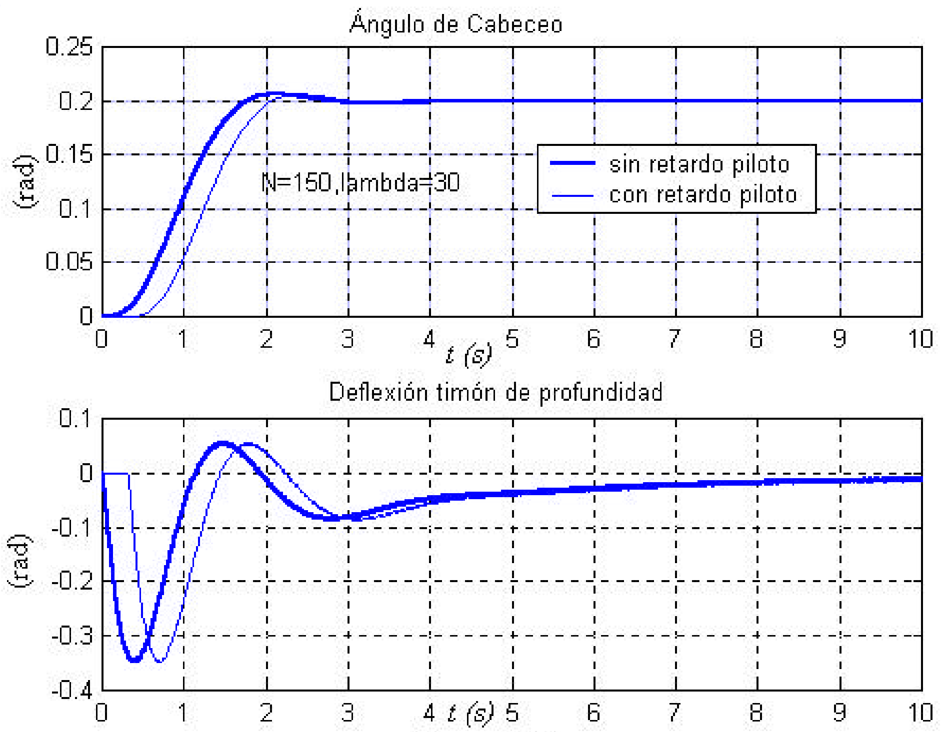


Figura 4.10

Variando los parámetros del GPC es relativamente sencillo conseguir una respuesta que cumpla todas las especificaciones. Para N=100 y ë=60 se obtiene la respuesta mostrada en la figura (4.11). Se observa que la S.O. ha aumentado con respecto a la respuesta mostrada en la figura (4.10), pero a cambio se ha conseguido disminuir el tiempo de subida.

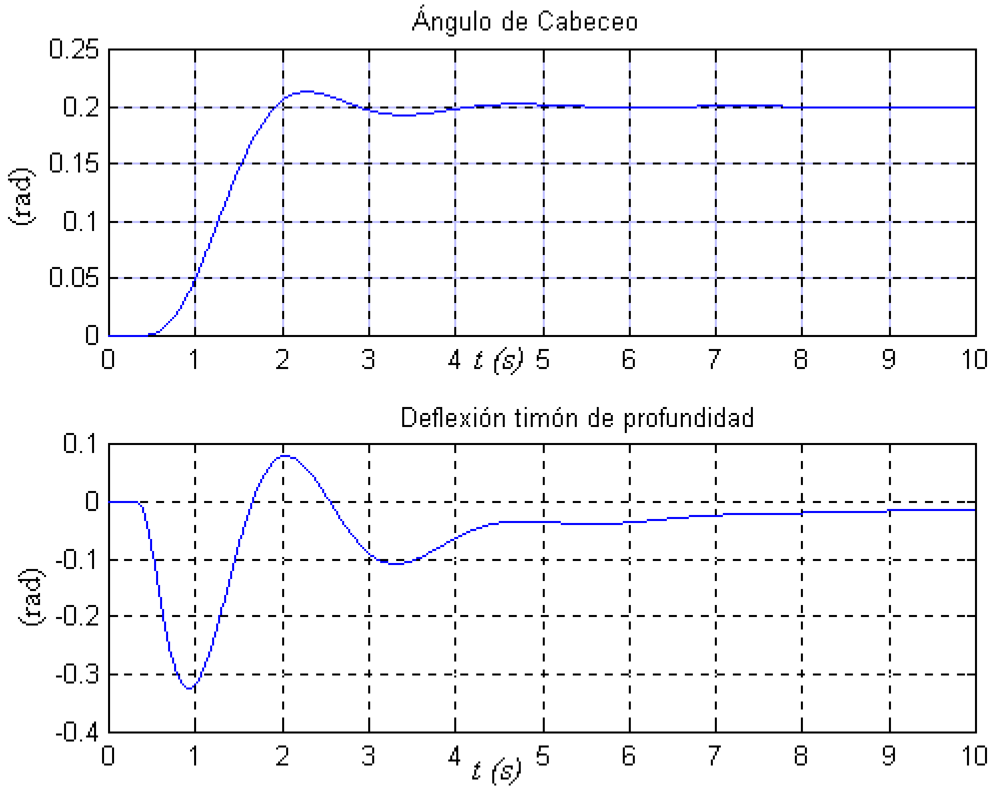


Figura 4.11

te =5s < 10s
e.r.p 0 < 0s
ts = 1.91s < 2s
S.O. = 6.5% < 10%

4.3 GPC CON DINÁMICA LATERAL

Debido a las limitaciones en saturación vistas en el caso longitudinal del control GPC analítico, se desarrollará en un apartado posterior un GPC iterativo considerando restricciones. En este GPC será muy sencilla la inclusión de dinámicas MIMO, por lo que se deja el estudio de la dinámica lateral para tal caso.

4.4 COMPORTAMIENTO ANTE PERTURBACIONES

El control GPC tiene la propiedad de rechazar tanto las perturbaciones instantáneas como permanentes. Sin embargo, en el caso de que el vector de realimentación de salidas ky contenga elementos de grandes magnitudes (el caso que se trata), el rechazo de las perturbaciones exige grandes valores de la señal de control. Estos valores provocarán la saturación de la señal de control, ante la cual el controlador GPC analítico no es capaz de recuperarse. Para solucionar esta situación se diseñará en apartados posteriores un controlador GPC iterativo que tenga en cuenta las restricciones en los valores de la señal de control.

A modo de ejemplo la figura 4.12 muestra la respuesta del sistema ante una perturbación en la salida mantenida del 10% del valor de referencia (0.1*0.2=0.02rad) incluyendo el bloque saturación del actuador. La figura 4.13 muestra la respuesta eliminando dicho bloque de saturación, junto con una ampliación donde se observa que efectivamente el sistema tiende a la referencia.

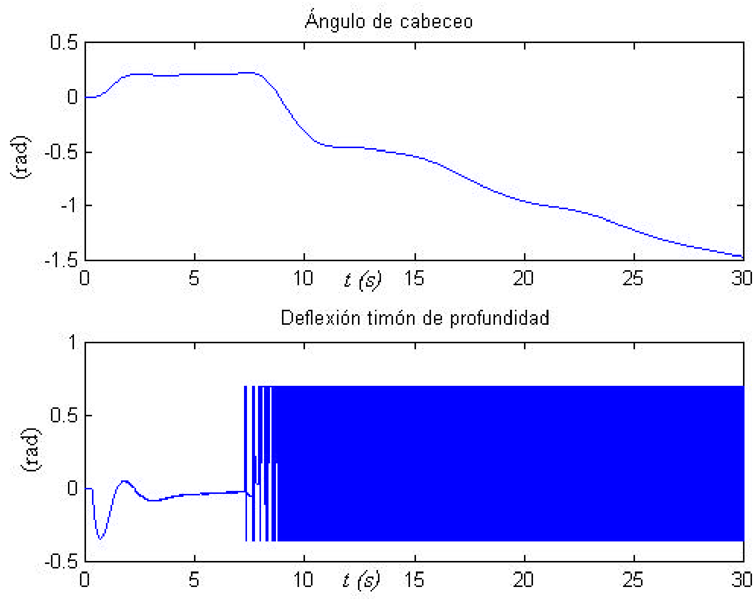


Figura 4.12

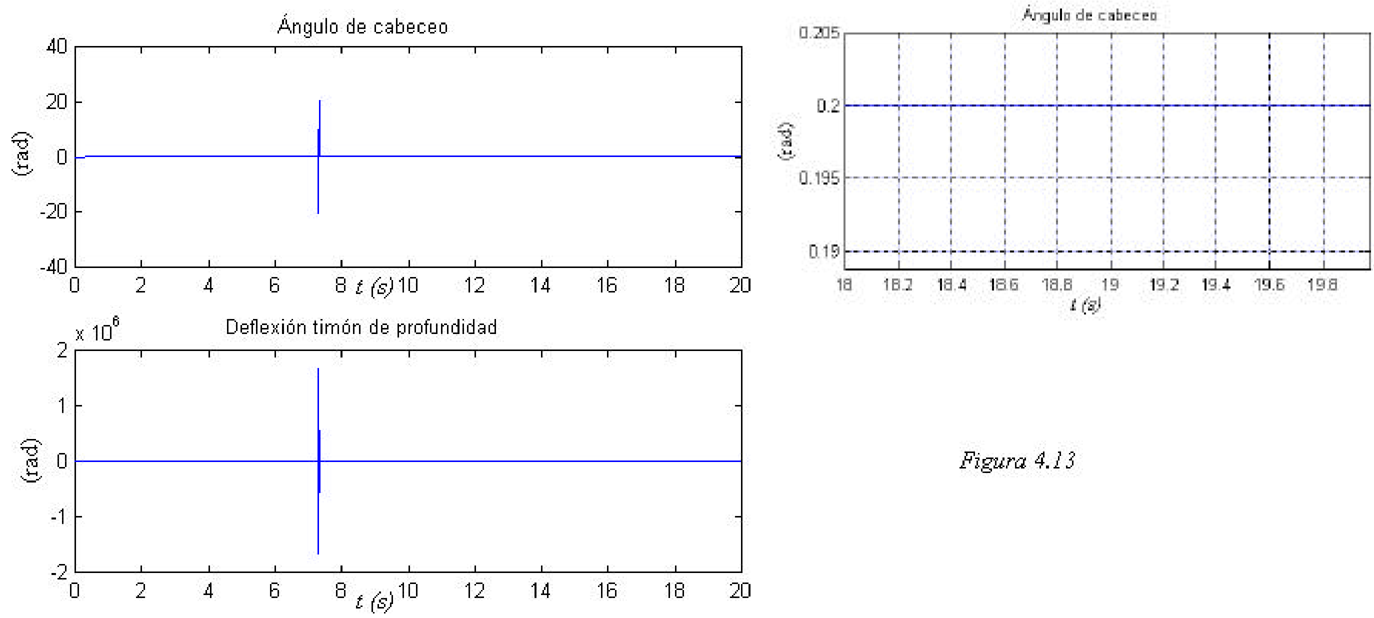

Ángulo de cabeceo
(rad)
t (s)
x 10^6
Deflexión timón de profundidad
(rad)
t (s)
Ángulo de cabeceo
(rad)
t (s)

Figura 4.13

Capítulo 5

Control GPC iterativo con restricciones

5.1 INTRODUCCIÓN AL CONTROL PREDICTIVO CON RESTRICCIONES

Se ha visto en el punto 4 que el control GPC genera una solución analítica en ausencia de restricciones. Sin embargo, si se consideran restricciones, la solución de la minimización de la función objetivo, se obtiene utilizando un algoritmo numérico de optimización con restricciones [20]. Para esta clase de problemas, en que la función objetivo es cuadrática, se cumple:

- La función objetivo debe ser convexa para que exista solución
- Debe haber al menos un valor para el cual las variables de optimización cumplan todas las restricciones impuestas.
- El espacio de restricciones debe ser convexo

La minimización de la función objetivo con restricciones lineales se conoce como programación cuadrática [20], quedando el problema de optimización para el GPC queda definido por:

$$\begin{cases} \text{Min } J = (G\,u+f\ \ -r)^T(G\,u+f\ \ -r)+\text{ë}\,u\ \ ^T u \\ \text{s.a } A\,u\ b \end{cases}$$

con A, b determinados de acuerdo al tipo de restricción, y r es el vector de referencias en el horizonte de predicción.

Las restricciones más comunes son las aplicadas al rango y velocidad de cambio de las variables manipuladas (actuaciones) y controladas (salidas). Para el caso de actuadores con limitación de rango (el caso tratado aquí) la restricción está dada matricialmente por:

$$\begin{cases} u_{\min}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} \leq \begin{bmatrix}u(t)\\u(t+1)\\\vdots\\u(t+\ Nu+1)\end{bmatrix} \leq \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}u_{\max} \quad \Leftrightarrow\ u_{\min}\,\overline{1} \leq\ u \leq\ \overline{1}u_{\max} \end{cases}$$

La resolución del problema con restricciones lleva aparejada una gran carga computación con factorizaciones de matrices incluidas. Para evitar situaciones de mal condicionamiento y desbordamiento, los valores de los horizontes de control y predicción no deben tomarse excesivamente grandes. De ahora en adelante se tomará 50 como valor para el horizonte de predicción, y 20 para el horizonte de control. Se han comprobado posteriormente los buenos resultados alcanzados.

Las funciones necesarias para programar el control GPC con restricciones han sido cedidas por el Profesor Daniel Limón Marruedo del Departamento de Ingeniería de Sistemas y Automática. Algunas de ellas han sido modificadas por mí. En los apéndices A.3.2, A.3.3, A.3.4, A.3.5, A.3.6, A.3.7 y A.3.8 se incluyen los archivos con las modificaciones necesarias.

Para la programación en Matlab y simulación en Simulink de un control GPC con restricciones será necesario el uso de S-functions (system functions). Una introducción al manejo de estas funciones de sistema se incluye en el apéndice A.3.1. Para una descripción más detallada se puede consultar [21].

5.2 GPC ITERATIVO CON DINÁMICA LONGITUDINAL

El problema de control es el mismo expuesto en el punto 4.2. Es decir, se quiere seguir una referencia de 0.2 rad en el ángulo de cabeceo del avión actuando sobre el timón de profundidad. Para ello se considerará una dinámica completa compuesta por piloto, actuador y dinámica longitudinal avión (idéntico a punto 4.2.3). Adicionalmente se incluirá en la cadena directa un filtro paso baja (ecuación 5.1) para disminuir las oscilaciones de alta frecuencia en la señal de actuación.

$$H(s) = \frac{1.4}{(0.5s+1)^2} \quad \overset{Tm=0.06}{\Leftrightarrow} \quad H(z) = 10^{-2}\,\frac{(9.25z+8.54)}{(z-0.887)^2} \qquad (5.1)$$

En el archivo de Matlab principal.m, incluido en el apéndice A.3.2, se definen las dinámicas, los parámetros del controlador, y se calcula la predicción. La sucesión de tareas realizadas en dicho archivo se muestran en la figura 5.1.

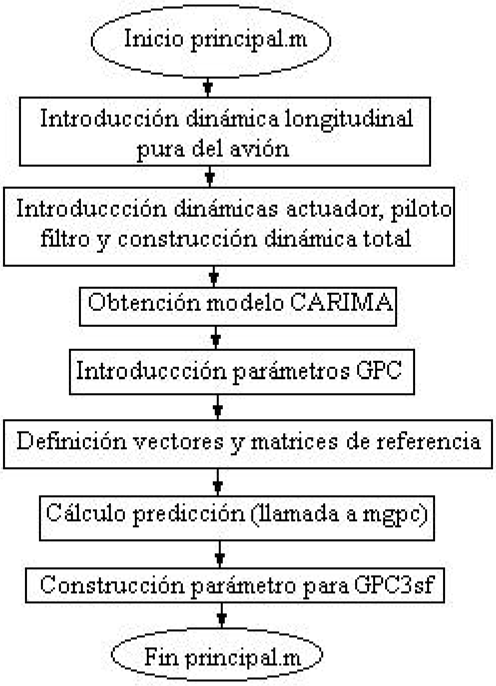


Figura 5.1

Las figuras 5.2 y 5.3 muestran el modelo de Simulink para el control GPC con restricciones. El modelo incluye la posibilidad de añadir restricciones en las variables de salida y en la velocidad de cambio de la variable de control, dichas restricciones estarán desactivadas.

En la figura 5.3 se observa la inclusión de un bloque S-function que implementa la función GPC3sf incluida en el apéndice A.3.5, cuya ejecución se realiza en cada paso de muestreo. Esta función es la encargada de resolver el problema de programación cuadrática asociado al GPC con restricciones, para ello hace uso de la función quadprog de Matlab.

La función GPC3sf recibe como parámetros los correspondientes a una S-function (t, x, u, flag, ver A.3.1) más un parámetro adicional par (ver figura 5.3). Este parámetro es una estructura creada tras la ejecución del archivo principal.m incluido en el apéndice A.3.2. La estructura par incluye distintas constantes (nº entradas, nº salidas, horizonte de predicción, ...) y los valores para la predicción (GN123, GpN12, FN12) obtenidos tras la resolución de la ecuación diofántica con la ejecución del archivo Mgpc.m (apéndice A.3.7). Todos estos valores serán usados por la función GPC3sf.

A continuación se resaltan algunos valores de la estructura par en el caso del control de la dinámica longitudinal, que serán útiles en explicaciones posteriores: par.na=9 (orden del denominador de la cadena directa discreta), par.nb=14 (orden del numerador de la cadena directa discreta), par.m=1 (nº de entradas al sistema, deflexión timón profundidad), par.n=1 (nº de salidas del sistema, ángulo de cabeceo).

En las siguientes líneas se comentan algunas sentencias de código del archivo GPC3sf.m (en particular de la función mdlInitializeSizes incluida en este archivo) incluidas en el apéndice A.3.5:

- sizes.NunContStates=0: indica que existen 0 estados continuos.

- sizes.NumDiscStates=(struc.na+1)*struc.n+(struc.nb+1)*struc.m: la parte derecha de la ecuación toma valor 25 para el caso longitudinal, y definen el nº de estados discretos. Los elementos x(1) a x(10) se asocian con las salidas medidas y_k a y_{k-9} respectivamente, los elementos x(11) a x(25) se asocian con las señales de control u_{k-1} a u_{k-15} respectivamente. Al finalizar cada iteracción, x(11) tomará el valor de la señal de control actual a aplicar por el controlador.

- sizes.NumOutputs=struc.m: define el nº de salidas del bloque S-function GPC3sf, que coincide con el nº de entradas al sistema a controlar (nº señales de actuación). Toma valor 1 para el caso longitudinal, indicando como única salida del bloque GPC3sf la señal de control calculada para aplicar al timón de profundidad.

- sizes.NumInputs=2*struc.n+(4*struc.m+2*struc.n+4+1): define el nº de entradas al bloque S-function GPC3sf. Dichas entradas se identifican dentro de la función GPC3sf.m por el vector u. Toma valor 13 (ver figura 5.3) para el caso longitudinal correspondiendo a:

    - u(1)→ angulo de cabeceo
    - u(2)→ referencia cabeceo
    - u(3), u(4), u(5)→ flag de activación restricción en velocidad variación de la señal de control, valores máximos y mínimos para dicha variación (no se usa).
    - u(6), u(7), u(8)→ flag de activación restricción en rango de la señal de control, valores máximos y mínimos para dicho rango.
    - u(9), u(10), u(11)→ flag de activación restricción en rango de la señal de salida, valores máximos y mínimos para dicho rango (no se usa).
    - u(12)→flag de activación de restricciones en general.
    - u(13)→ë, peso en la función de coste para la señal de actuación.

- sizes.DirFeedthrough=0: desactiva la alimentación directa. Indica que en la función mdlOutputs (correspondiente a flag=3 dentro de GPC3sf.m) no se hará uso de las entradas al bloque S-function GPC3sf.m.

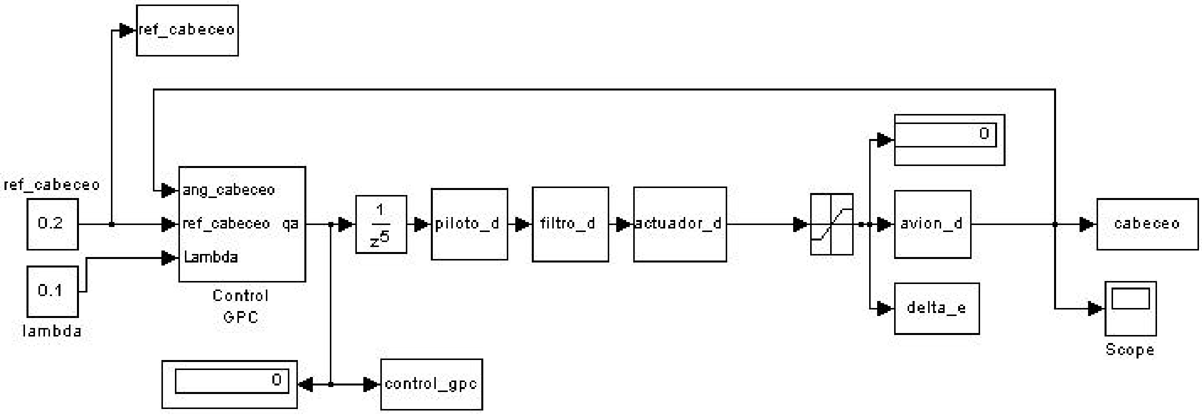


Figura 5.2

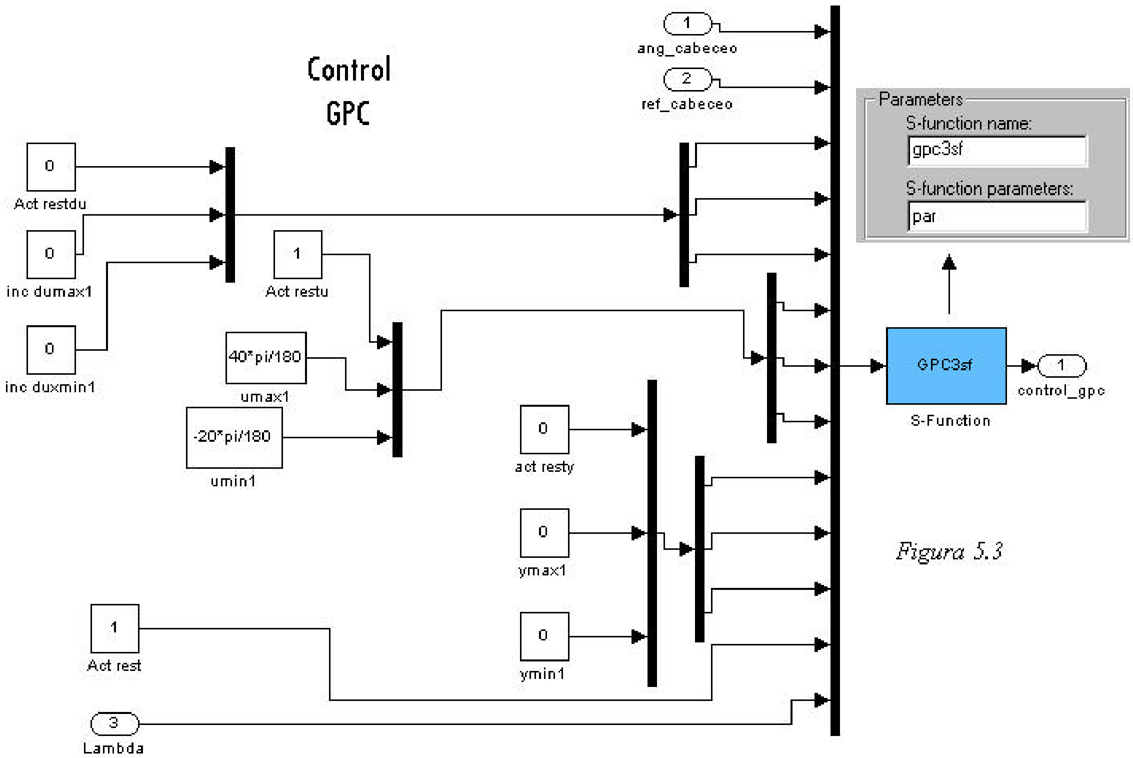


Figura 5.3

Con el objeto de disminuir en lo posible la carga computacional acarreada por el control GPC iterativo, se ha elegido un tiempo de muestreo de 0.06 s, que aun siendo mayor que los 0.008 s tomados para el control GPC analítico (ver capítulo 5) dará lugar a una correcta simulación. El retardo introducido por el piloto (0.3 s), se traduce ahora en 5 periodos de muestreo (ver figura 5.2).

En la figura 5.2 se puede observar que la señal de control (control_gpc) proporcionada por el bloque Control GPC no es directamente la señal proporcionada por los actuadores, sino que hay varios bloques intermedios. Esto hace que la restricción sobre las variables manipuladas afecte directamente a la señal control_gpc e indirectamente a la señal de los actuadores (delta_e). Para mantener el mismo nivel estático en la señal control_gpc y delta_e, se ha escalado convenientemente el filtro intermedio tras realizar una simple simulación. Con este escalado se puede asegurar que si la señal control_gpc se mantiene dentro del rango dado por la restricción [umin1,umax1] (figura 5.3), el actuador (delta_e) no saturará.

A continuación, en la figura 5.4, se muestran los resultados de simulación para tres valores distintos de ë. El horizonte de control se ha fijado a 27, y el horizonte de predicción se ha elegido igual a 50 en todos los casos. En la figura 5.4 se observa que, a medida que aumenta ë lo hacen también el tiempo de subida y la sobreoscilación de la respuesta en cabeceo. Obteniéndose para ë=0.1 la respuesta más rápida y con menor sobreoscilación. En cualquier caso el error en régimen permanente se hace prácticamente nulo. Igualmente se observa que al disminuir ë, aumenta la amplitud de la actuación en los instantes iniciales (delta_e).

En la figura 5.4 se ha incluido la representación de la señal control_gpc generada por el bloque controlador GPC. Observando esta señal para los casos con ë=0.1 y ë=1 se comprueba como efectivamente el controlador tiene en cuenta la restricción de rango, saturando en –0.35 rad (-20º). La saturación software de ésta señal junto con el filtro intermedio escalado convenientemente, impiden la saturación del timón de profundidad (ver figura 5.4).

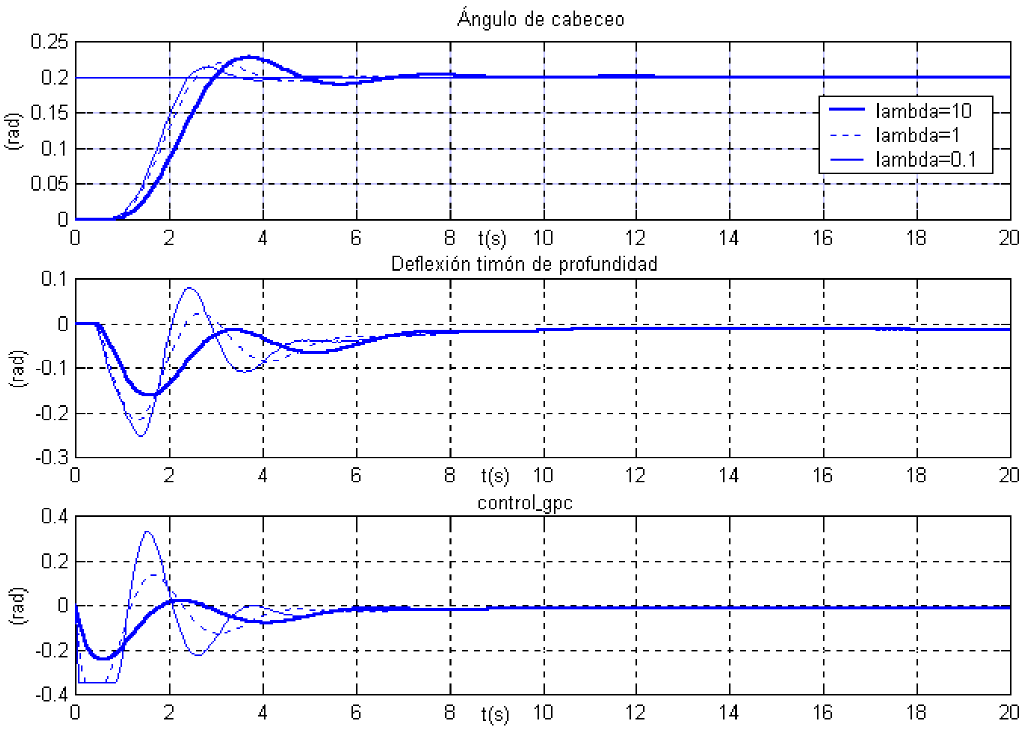


Figura 5.4

5.3 GPC ITERATIVO CON DINÁMICA LATERAL

El problema de control es el ya expuesto en el punto 3.4. Se trata de controlar el ángulo de derrape y alabeo del avión actuando sobre los alerones y el timón de dirección. Para ello se considerará el sistema completo formado por piloto, actuadores y dinámica lateral del avión.

La introducción de las dinámicas, de los parámetros del controlador, y el cálculo de la predicción se realiza con el programa en Matlab mostrado en el apéndice A.3.2. La sucesión de tareas realizadas en dicho archivo coinciden con las mostradas en la figura 5.1.

El modelo para la simulación del control GPC se muestra en las figuras 5.5 y 5.6, donde se ha tomado un tiempo de muestreo de 0.05 s, traduciéndose el retardo de 0.15 s introducido por el piloto en 3 periodos de muestreo. Los subsistemas piloto y actuadores son los mismos expuestos en los puntos 3.4.2 y 3.4.3.

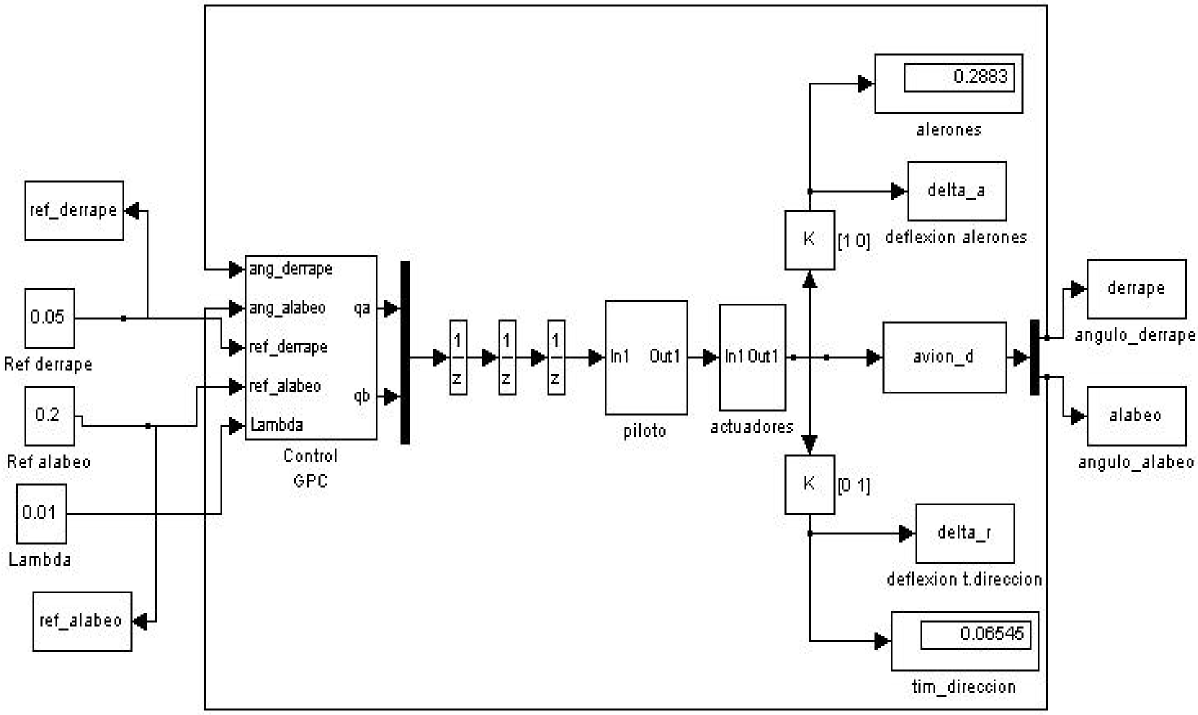


Figura 5.5

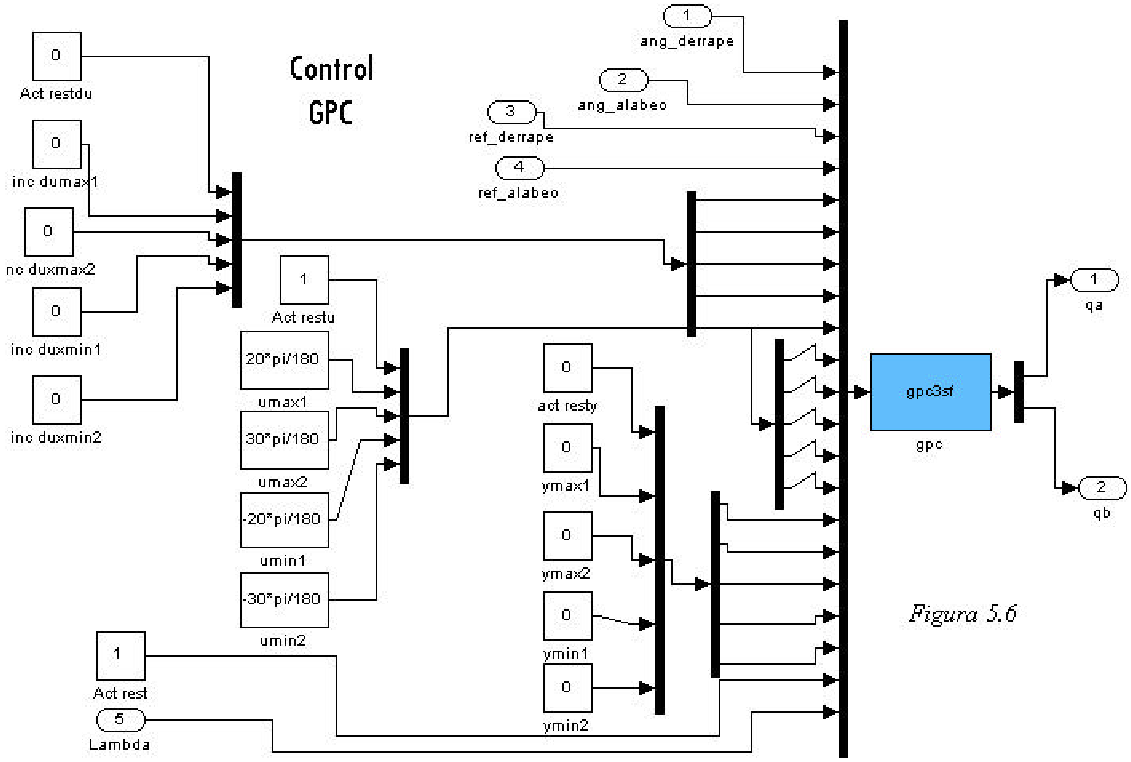


Figura 5.6

Realizando distintas simulaciones para distintos horizontes de control, horizontes de predicción y valores de lambda, no se obtiene ninguna respuesta aceptable. Sin embargo, si se elimina la restricción en el rango de los actuadores y el bloque de saturación correspondiente, se obtiene el resultado mostrado en la figura 5.7 para ë=0.01, N2=34 (4 + horizonte de predicción), N3=29 (horizonte de control). La figura 5.8 corresponde al resultado de la simulación para los mismos parámetros considerando la restricción en los actuadores.

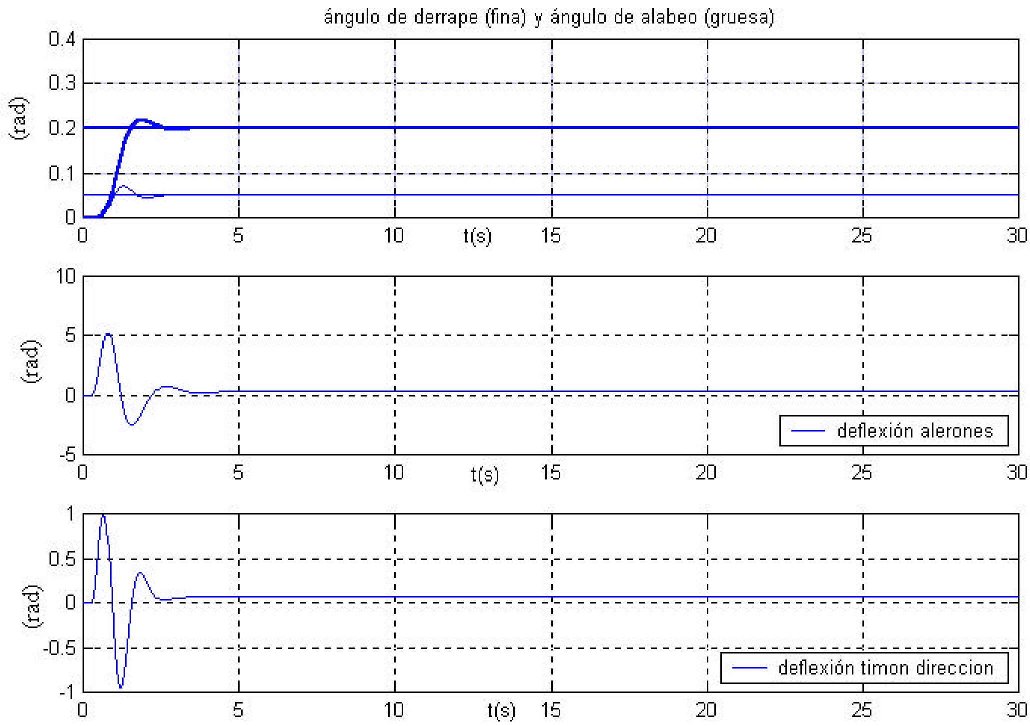


Figura 5.7

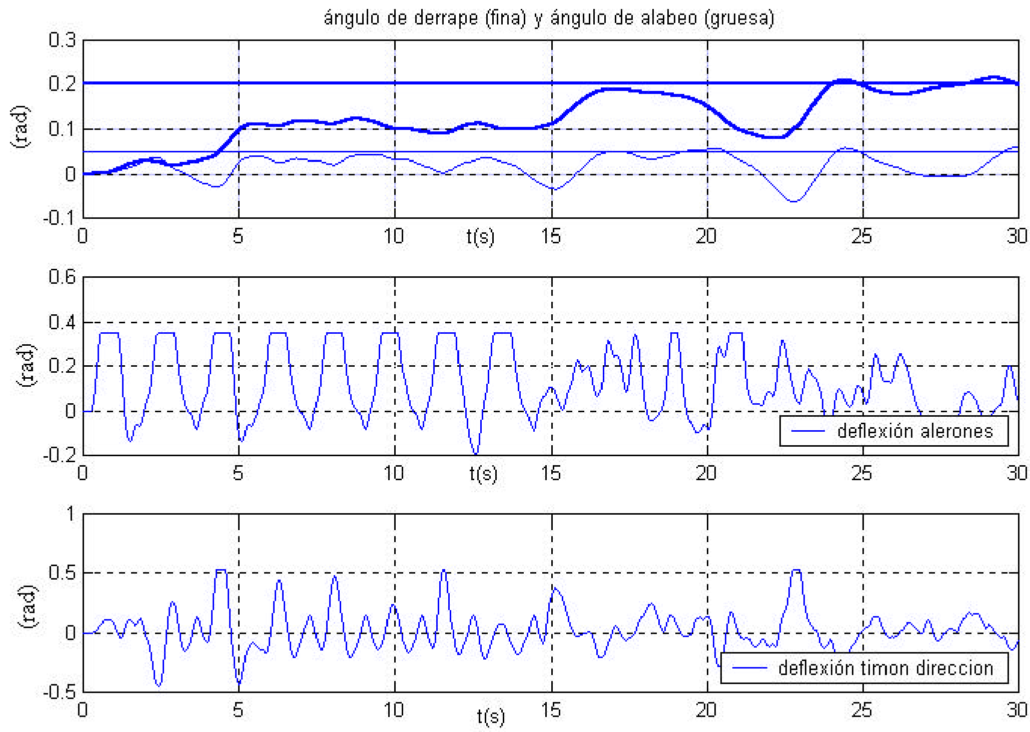


Figura 5.8

A la vista de los resultados se concluye que no se ha conseguido un control GPC MIMO adecuado para el control lateral del avión debido a la limitación de rango de los actuadores.

5.4 COMPORTAMIENTO ANTE PERTURBACIONES

El control GPC tiene la propiedad de rechazar tanto las perturbaciones instantáneas como permanentes. En este apartado se comprobará dicha propiedad sobre el control GPC iterativo de la dinámica longitudinal, resultando que han desaparecido los problemas de saturación vistos en el punto 4.4.

Para incluir las perturbaciones en Simulink basta con insertar algún sumador y bloque escalón, como se muestra en la figura 5.9.

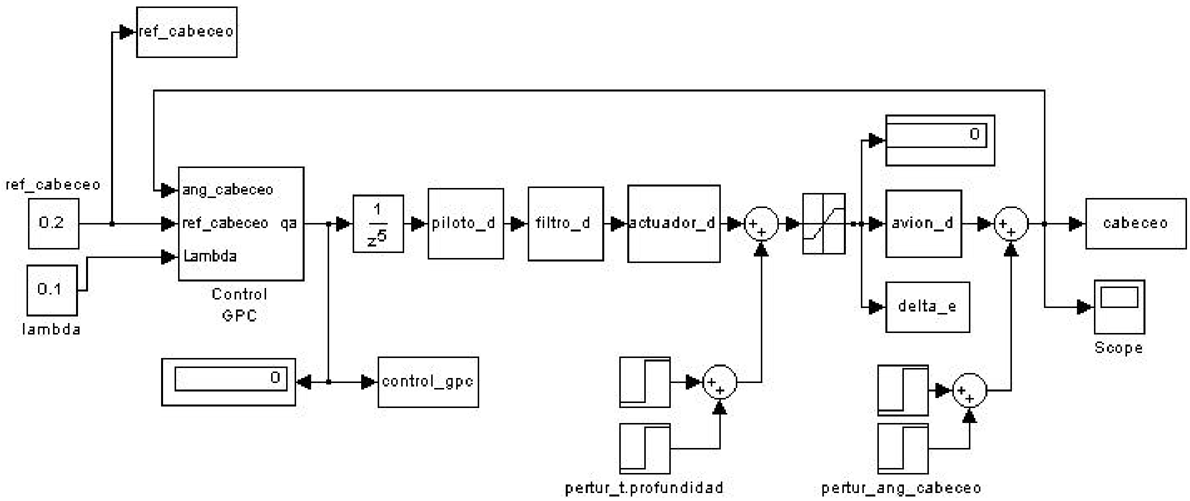


Figura 5.9

5.4.1 Perturbaciones a la salida

A continuación se introduce una perturbación instantánea (1 décima de segundo) a la salida (ángulo de cabeceo) del 50% del valor de referencia (0.2*0.5=0.1rad) a los 10 segundos. Se puede observar en la figura 5.10 que el controlador rechaza dicha perturbación correctamente.

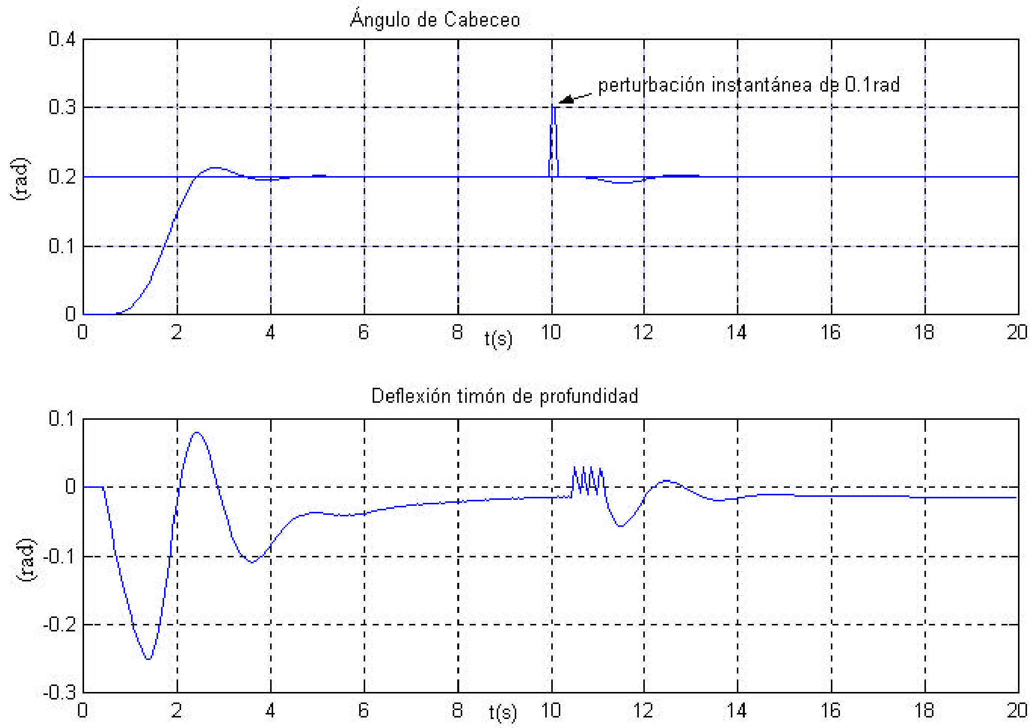


Figura 5.10

Si la misma perturbación se hace mantenida mantenida (hasta fin de simulación) el controlador GPC iterativo también la rechaza como se observa en la figura 5.11.

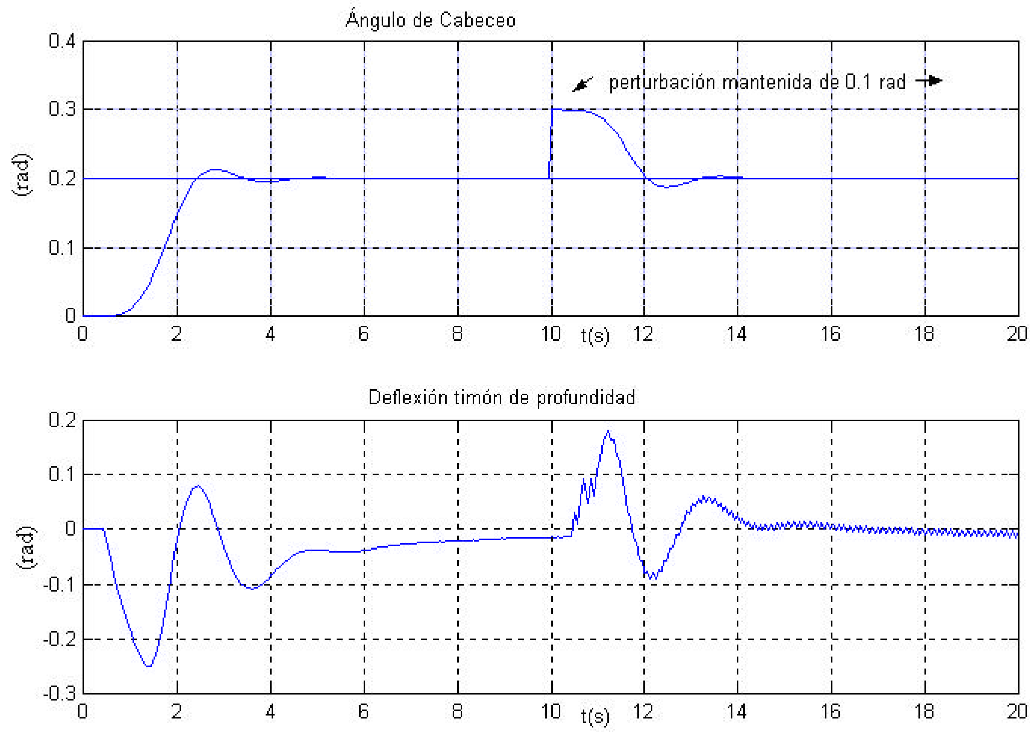


Figura 5.11

En la figura 5.11 se observa la aparición de una oscilación de alta frecuencia en la variable manipulada (catering). Para disminuir estas mini oscilaciones se puede disminuir el periodo de muestreo. La figura 5.12 muestra la respuesta del sistema con control GPC para un periodo de muestreo de 0.03seg (la mitad que en el anterior). Dicha disminución del periodo de muestreo conlleva cambios en el bloque Simulink y en el archivo mostrado en el apéndice A.3.2, pues el retardo introducido por el piloto pasa a ser 10 periodos de muestreos (en lugar de cinco).

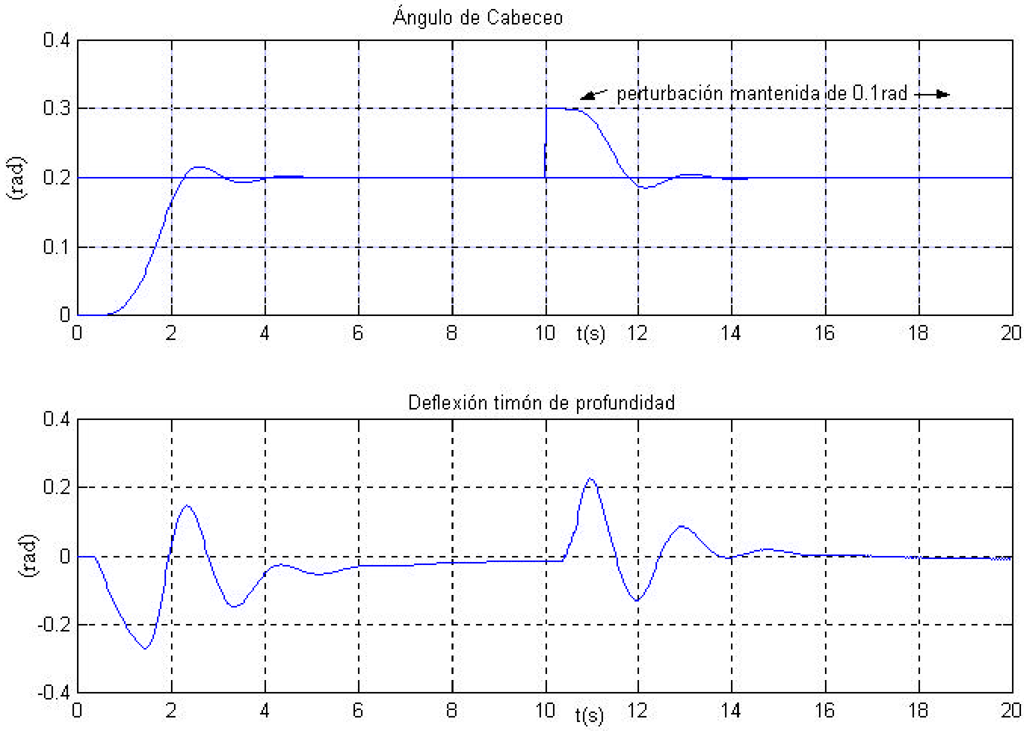


Figura 5.12

La disminución del periodo de muestreo trae consigo un aumento de la carga computacional. Otra opción para disminuir las mini oscilaciones sin aumentar excesivamente la carga computacional consiste en la introducción de unas sencillas líneas de código Matlab en el archivo GPC3sf.m mostrado en el apéndice A.3.5. El código introducido da por buena la señal de control actual (uk(1)) calculada por el controlador GPC si junto con las dos señales de control anterior (uk(2), uk(3)) dan lugar a una función creciente o decreciente. En caso contrario se deduce que existe una oscilación, y se asigna como señal de control actual la señal de control anterior (uk(2)). El código introducido en la función mdlUpdate del archivo GPC3sf.m antes de la actualización del estado es:

```
if(uk(1)>uk(2))
   if(uk(2)<uk(3))
      uk(1)=uk(2);
   end
end

if(uk(1)<uk(2))
   if(uk(2)>uk(3))
     uk(1)=uk(2);
   end
end
```

Con estas modificaciones la respuesta del sistema es la mostrada en la figura 5.13. Obviamente esta técnica se puede usar junto a una disminución del periodo de muestreo mejorando los resultados.

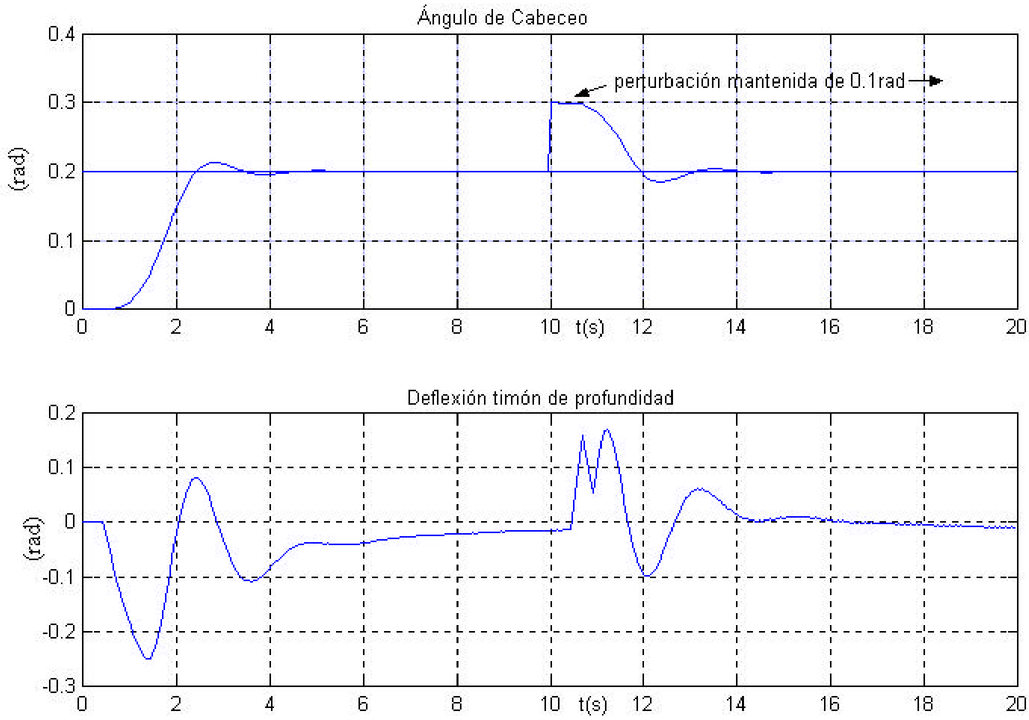


Figura 5.13

5.4.2 Perturbaciones a la entrada

Si se introduce una perturbación de 0.1 rad instantánea (de t = 10 a t= 10.1 seg) en la deflexión del timón de profundidad se puede observar en la figura 5.14 que el sistema se recupera rápidamente. De hecho esta recuperación ocurre incluso si la amplitud de la perturbación es tan grande que satura el actuador, como se observa en la figura 5.15 donde la perturbación introducida es de 1 rad.

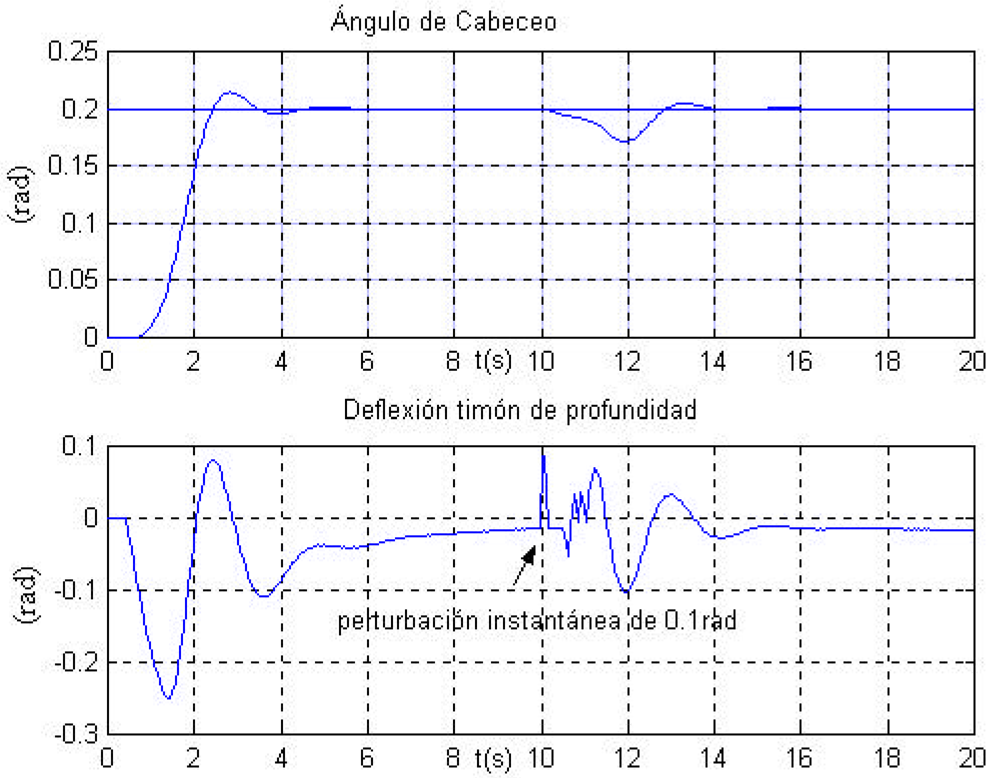


Figura 5.14

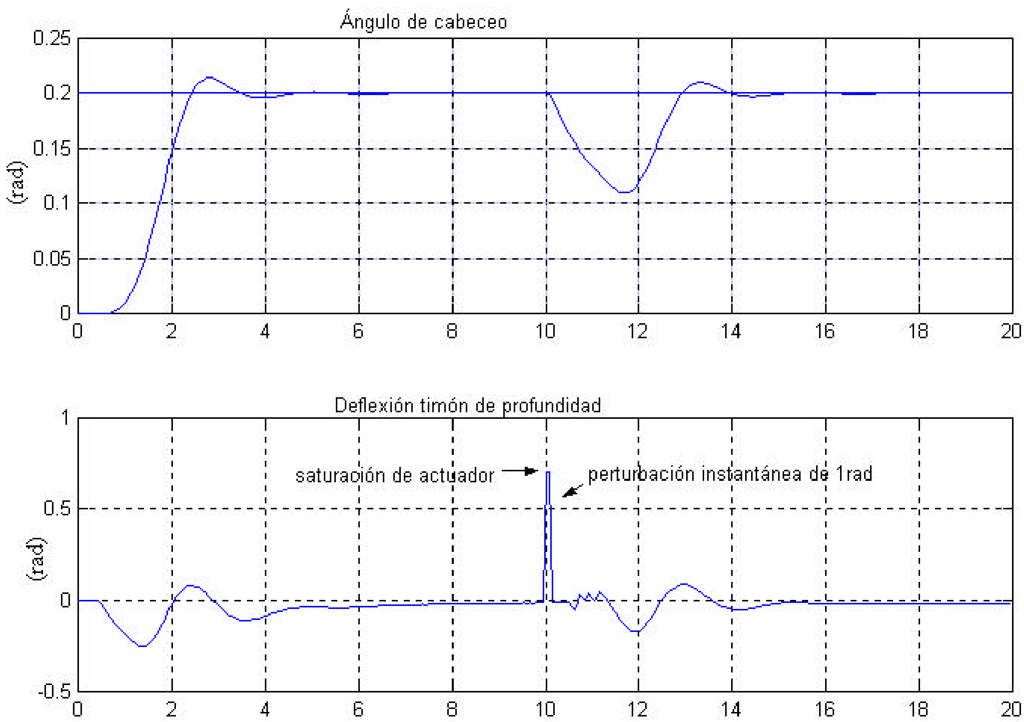


Figura 5.15

Si la perturbación a la entrada de 0.1 rad se hace mantenida el controlador también la rechaza, aunque en este caso se produce una mayor oscilación en el ángulo de cabeceo (ver figura 5.16).

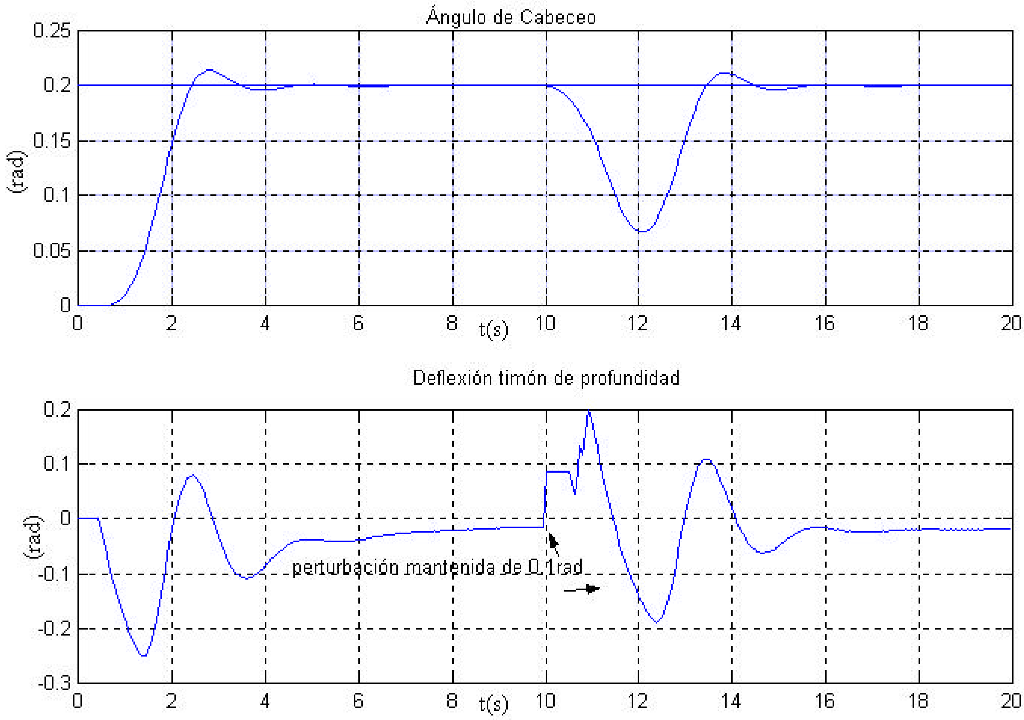

Ángulo de Cabeceo
(rad)
t(s)
Deflexión timón de profundidad
(rad)
perturbación mantenida de 0.1rad
t(s)

figura 5.16

Capítulo 6

Control PID

6.1 DEFINICIÓN Y AJUSTE CONTROL PID EN TIEMPO DISCRETO

La ecuación general en tiempo continuo de un control PID viene dada por:

$$u(t) = Kc\left[e(t) + \frac{1}{Ti}\int_0^t e(t)dt + Td\,\frac{de(t)}{dt}\right]$$

Discretizando esta ecuación para un tiempo de muestreo T, se obtiene la siguiente ley de control de un controlador PID discreto [22]:

$$\mathrm{u(k) = u(k-1) + Kp(y(k-1) - y(k)) + KI(r(k) - y(k)) + KD(2y(k-1) - y(k-2) - y(k))}$$

donde (6.1)

$$\mathrm{Kp = Kc - 0.5KI, \quad KI = \frac{Kc}{Ti}T, \quad KD = \frac{KcTd}{T}}$$

Takahashi, Clan y Auslander proponen un conjunto de reglas [23], que utilizan los dos métodos propuestos por Ziegler y Nichols para el controlador PID en continuo, a fin de determinar valores aceptables para Kp, K_I y K_D. La siguiente tabla muestra los valores propuestos por Takahashi y otros para el ajuste de los parámetros del controlador PID discreto por el método de las oscilaciones sostenidas:

Control	Kp	K_I	K_D
PID	0.6Ku-0.5KI	$\frac{6KuT}{5Tu}$	$\frac{3KuTu}{40T}$

Donde Ku es la ganancia de un controlador proporcional que lleva el proceso al límite de su estabilidad, y Tu es el período de la oscilación resultante.

6.2 APLICACIÓN A LA DINÁMICA LONGITUDINAL

6.2.1 Especificaciones de control

En este punto se partirá de la dinámica longitudinal del avión expresada en la ecuación (2.37) o su equivalente en función de transferencia dada en (2.44). Una vez añadidas las dinámicas del piloto y actuador se realizará un control PID sobre el ángulo de cabeceo (pitch) del avión, actuando sobre el timón de profundidad.

Como punto de partida las especificaciones sobre la respuesta en cabeceo del avión deseada serán las expuestas ya en otros capítulos :

Referencia cabeceo 0.2 rad (11.46º)
Sobreoscilación < 10%
Tiempo subida < 2 seg
Error en permanente < 2%
Tiempo establecimiento < 10 seg.

Dichos valores se han tomado del Control Tutorial for Matlab (The University of Michigan) [10].

6.2.2 Resultados del control PID

Para calcular la ganancia crítica Ku y el periodo de oscilación Tu es necesario definir previamente el sistema. Debido al efecto oscilante de la acción del piloto se introduce un filtro paso de baja en el sistema total para conseguir una señal de control adecuada. En el apéndice A.4.1 se muestra el archivo en Matlab con la definición de los elementos del sistema original junto con el filtro introducido.

Una vez definido el sistema es fácil determinar Ku y Tu mediante el uso de la herramienta Simulink, resultando:

Ku = -1.5555, Tu = 4.5 s.

Haciendo uso de las ecuaciones (6.1) se obtienen los siguientes valores:

Kp = -0.9316408, KI = -0.0033184, KD = -65.62265625

El esquema en Simulink resultante con el controlador PID se muestra en la figura 6.1.

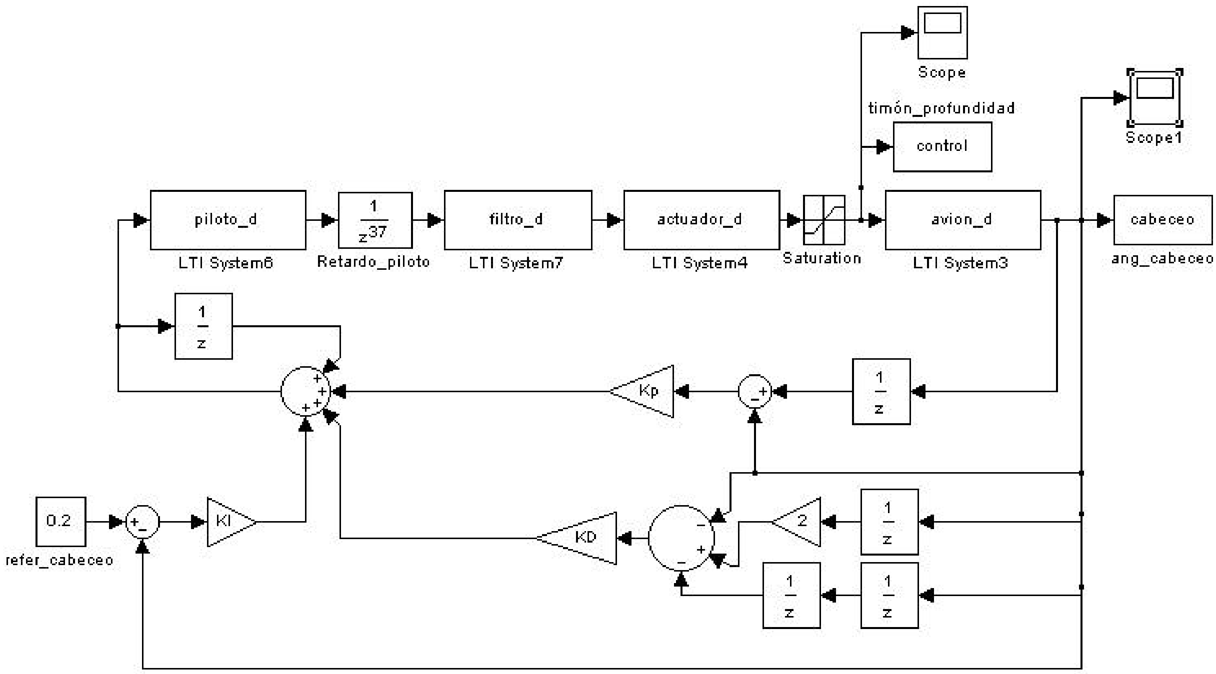


Figura 6.1

La figura 6.2 muestra el resultado de simulación, que no se ajusta a las especificaciones.

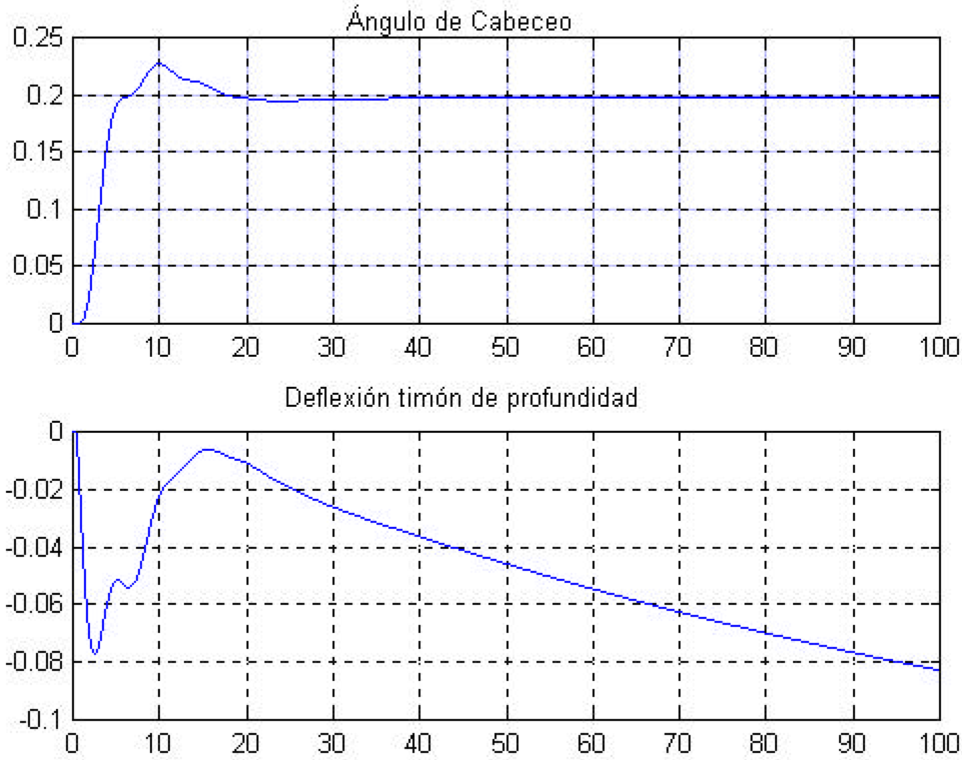


Figura 6.2

Se puede intentar ajustar la respuesta del sistema modificando mediante el ajuste fino de los parámetros del controlador. Sin embargo esta tarea resultó infructuosa, pues la respuesta con la sintonización de partida no cumple ni la especificación de sobreoscilación ni las especificaciones temporales. Por lo que un aumento en la constante KI para reducir el tiempo de estabilización aumenta la sobreoscilación, mientras que un aumento de KD para estabilizar, ralentiza el sistema. La figura 6.3 muestra como ejemplo el resultado de la simulación con los valores de KD y KI duplicados.

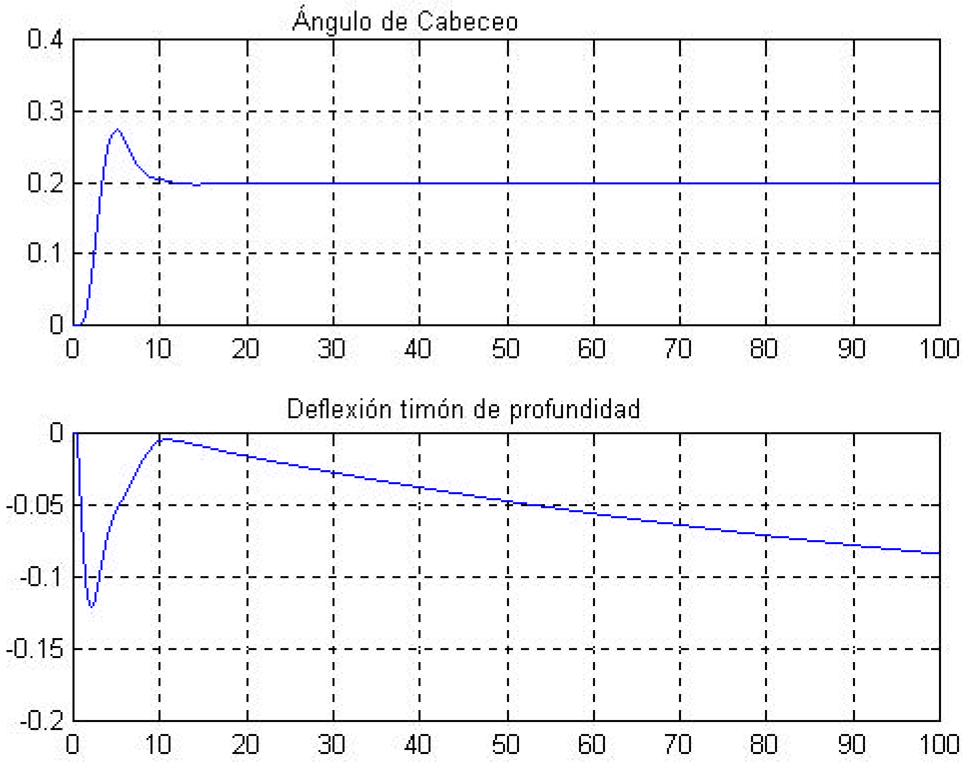


Figura 6.3

6.3 APLICACIÓN A LA DINÁMICA LATERAL

6.3.1 Especificaciones de control

En este punto se partirá de la dinámica lateral del avión expresada en la ecuación (2.38) para realizar un control PID sobre el ángulo de derrape (sideslip, â) y el ángulo de alabeo (roll, ö) del avión actuando sobre los alerones y el timón de dirección.

En primer lugar se realizará un estudio del acoplamiento del sistema mediante el cálculo de la matriz de ganancias relativas (MGR). Posteriormente se intentará ajustar el controlador para la mejor asignación de pares entrada-salida.

Se intentará seguir una referencia de 0.05 rad (3º) para el ángulo de derrape, y de 0.2 rad (13º) para el ángulo de alabeo.

6.3.2 Estudio del acoplamiento del sistema

El sistema a tratar presenta dos salidas (ángulo de alabeo y ángulo de derrape) y dos entradas o actuaciones (deflexión alerones y deflexión timón de dirección), lo que implica cuatro funciones de transferencia. Mediante al cálculo de la MGR podría ser posible determinar una buena asociación entrada-salida. El el apéndice A.4.2 se muestra el archivo en Matlab que contiene el código para el cálculo de la MGR del avión comercial estudiado. La sucesión de tareas realizadas en dicho archivo se muestran en la figura 6.4.

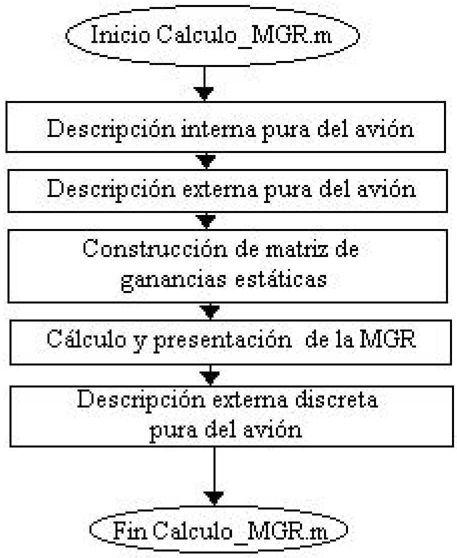


Figura 6.4

La ejecución de este archivo proporciona la siguiente MGR:

$$MGR = \begin{bmatrix} -1.04393234345017 & 2.04393234345017 \\ 2.04393234345017 & -1.04393234345017 \end{bmatrix}$$

Atendiendo al resultado se deben descartar los emparejamientos entrada-salida cuyo coeficiente en la MGR sea negativo. Por lo cual los emparejamientos correctos serán aquellos con coeficientes positivos y valores cercanos a uno. Por tanto se puede dar por bueno el control del ángulo de alabeo (salida 2) actuando sobre los alerones (entrada 1), pues el coeficiente correspondiente a la posición (2,1) en la MGR toma valor 2.04. Por su parte el control del ángulo de derrape (salida 1) se realizará actuando sobre el timón de dirección (entrada 1).

Con estas asignaciones entrada-salida, la dinámica para el control del ángulo de alabeo corresponde al sistema descrito en el apéndice A.4.2 como sistema2d (función de transferencia de alerones a alabeo), mientras que la dinámica para el control del ángulo de derrape corresponderá al sistema3d (función de transferencia de timón dirección a derrape).

6.3.3 Ajuste del control PID

En primer lugar es necesario definir las dinámicas de los actuadores y el piloto mediante unas sencillas líneas de código en Matlab (como se hizo en el punto 3.4 para el control LQR):

```
%Definición de la función de transferencia de los actuadores. Habrá que
%añadir saturación en modelo Simulink
Tm=0.008;
sisc=tf([1],[0.0169 0.121 1]);        %f.t actuadores en t.continuo
actuador_d=c2d(sisc,Tm);              %f.t actuadores en t.discreto

%Definición función transferencia piloto
Tp=0.15;          %retardo introducido por piloto en segundos
s=tf('s');
aux=1.5/(0.13*s+1);
z=tf('z',Tm);
rp=Tp/Tm-mod(Tp/Tm,1);  %retardo en tiempos de muestreo (parte entera)
piloto=c2d(aux,Tm,'tustin')*z^(-rp);
```

Aprovechando los resultados del apartado 6.3.2 podemos establecer el esquema de control de la figura 6.5.

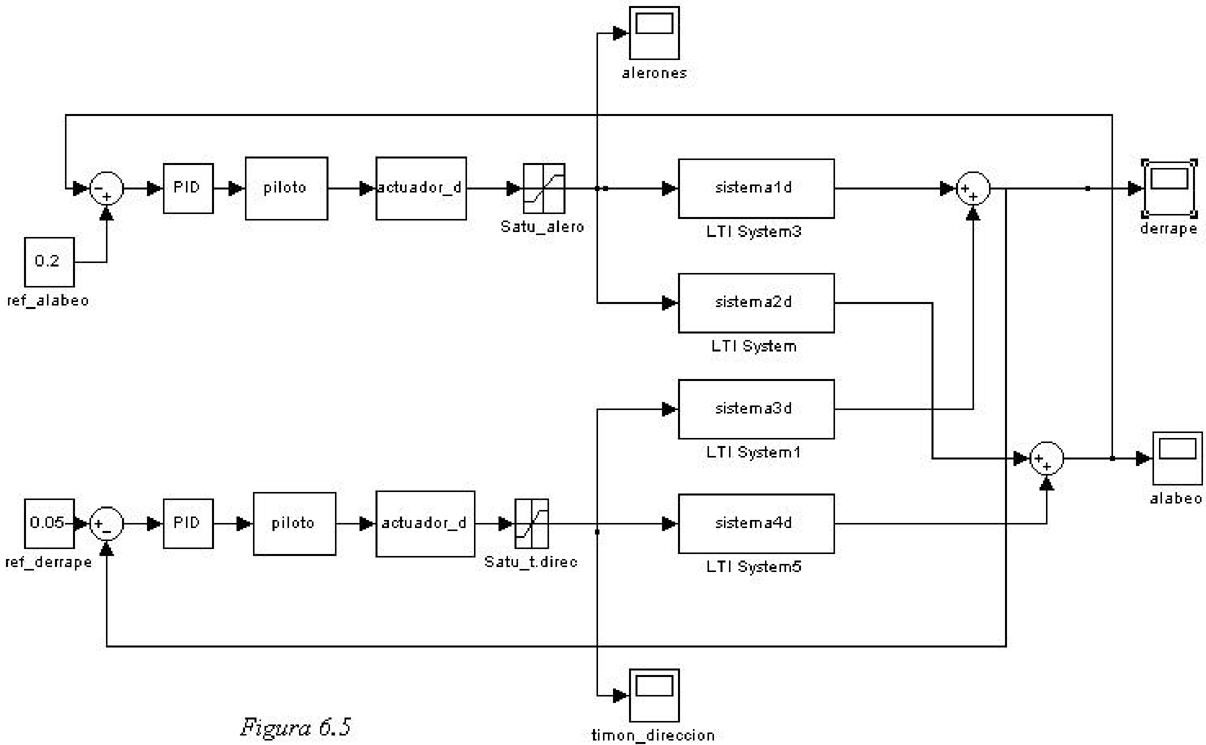


Figura 6.5

La manera de proceder para el ajuste de los controladores PID es sencilla. Para ajustar el control PID sobre el alabeo se anula la acción del bloque PID inferior (ganancia nula) y se ajusta el bloque PID superior por el método de las oscilaciones sostenidas, como ya se hizo con la dinámica longitudinal en el punto 6.2. Para el ajuste del PID sobre el derrape se anula la acción del bloque PID superior y se ajusta el bloque PID inferior de la misma manera. Sin embargo este intento resulta infructuoso, pues ante cualquier ganancia el sistema realimentado se presenta inestable. Esta inestabilidad es fácil de observar en el lugar de las raíces. Para obtener el lugar de las raíces correspondiente al control del derrape se introduce la siguiente línea en Matlab:

rltool(piloto*actuador_d*sistema3d)

El lugar de las raíces obtenido es el mostrado en la figura 6.6, donde se dibujan con cuadrados los polos del bucle cerrado para una ganancia unidad. La zona cercana a z = 1 se muestra ampliada en la figura 6.7 y 6.8. Se puede observar que hay una rama en el eje real fuera del círculo de radio unidad, lo que corresponde a un polo en bucle cerrado inestable. En la figura 6.6 también se observan dos polos complejos conjugados inestables, aunque estos polos pueden entrar en el círculo unidad si se disminuye la ganancia por debajo de la unidad.

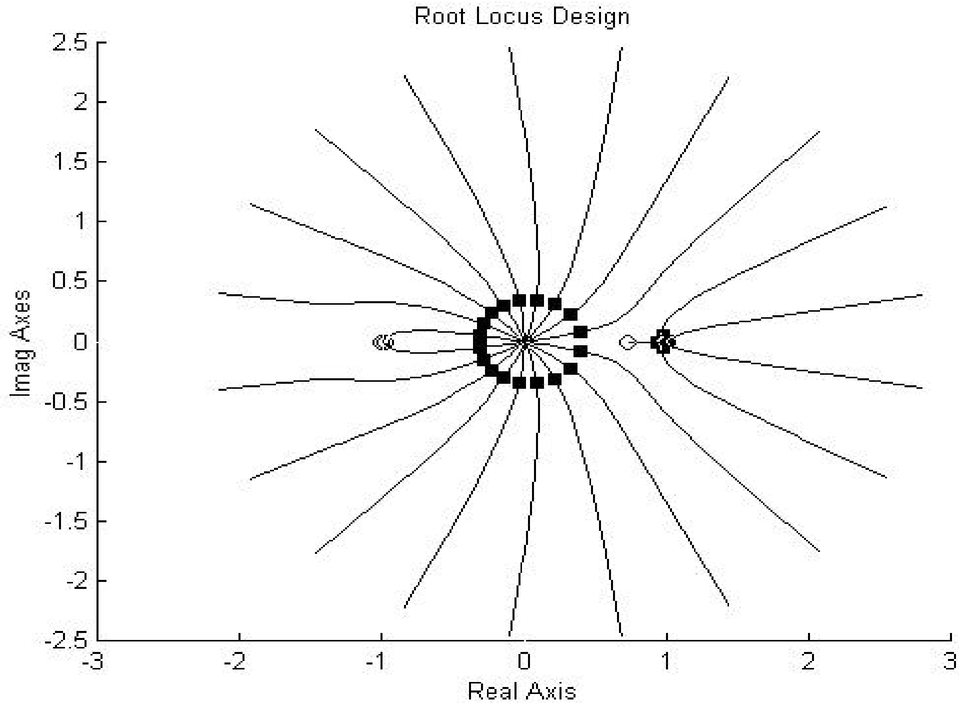


Figura 6.6

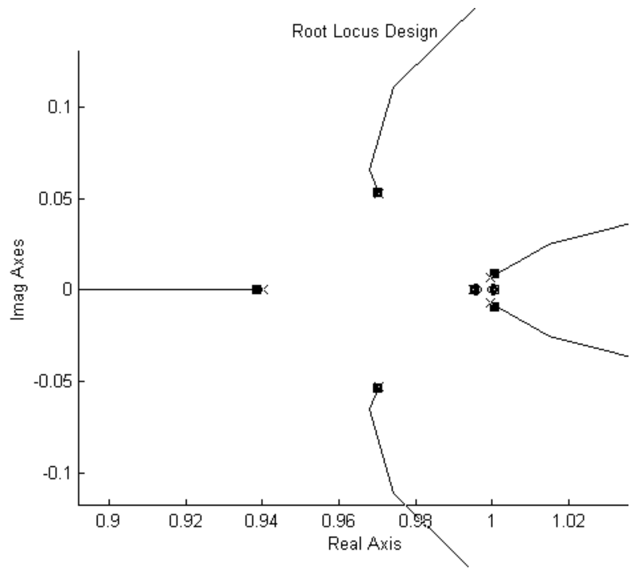


Figura 6.7

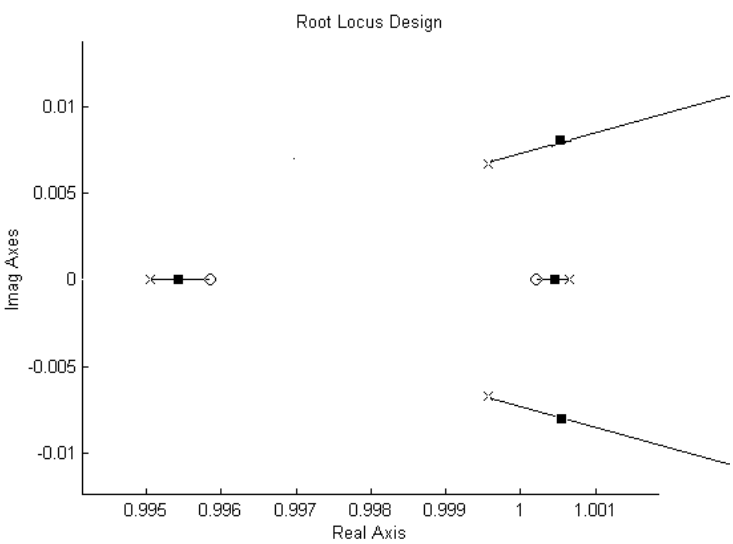


Figura 6.8

Puesto que no es posible la sintonización del PID usando el método de las oscilaciones sostenidas, podría intentarse realizar un diseño del PID usando el lugar de las raíces. Sin embargo si se considera el PID discreto como un par de polos en z = 0 y z =1 y un par de ceros (reales o complejos conjugados) a colocar, no es posible encontrar una posición de los ceros tal que estabilice el sistema.

6.4 CONCLUSIONES DEL CONTROL PID

La estructura de control PID es universalmente utilizada en la industria. Hoy en dia, a pesar de la abundancia de sofisticadas herramientas y métodos avanzados de control, el controlador PID es aún el más ampliamente utilizado en la industria moderna, siendo extremadamente beneficiosos en el control de muchas aplicaciones. Sin embargo en el caso del sistema que se trata en este proyecto no ha sido posible conseguir un control PID adecuado (ni para el caso longitudinal ni para el transversal). Esto no implica que el control PID no sea posible para aplicaciones aeronáuticas (o en su defecto la náutica), sino que no ha sido posible para un sistema concreto con unos parámetros de vuelo y especificaciones concretas. Un ejemplo de aplicación del control PID del ángulo de cabeceo de un avión sin considerar las dinámicas de actuador y piloto, se puede encontrar en el Control Tutorials for Matlab (CTM) [10].

Capítulo 7

Fase de aterrizaje de una aeronave

7.1 MODELO DE UNA AERONAVE EN FASE DE ATERRIZAJE

En este proyecto se va a utilizar un modelo con dos grados de libertad de una aeronave que solo puede moverse en los ejes longitudinal y vertical. Las señales de control del sistema serán la deflexión del timón de profundidad (elevator) y el empuje o estrangulador de gas (throttle valve). Las ecuaciones obtenidas en la parte I pueden extenderse para el aterrizaje considerando el empuje como una nueva entrada, obteniéndose el conjunto de ecuaciones 7.1 [24]:

$$\left.\begin{aligned} u' &= Xu\,u + Xw\,w + Xq\,q - g(\pi/180)\theta\cos\gamma_0 + X_{\delta e}\delta_e + X_{\delta T}\delta_T \\ w' &= Zu\,u + Zw\,w + (Zq - (\pi/180)U_0)q + g(\pi/180)\theta\,\mathrm{sen}\gamma_0 + Z_{\delta e}\delta_e + Z_{\delta T}\delta_T \\ q' &= Mu\,u + Mw\,w + Mq\,q + M_{\delta e}\delta_e + M_{\delta T}\delta_T \\ \theta' &= q \end{aligned}\right\} \quad (7.1)$$

donde las variables y constantes tienen el siguiente significado físico:

u: incremento de la velocidad longitudinal (pies/s)
w: incremento de la velocidad vertical (pies/s)
q: velocidad angular de cabeceo (grado/s)
θ: ángulo de cabeceo (grados)
δ_e: deflexión timón de profundidad (grados)
δ_T: gas
U_0: velocidad longitudinal estacionaria (pies/s)
γ_0: ángulo de trayectoria (grados)
g: aceleración de la gravedad (32.2 pies/s^2)
X_*, Z_*, M_*: derivadas de estabilidad y control

Los cálculos realizados para la obtención de la dinámica de una aeronave en fase crucero en el capítulo 2 del presente proyecto son válidos para la fase de aterrizaje (ecuaciones (2.1) a (2.36). Eso sí, es necesario modificar los valores de los parámetros del modelo de acuerdo al punto de operación del sistema.

Para completar el modelo es necesario introducir como nueva variable la altura de la aeronave h (pies). Para ello se formula la siguiente ecuación:

$$h' = (u + U_0)\,\mathrm{sen}q - w\cos q \qquad (7.2)$$

Para obtener un modelo lineal es necesario realizar una aproximación a (7.2). Si se considera è 0 y u 0, la ecuación (7.2) se puede escribir como:

$$h' = U_0 q\,(P\ /180) - w \qquad (7.3)$$

Las ecuaciones (7.1) y (7.3) proporcionan la descripción interna en el espacio de estados de un sistema con vector de estados $\mathbf{X} = \left[u\ w\ q\ q\ \ h\right]^T$, y vector de actuaciones **u**=[$ä_e$ $ä_T$], pudiéndose describir las ecuaciones de transición de estados según:

$$\mathbf{X}' = \begin{bmatrix} Xu & Xw & Xq & -g(P\ /180)\cos g_0 & 0 \\ Zu & Zw & (Zq-(P\ /180)U_0) & +g(P\ /180)\,\mathrm{sen}g_0 & 0 \\ Mu & Mw & Mq & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & U_0(P\ /180) & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} X_{de} & X_{dT} \\ Z_{de} & Z_{dT} \\ M_{de} & M_{dT} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} .\mathbf{u} \qquad (7.4)$$

Tomando los siguientes valores típicos para la fase de aterrizaje de una aeronave [25]:,

Xu=-0.038	Zu=0.313	Mu=-0.0211	
Xw=-0.0513	Zw=-0.605	Mw=0.157	
Xq=0.00152	Zq=-0.0410	Mq=-0.612	(7.5)
$X_{äe}$=0.00005	$Z_{äe}$=-0.146	$M_{äe}$=0.459	
$X_{äT}$=0.158	$Z_{äT}$=0.031	$M_{äT}$=0.0543	
g_0=-3º	U_0=235(pies/s)		

se obtiene el sistema dado por las siguientes ecuaciones:

$$\mathbf{X}' = \begin{bmatrix} -0.0380 & -0.0513 & 0.00152 & -0.5612 & 0 \\ 0.3130 & -0.6050 & -4.1425 & -0.0294 & 0 \\ -0.0211 & 0.157 & -0.612 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 4.1015 & 0 \end{bmatrix} .\mathbf{X} + \begin{bmatrix} 0.00005 & 0.158 \\ -0.146 & 0.031 \\ 0.459 & 0.0543 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} .\mathbf{u} \qquad (7.6)$$

Finalmente, será útil en los siguientes capítulos la ecuación (7.7), en la que x' representa la derivada temporal de la distancia horizontal recorrida por la aeronave.

$$x' = (u + U_0)\cos q + w\,\mathrm{sen}q \qquad (7.7)$$

De nuevo se puede realizar la aproximación è 0 y u 0, transformándose (7.7) en:

$$x' = U_0 + wq\,(P\ /180) \qquad (7.8)$$

7.2 PROCESO DE ATERRIZAJE

El proceso de aterrizaje se suele dividir en varias fases de forma arbitraria según los autores. Esencialmente el aterrizaje puede dividirse en 2 fases: fase de planeo (glide), y fase de recogida (flare).

En la primera fase (fase de planeo) el avión se encuentra inicialmente a una altura h(0) de unos 500 pies (152.4 metros) sobre el suelo, y debe descender siguiendo una trayectoria rectilínea conocida como GSC (Glide Slope Centerline), marcada por una estación transmisora situada en tierra (ver figura 7.1). Este descenso se realiza con una velocidad aproximada de 10 pies/s (3.048 m/s), manteniendo el ángulo de cabeceo entre –5 y +5 grados.

La fase de recogida comienza cuando la aeronave se encuentra de 20 a 70 pies sobre el suelo (de 6 a 21 m). La velocidad de descenso debe disminuirse considerablemente, alcanzando un valor aproximado de 2 pies/s, que permita al tren de aterrizaje disipar la energía del impacto al tomar tierra.

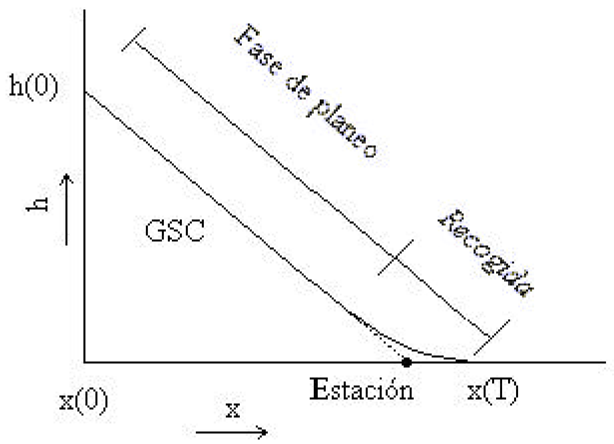


Figura 7.1

Para que el aterrizaje sea considerado un éxito debe respetar ciertas condiciones. Previamente es necesario definir ciertas variables:

- Temporalmente se considerará t=0 el instante de inicio de la fase de planeo, y t=T el instante en que la aeronave toma contacto con el suelo (ver figura 7.1)

- Se definirá x (downrange) como el desplazamiento horizontal de la aeronave desde el comienzo de la fase de planeo hasta la toma de contacto con tierra. Si se consideran los valores de los parámetros (7.5) de la sección anterior ($\tilde{a}_0$=-3º) y una altura inicial h(0)=500 pies, el valor inicial para x vendrá dado por:

$$x(0) = \frac{h(0)}{\tan(\gamma_0)} = -9540.6 \text{ pies} \qquad (7.9)$$

Con estas definiciones previas, es posible establecer las siguientes condiciones para que un aterrizaje pueda ser considerado exitoso [26]:

(7.10)

–3 h'(T) 0 pies/s. La velocidad de descenso en el momento de la toma en tierra (touchdown) debe ser inferior a 3 pies/s.

-300 x(T) 1000 pies. Limita el desplazamiento longitudinal máximo y mínimo del avión durante el proceso completo de aterrizaje. Para los datos de la aeronave del capítulo anterior (x(0)=-9540.6 pies), el desplazamiento mínimo será 9240.6 pies (9540.6-300) y el desplazamiento máximo será 10540.6 pies (1000+9540.6).

200 x'(T) 270 pies/s. Limite de la velocidad horizontal en la toma en tierra.

-10 è(T) 5 grados. El ángulo de cabeceo del avión en el momento de tomar contacto con el suelo debe estar en el rango [-10,+5] grados.

-5 è(t) +5 grados. El ángulo de cabeceo del avión durante la fase de planeo debe estar en el rango [-5,+5] grados.

Para conseguir cumplir estas condiciones se puede manipular el timón de profundidad y el empuje mediante el control de combustible (throttle). Normalmente los sistemas de aterrizaje automático (ALS, Automatic Landing System) usan el empuje para mantener constante la velocidad de la aeronave, dejando al timón de profundidad la misión de seguir la trayectoria de descenso amortiguando todas las posibles oscilaciones. Aunque también es posible encontrar ALSs que realizan el control de la aeronave exclusivamente por propulsión, capaces incluso de controlar el movimiento direccional de la aeronave haciendo uso de un empuje diferencial (motores izquierda y derecha con distinta potencia de propulsión) [27].

7.3 OTROS FACTORES EN EL ATERRIZAJE

De las fases del vuelo, el despegue y el aterrizaje son críticos, dado que, al fin y al cabo, son los momentos en los que el aparato está más cerca del suelo. En el aterrizaje, el avión no hace casi uso de sus motores, y depende de la sustentación para que el avión se acerque a la pista con una tasa de descenso adecuada y a una velocidad lo suficientemente alta como para tener control del avión y que éste no entre en pérdida, y lo suficientemente baja como para no utilizar demasiada longitud de pista o reventar los neumáticos del tren de aterrizaje. Para facilitar el despegue y aterrizaje, lo importante es conseguir que la velocidad del aire que entra en las alas del avión sea lo más alta posible, ya que es esta velocidad la que determina la sustentación producida por las mismas. Lo ideal es conseguir que con una mínima velocidad de avance sobre el suelo tengamos una máxima velocidad del aire.

La velocidad y dirección del viento es pues, un factor importante a la hora de realizar una maniobra de aterrizaje. Sin embargo es frecuente la presencia de turbulencias, entendiendo éstas como el cambio de dirección y/o velocidad del viento en tramos de vuelo extremadamente cortos; estos flujos irregulares producen sobre las aeronaves cambios repentinos en la trayectoria y en la sustentación. Según su origen las turbulencias se pueden dividir en:

- Turbulencia Mecánica: debida a los rozamientos del aire con las irregularidades del terreno; predomina, por tanto, en las capas inferiores.

- Turbulencia Térmica*:* debida a una inestabilidad térmica del aire. Suele predominar en las altitudes medias, a excepción de la CAT (turbulencia en aire claro) que es frecuente en la alta troposfera y la baja estratosfera.

- Efecto suelo: con el avión volando a una distancia del suelo igual o inferior a la envergadura del ala, se produce un incremento de eficiencia del ala que mantiene al avión en el aire a velocidades más bajas de las normales. La cercanía del suelo afecta a la distribución y circulación del flujo de aire a lo largo del ala, resultando una disminución de la resistencia inducida. Cuanto más cerca del suelo esté el ala, mayor será la intensidad de este fenómeno, o lo que es lo mismo: a medida que el avión se separa del suelo el efecto suelo disminuye.

Los sistemas de aterrizaje automático (ALS) convencionales pueden proporcionar un aterrizaje suave. Sin embargo, estos sistemas están diseñados para trabajar dentro de un marco reducido de condiciones externas favorables. Si las condiciones de vuelo no se encuentran dentro de este marco, el ALS queda deshabilitado y el piloto debe hacerse cargo de la situación. Esta situación requiere una gran experiencia y esfuerzo por parte del piloto, y aún así la conmutación de control automático (ALS) a manual (piloto) puede provocar accidentes debido a la diferencias entre la ley de control aplicada por ambos. Es deseable el diseño de ALS que amplíen el marco de condiciones para un aterrizaje seguro.

En los últimos años se han desarrollado muchos ALS con el objetivo de ampliar el marco de condiciones aptas para el aterrizaje. Todos ellos tratan de adquirir la destreza de un piloto humano en situaciones adversas. Con este fin se han encontrado soluciones que hacen uso de la aplicación de redes neuronales a los ALS basándose en el modelo inverso de una aeronave [26]. En [25] se presenta una técnica de adaptación de ganancia para mejorar la robustez, en la cual se hace uso de un mapeo entrada/salida para adquirir y almacenar un modelo no lineal de piloto.

En los dos capítulos que siguen se desarrollarán dos sistemas de control automático del aterrizaje de una aeronave en condiciones ideales. El primero de ellos usará la técnica de realimentación del vector de estados por optimización LQR, y el segundo empleará el control robusto H con el cual se consigue una mayor robustez ante posibles perturbaciones.

Capítulo 8

Control LQR en aterrizaje

8.1 INTRODUCCIÓN

En el presente capítulo se realizará un control LQR para el aterrizaje de la aeronave cuya dinámica se ha modelado con el sistema de ecuaciones de transición de estados (7.6). Las condiciones iniciales previas a la maniobra de aterrizaje serán: h(0)=500 pies, x(0)=-9540.6 pies (7.9), ã$_0$=-3º, è$_0$=u_0=w_0=q_0=0. Para que la maniobra de aterrizaje sea considerada satisfactoria habrán de cumplirse las condiciones descritas en el capítulo 7, ecuación (7.10). Además el ángulo de cabeceo deberá estar en todo momento en el rango [-5,+5] grados.

Puesto que estamos ante un problema de seguimiento de una trayectoria no nula (ni constante), será necesario crear una función de escalado de la referencia para el correcto seguimiento (al igual que se hizo en el capítulo 3).

El esquema de control LQR del sistema se muestra en la figura 8.1. Se observa la presencia de dos señales de actuación (delta_e o deflexión del timón de profundidad, y empuje), de una matriz de realimentación de estados K_{LQR}, y de una matriz de escalado de la referencia $K_{escalado}$. En las siguientes secciones se verá como se generan la trayectoria de referencia y su matriz de escalado. Las variables a controlar serán u y h. Se pretende que la altura h siga una trayectoria de descenso específica manteniéndose la velocidad longitudinal constante. Esto último implica incremento de velocidad longitudinal nulo (u=0).

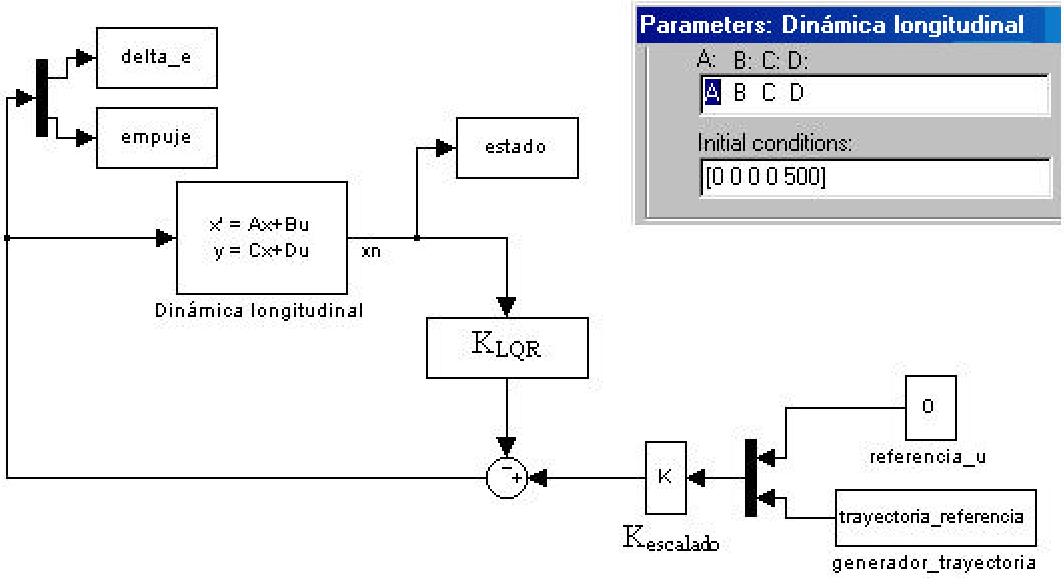


Figura 8.1

Para la simulación del modelo de la figura 8.1 es necesario ejecutar previamente el archivo principal.m incluido en el apéndice A.5.1. La sucesión de tareas realizadas en este archivo se muestran en la figura 8.2. Una de las tareas consiste en la petición del instante de conmutación entre la fase de planeo y la de recogida desde el inicio de la fase de aterrizaje, se verá en secciones posteriores la importancia de la elección de este parámetro.

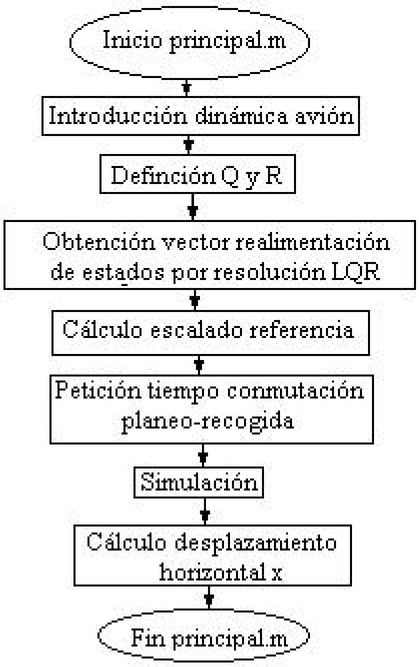


Figura 8.2

8.2 TRAYECTORIA DE REFERENCIA

Puesto que se desea mantener constante la velocidad longitudinal del avión, la señal de referencia para la variable u es constante e igual a cero. Sin embargo, la generación de la señal de referencia para la altura h es más compleja, puesto que el descenso del avión debe realizarse siguiendo una trayectoria adecuada en las fases de planeo y recogida.

La trayectoria a seguir en la fase de planeo es rectilínea como ya se adelantó en el capítulo 7. La fase de planeo comienza a una altura h(0) de 500 pies, siendo x(0)=-9540.6 pies (ver ecuación (7.9)), y se descenderá a una velocidad de 12 pies/s hasta que se produzca la conmutación a la fase de recogida en el instante txon. Esta parte de la trayectoria es fácil de generar en simulink mediante el uso de un bloque rampa como muestra la figura 8.3. Los bloques escalón y producto que aparecen en figura 8.3 sirven para desactivar la trayectoria rectilínea la finalizar la fase de planeo (txon).

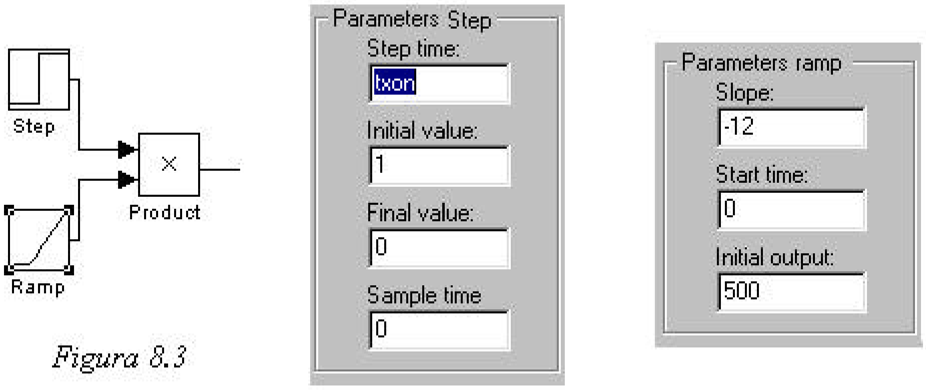


Figura 8.3

En la fase de recogida la trayectoria a seguir se genera de forma que la velocidad de descenso disminuya suavemente desde los 12 pies/s anteriores a 1.5 pies/s. Se asumirá que la fase de recogida comienza en el instante txon (final de fase de planeo), en ese instante la trayectoria generada en la fase de planeo alcanzará un valor h(txon) dado por la ecuación (8.1).

$$h(txon) = 500 - 12 * txon \quad (pies) \qquad (8.1)$$

La velocidad de descenso en el instante txon será h'(txon)=-12 pies/s, y la velocidad de descenso esperada para el instante de toma en tierra (T) será h'(T)=-1.5 pies/s. Con estos datos se puede generar la trayectoria de referencia descrita con la ecuación:

$$\begin{aligned} h(t) &= \frac{h(txon)}{h'(txon) - h'(T)}\left(h'(txon)\, e^{-(t-txon)/t} - h'(T)\right) \\ t &= -h(txon) / (h'(txon) - h'(T)) = h(txon) / 10.5 \end{aligned} \qquad (8.2)$$

La generación de la trayectoria en la fase de recogida en Simulink se realiza con el modelo mostrado en la figura 8.4. El bloque escalón sirve para activar la fase de recogida en el instante txon, manteniéndose desactivada para instantes anteriores.

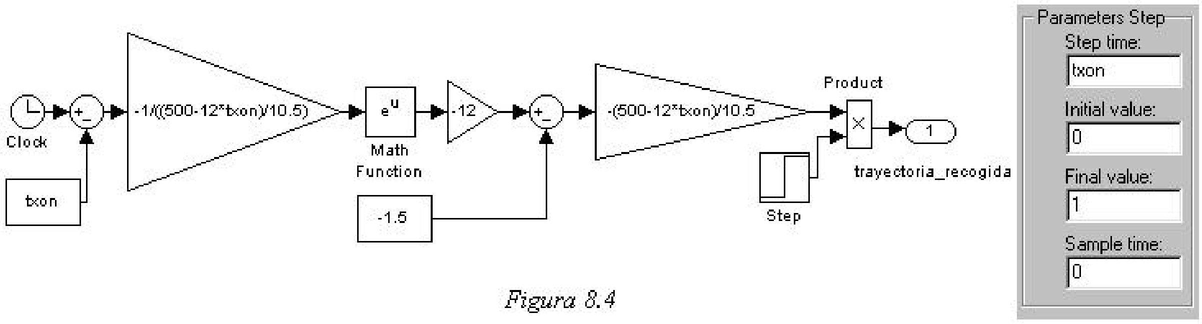


Figura 8.4

El bloque generador_trayectoria de la figura 8.1 se construye como muestra la figura 8.5, correspondiendo los bloques fase_planeo y fase_recogida a los modelos de las figuras 8.3 y 8.4 respectivamente.

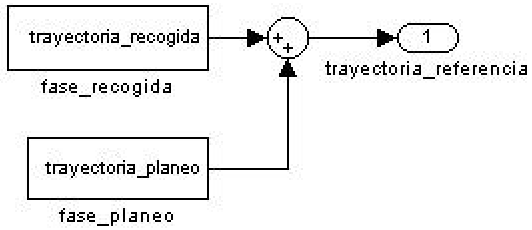


Figura 8.5

En la generación de la trayectoria se ha dejado como parámetro libre el instante de conmutación entre fases (txon), en secciones posteriores se ensayarán simulaciones con distintos valores para este parámetro.

8.3 ESCALADO DE LA TRAYECTORIA Y CÁLCULO DEL DESPLAZAMIENTO HORIZONTAL

8.3.1 Escalado de la trayectoria

La función mirscale_continuo.m incluida en el apéndice A.5.2 es la encargada de calcular la matriz de escalado a aplicar a la referencia ($K_{escalado}$ en la figura 8.1). En este archivo se plantea un sistema de ecuaciones bajo la base de que en régimen permanente la derivada del vector estado es nula. A continuación se detallan los pasos para alcanzar el sistema de ecuaciones planteado en el apéndice A.5.2.

- Notación: ref_h será la referencia en altura (trayectoria_referencia en figura 8.1); ref_u será la referencia en velocidad longitudinal (referencia_u en figura 8.1); A y B serán las matrices dinámicas del sistema en tiempo continuo; K_{LQR} es la matriz de realimentación de estados, de 2 filas y 5 conlumnas.

- Se parte de una matriz de escalado $K_{escalado}$ (en figura 8.1) de la forma:

$$K_{escalado} = \begin{bmatrix} 0 & N \\ 0 & M \end{bmatrix}$$

- Se plantea el sistema inicial:

$$A\begin{bmatrix} u \\ w \\ q \\ q \\ h \end{bmatrix} + B\left(\begin{bmatrix} 0 & N \\ 0 & M \end{bmatrix}\cdot\begin{bmatrix} ref\_u \\ ref\_h \end{bmatrix} - K\begin{bmatrix} u \\ w \\ q \\ q \\ h \end{bmatrix}\right) = 0 \qquad (8.3)$$

- Asumiendo que en régimen permanente u=ref_u=0, h=ref_h, y dividiendo por ref_h el sistema (8.3) se transforma en:

$$A\begin{bmatrix} 0 \\ w / ref\_h \\ q / ref\_h \\ q / ref\_h \\ 1 \end{bmatrix} + B\left(\cdot\begin{bmatrix} N \\ M \end{bmatrix} - K\begin{bmatrix} 0 \\ w / ref\_h \\ q / ref\_h \\ q / ref\_h \\ 1 \end{bmatrix}\right) = 0 \qquad (8.4)$$

- Se definen las variables x=w/ref_h, y=q/ref_h , z=è/ref_h. Con esto el sistema queda como

$$A\begin{bmatrix} 0 \\ x \\ y \\ z \\ 1 \end{bmatrix} + B\begin{bmatrix} N \\ M \end{bmatrix} = K\begin{bmatrix} 0 \\ x \\ y \\ z \\ 1 \end{bmatrix} \qquad (8.5)$$

que coincide con el sistema planteado en el apéndice A.5.2, y de cuya resolución obtenemos el valor de N y M para la construcción de la matriz de escalado.

8.3.2 Cálculo del desplazamiento horizontal de la aeronave

Tras la simulación del modelo de la figura 8.1 quedan definidas todas las variables del vector de estado, esto es: u, w, q, è, h. Sin embargo no se tiene información directa sobre el desplazamiento horizontal de la aeronave (downrange).

La ecuación (7.8) proporciona una aproximación a la derivada temporal de x a partir de las variables de estado. A partir de esta ecuación es fácil obtener x con un proceso de integración. Esto se realiza mediante las siguientes líneas de código del archivo principal.m (apéndice A.5.1):

```
%definición derivada temporal del downrange
xderaprox=235+w.*(theta*pi/180);

%definición del downrange
x=integra(xderaprox,tout,-9540.6);
```

La función integra.m se incluye en el apéndice A.5.3. Esta función usa la regla del trapecio para obtener x mediante la integración de su derivada (xderaprox). Para tal fin parte de x(0)=-9540.6 pies y usa el vector de tiempos tout. El uso de este vector es necesario pues Simulink no divide el tiempo de simulación en intervalos de paso constantes sino variables.

8.4 SIMULACIONES

En todas las simulaciones incluidas en esta sección se han fijado la matrices de pesos Q y R:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{bmatrix} \qquad R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

Recordando que el vector de estados es $\mathbf{X} = \begin{bmatrix} u & w & q & q & h \end{bmatrix}^T$, tenemos un peso 100 sobre el incremento de velocidad longitudinal u, y un peso 20 sobre la altura respecto a tierra de la aeronave h. En la sección 8.5 se analizará el efecto sobre el control de modificar el valor de las matriz de pesos Q.

En los apartados que siguen se estudiarán 3 casos mediante simulación que se diferencian en la elección del instante txon de conmutación entre fases, y en la elección de la velocidad de descenso deseada en el momento de la toma en tierra (h'(T)). Estos parámetros se variarán justificadamente hasta conseguir los resultados esperados. Los valores adoptados en cada simulación son:

- Primera simulación: txon=38 s, h'(T)=-1.5 pies/s
- Segunda simulación: txon=38 s, h'(T)=-2.5 pies/s
- Tercera simulación: txon=40 s, h'(T)=-1.5 pies/s y h'(T)=-2 pies/s.

8.4.1 Primera Simulación

Como se ha adelantado anteriormente, el parámetro libre que permite distintas simulaciones es el instante de conmutación txon entre las dos fases de aterrizaje. Para una primera simulación y posterior análisis de resultados, se elegirá txon=38s, que corresponde según (8.1) a una altura esperada de 44 pies para entrar en la fase de recogida. La figura 8.6 muestra la evolución temporal de la altura, la figura 8.7 muestra la evolución temporal del desplazamiento horizontal x, la figura 8.8 muestra la evolución de la altura h en función de x, y la figura 8.9 muestra la evolución temporal del ángulo de cabeceo è.

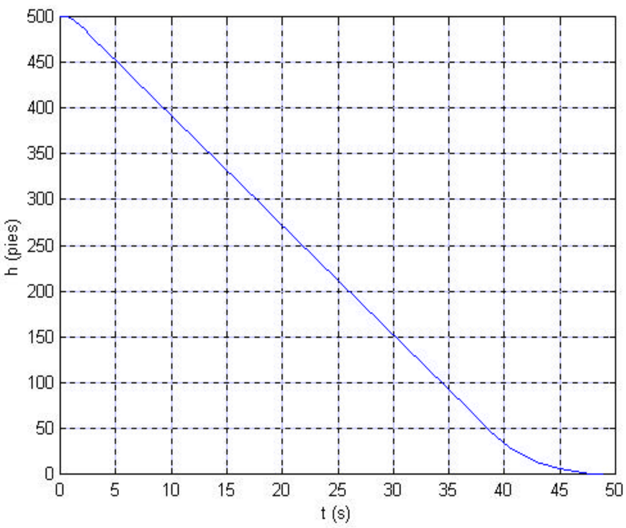


Figura 8.6

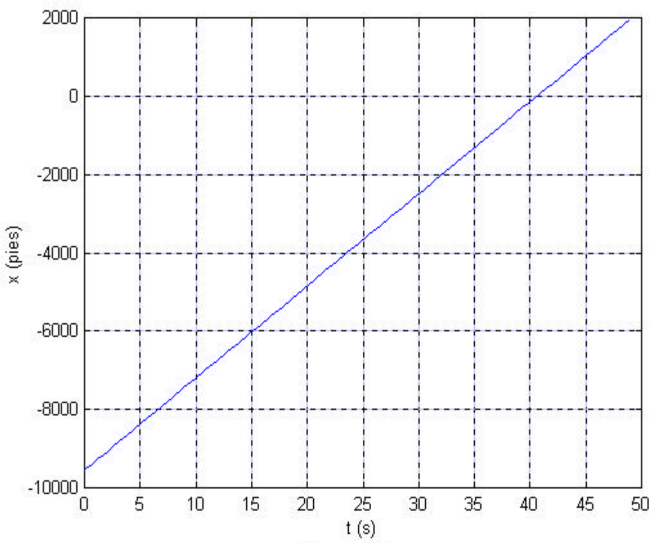


Figura 8.7

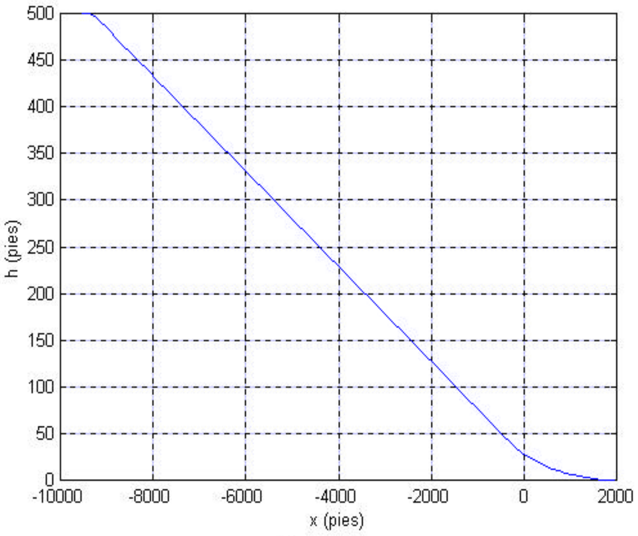


Figura 8.8

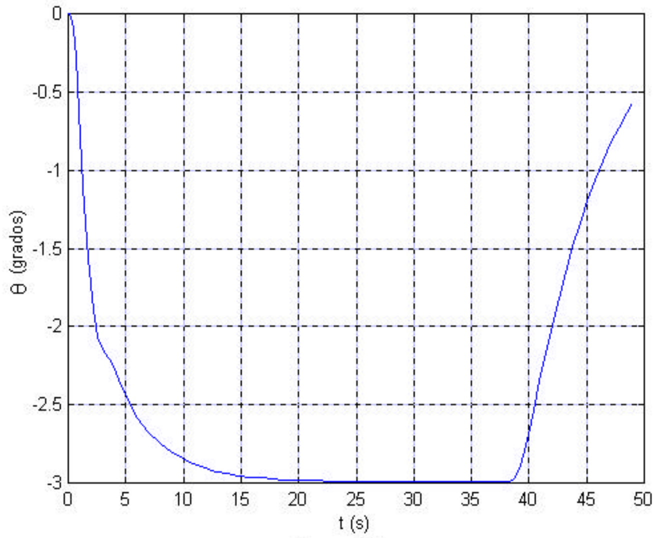


Figura 8.9

Recordando la notación T como el instante de toma de contacto de la aeronave con tierra, pueden deducirse los siguientes resultados de la observación de las gráficas anteriores:

T=47.8 s; x(T)=1684 pies; x'(T) x/ t =235 pies/s; h'(T) h/ t = -1.61 pies/s; è(T)=-0.7º

Según las condiciones deseadas para un aterrizaje exitoso (7.10), se observa que no se cumple la condición -300 x(T) pies 1000, por lo que el aterrizaje no se puede considerar exitoso y habrá que intentar otras simulaciones en los apartados siguientes.

Por otro lado se puede observar que en el instante de conmutación a la fase de recogida la altura de la aeronave es h(txon)=56 pies, que no coincide con los 44 pies esperados. La diferencia se debe al retardo producido en el seguimiento de la trayectoria, este retardo queda patente en la figura 8.10 donde se representan la evolución temporal de h y de la trayectoria de referencia, observándose un desfase de 1s entre ambas representaciones. Si se considera este retardo, la altura de la aeronave en el momento de finalizar la fase de planeo coincide con los 44 pies esperados, esto es h(txon+1)=44 pies. En la sección 8.5 se analizará el efecto de la matriz Q sobre este retardo.

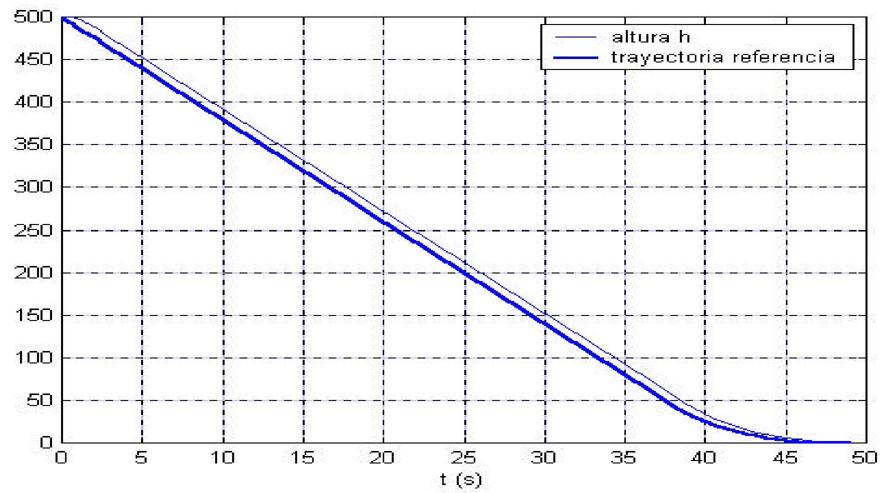


Figura 8.10

Las figuras 8.11, 8.12, 8.13, y 8.14 corresponden respectivamente las evoluciones temporales de: la deflexión del timón de profundidad; el empuje; el incremento de velocidad longitudinal u; la velocidad longitudinal ($u+U_0$). Estas figuras junto con la figura 8.9 proporcionan un ejemplo cualitativo muy interesante del proceso de aterrizaje:

- Al comienzo de la fase de planeo el avión se encuentra nivelado con ángulo de cabeceo è = 0.
- El timón de profundidad gira entonces hacia abajo, con lo que el morro del avión baja disminuyendo el ángulo de cabeceo y el de ataque (ver 2.3). La disminución del ángulo de ataque manteniéndose constante la velocidad longitudinal del avión provoca el descenso del avión.
- Para mantener la velocidad longitudinal del avión constante habiendo disminuido el ángulo de ataque es necesario una disminución del empuje o gas, como así se puede observar en la figura 8.12.
- Al comienzo de la fase de recogida aumenta gradualmente el ángulo de cabeceo y el de ataque gracias a una deflexión positiva en el timón de profundidad.
- Al haber aumentado el ángulo de ataque, para mantener constante la velocidad es necesario un aumento de potencia o gas. Eso sí, el aumento de gas será siempre inferior (<0) al valor inicial de vuelo equilibrado, pues el ángulo è es menor que su valor inicial 0 en todo caso.

Para una mejor comprensión de las líneas anteriores es aconsejable la lectura del apéndice B.1. A lo que hay que añadir que gracias a la observación de la figura 8.8, se deduce para la fase de planeo un ángulo de trayectoria constante ã -3º (pendiente de h-x), por lo que el ángulo de ataque será á 3+è grados. Esto permite usar el ángulo de cabeceo en lugar del ángulo de ataque para explicar el comportamiento cualitativo anterior.

Observando las figuras 8.13 y 8.14 se comprueba que la velocidad longitudinal del avión permanece prácticamente constante con una variación del 0.38% (0.9*100/235).

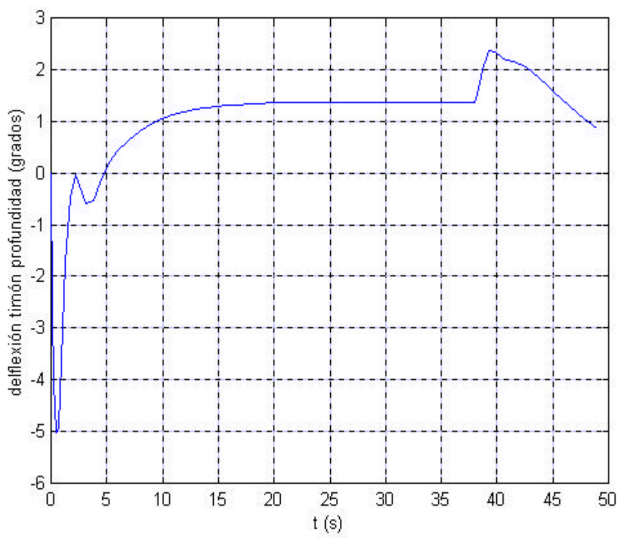


Figura 8.11

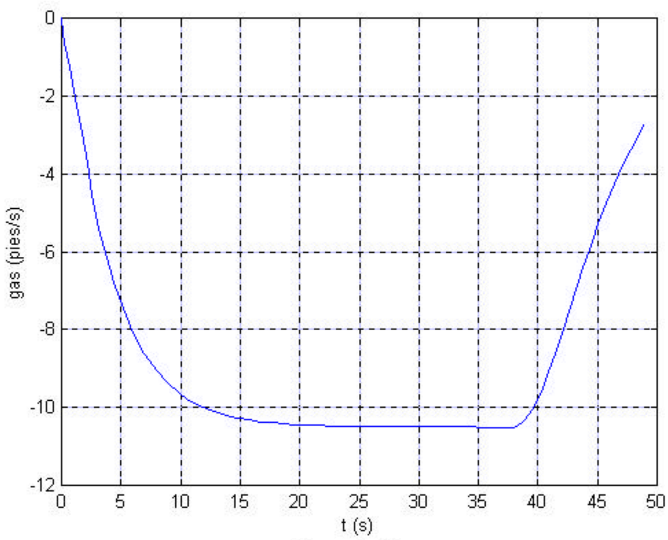


Figura 8.12

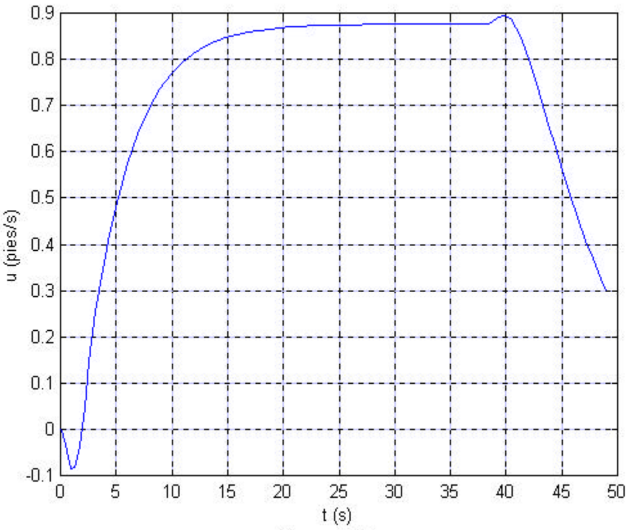


Figura 8.13

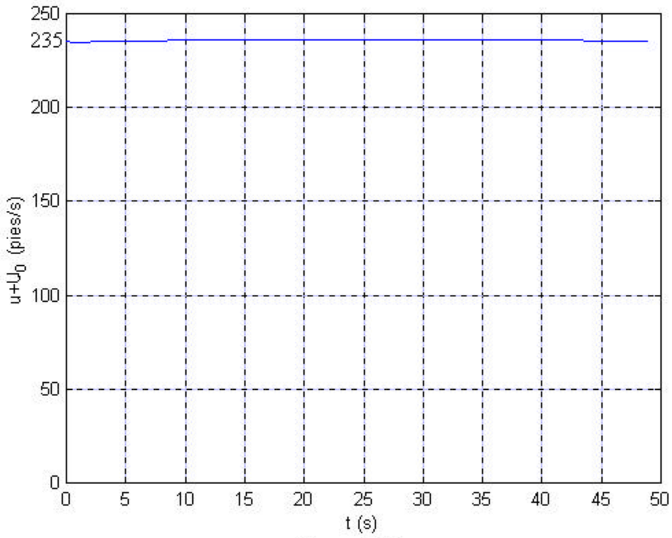


Figura 8.14

8.4.2 Segunda Simulación

El resultado de la simulación del apartado anterior no fue satisfactorio, puesto que no se cumplía la restricción sobre x(T). La trayectoria de la fase de recogida se programó para obtener un velocidad de descenso de 1.5 pies/s en el momento de la toma de contacto con tierra. En este apartado se intentará solucionar el problema de la restricción en x(T) aumentando la velocidad de descenso a 2.5 pies/s en el momento de la toma, puesto que este valor sigue respectando la restricción sobre h’(T).

Para realizar la simulación con velocidad de descenso en la toma de 2.5 pies/s solo se debe modificar el esquema de la figura 8.4, cambiando el bloque constante –1.5 por otro con valor –2.5, y sustituyendo el valor 10.5 (12-1.5) de los bloques triangulares de ganancia por 9.5 (12-2.5).

Los resultados de la simulación para la fase de planeo son idénticos a los del apartado anterior mostrados en las figuras 8.6 a 8.9, variando éstos en la fase de recogida. Las figuras 8.15 a 8.18 muestran los resultados de simulación en la fase de recogida, comparándolos con los obtenidos en la simulación del apartado anterior (gráficos discontinuos). De esta comparación se deduce una fase de recogida menos suave y de menor duración para la simulación de este apartado (h’(T) -2.5). Resultando:

T=46.3 s; x(T)=1340 pies; x’(T) x/ t =235 pies/s; h’(T) h/ t = -2.62 pies/s; è(T)=-1º

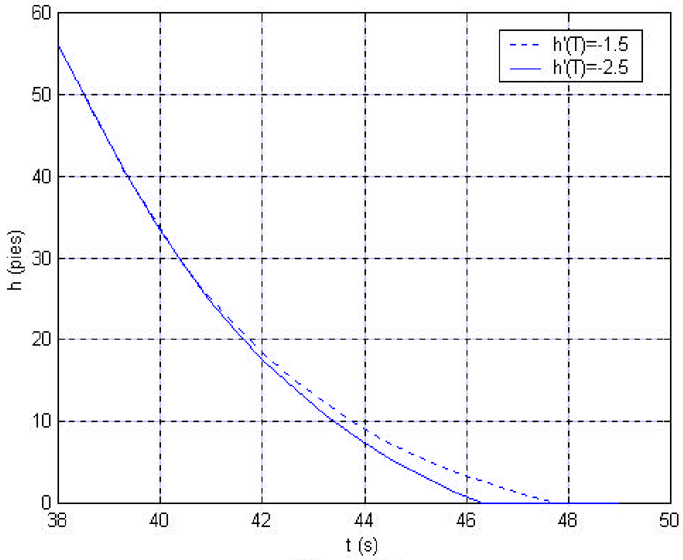


Figura 8.15

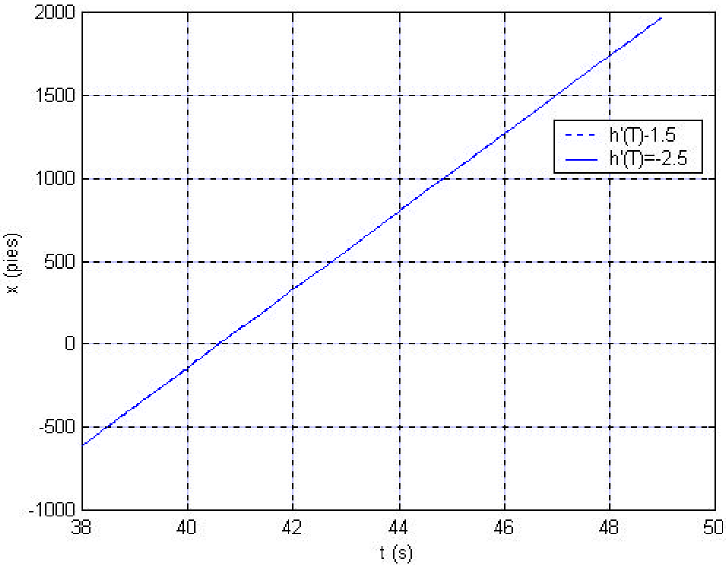


Figura 8.16

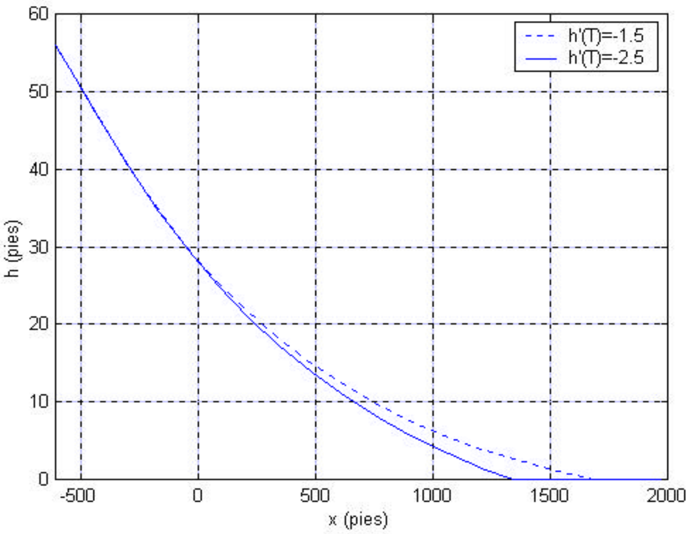


Figura 8.17

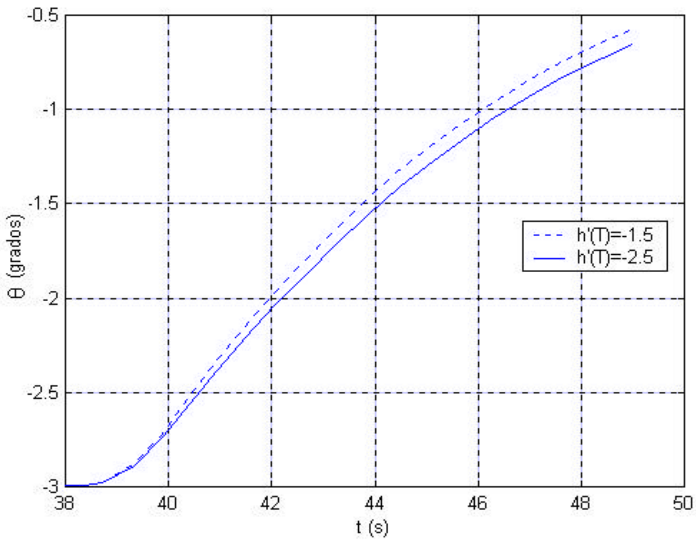


Figura 8.18

En la figura 8.17 se puede apreciar el efecto esperado en la disminución de x(T). Sin embargo esta disminución no es suficiente, ya que x(T) sigue sin cumplir la condición en (7.10) correspondiente (-300 x(T) pies 1000).

8.4.3 Tercera Simulación

En este apartado se ensayará una nueva solución para conseguir disminuir x(T). En un primer intento se retrasará el comienzo de la fase de recogida dejando fija la velocidad de descenso en la toma de contacto con tierra a su valor inicial del apartado 8.4.2, esto es 1.5 pies/s. El retraso en la conmutación de fases conlleva una disminución de la altura con la que se inicia la fase de recogida h(txon). Antes de simular es necesario comprobar que dicha altura permanece en el rango [20 70] pies recomendado en el capítulo 3 tras la simulación. Si se acude a la ecuación (8.1) y se opta por trabajar al límite h(txon)=20 pies, resulta un instante de conmutación txon=40 s. Este valor es el máximo posible para comenzar la fase de recogida, sin embargo resulta insuficiente, pues como se observa en la figura 8.19, x(T) es mayor que 1000 pies.

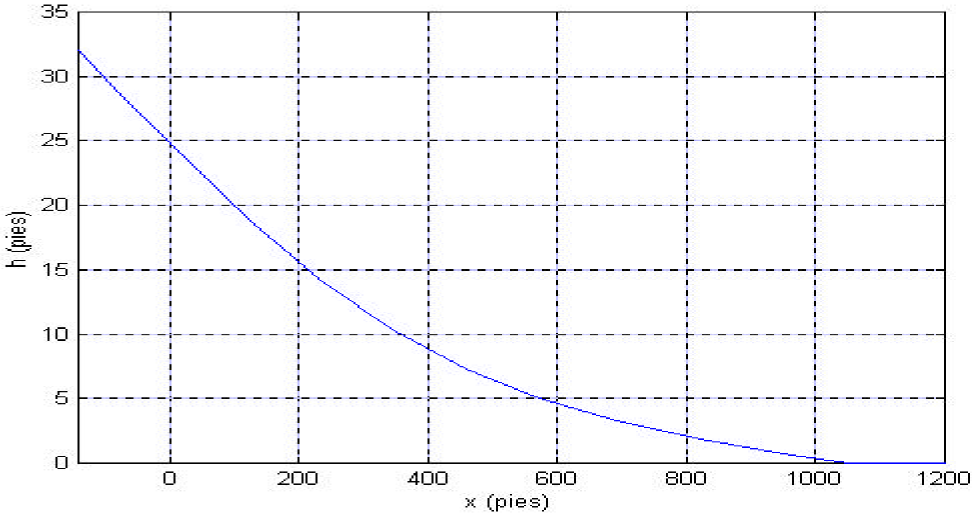


Figura 8.19

Tras un primer intento fallido, es posible conseguir un resultado satisfactorio si combinamos el retraso en el comienzo de la fase de recogida con el aumento en la velocidad de descenso en la toma ensayado en el apartado 8.4.2. Con este fin se adoptará una velocidad de descenso en la toma de contacto en tierra de 2 pies/s, y un instante de conmutación txon=40 s.

En lo que se refiere a la fase de planeo se hace notar que la única diferencia con los apartados 8.4.1 y 8.4.2 es un ligero aumento de su duración, que ahora durará 40 s en lugar de 38 s. Las figuras 8.20 y 8.21 muestra de forma comparativa los resultados más representativos de las tres simulaciones realizadas.

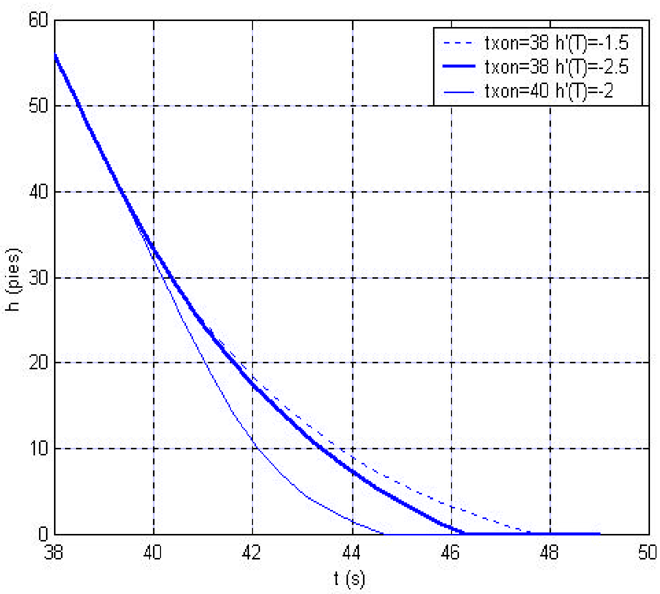


Figura 8.20

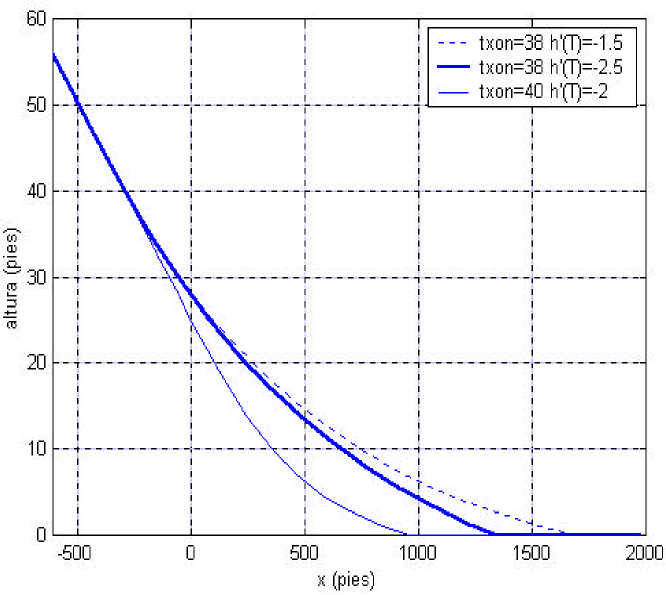


Figura 8.21

Con la nueva solución adoptada se cumplen todas las condiciones (7.10), resultando los siguientes valores:

T=44.7 s; x(T)=956 pies; x’(T) x/ t =235.1 pies/s; h’(T) h/ t = -2.2 pies/s; è(T)=-1.2º

Para finalizar este apartado se incluyen las figuras 8.22 a 8.25 con las representaciones de las actuaciones y de la velocidad longitudinal. Todas ellas son similares a las gráficas de las figuras 8.11 a 8.14, por lo que pueden ser sometidas al mismo análisis cualitativo realizado en el apartado 8.4.1.

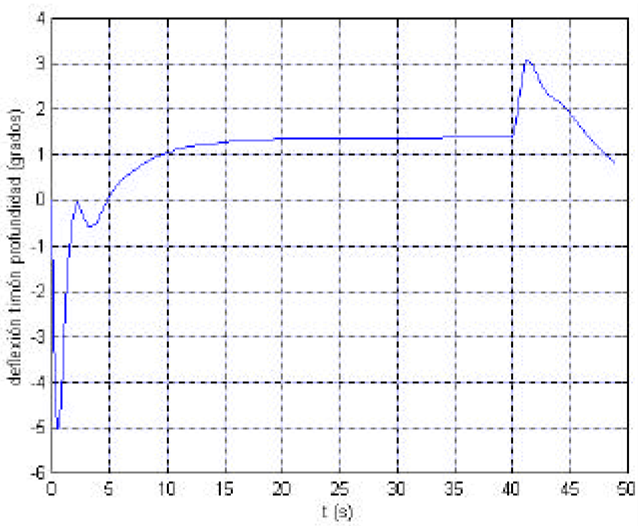


Figura 8.22

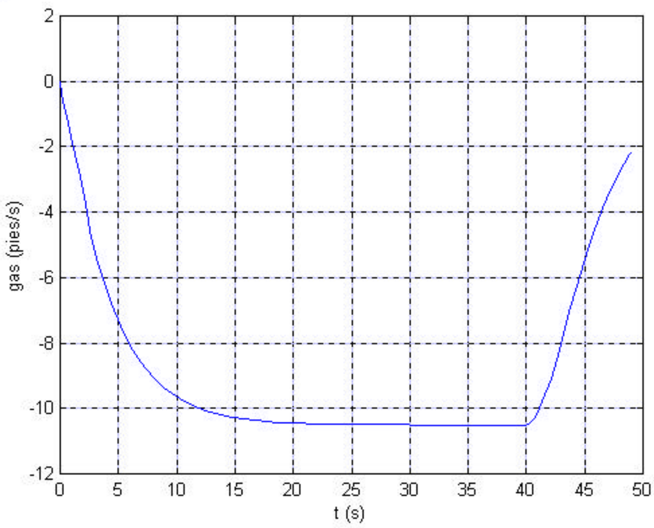


Figura 8.23

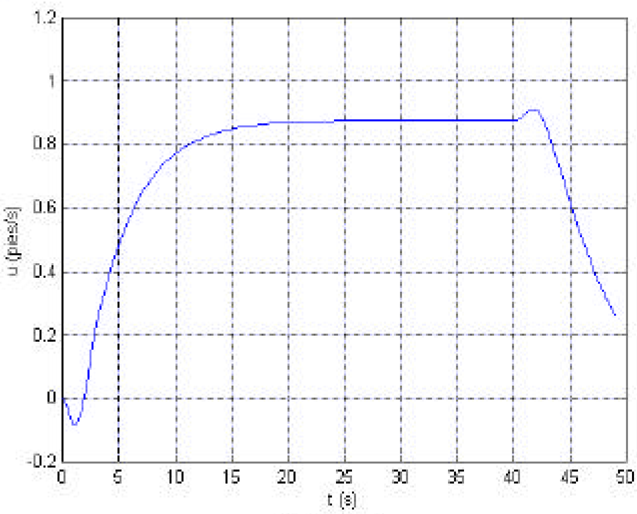


Figura 8.24

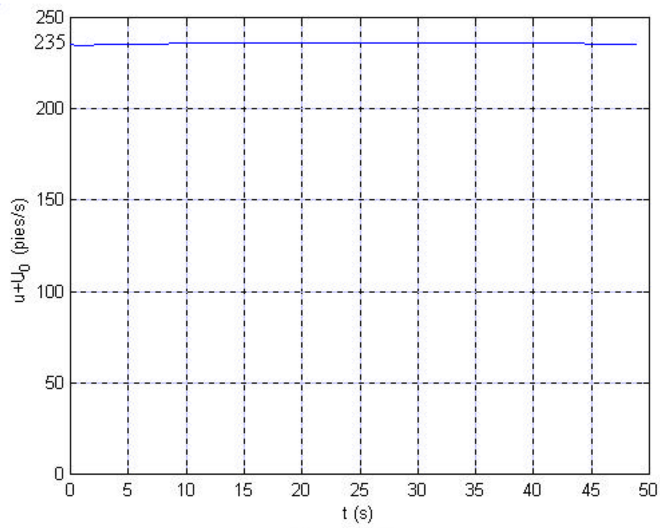


Figura 8.25

8.5 INFLUENCIA DE LA MATRIZ Q EN EL CONTROL

Atendiendo a los resultados satisfactorios obtenidos en el último apartado de la sección anterior, se elegirá para todas las simulaciones de esta sección un instante de conmutación txon= 40 s y una velocidad de descenso en la toma de contacto con tierra de 2 pies/s. Asumido esto, se realizarán distintas simulaciones para valores distintos de la matriz Q con R fija, analizando las variaciones en la repuesta del sistema.

$$Q = \begin{bmatrix} Pu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Ph \end{bmatrix} \qquad R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$$

La tabla 8.1 muestra para cada simulación: el instante de toma de contacto con tierra T, x(T), el ángulo de cabeceo en la toma è(T), el rango de variación de u, el rango de variación del empuje o gas, y el retardo entre la trayectoria de referencia y la trayectoria real seguida por la altura del avión al que ya se hizo referencia con la figura 8.10 (ver 8.4.1). Para todas las simulaciones resulta h’(T) -2 pies/s y x’(T) 235 pies/s.

Pu	Ph	T (s)	x(T) (pies)	è(T) grado	u (pies/s)	gas (pies/s)	retardo (s)
100	20	44.67	956	-1.2	[-0.09,0.91]	[-10.51,0]	1
20	20	44.64	948	-0.87	[-0.05,3.26]	[-8.2,0]	1
20	100	44.33	875	-0.87	[-0.03,3.50]	[-7.93,0]	0.8
0	100	44.31	868	0.35	[-0.01,11.93]	[-0.3,0.6]	0.8

Tabla 8.1

Los resultados mostrados en la tabla 8.1 pueden ser interpretados atendiendo a las características del control LQR como:

- Ph corresponde al peso en la función objetivo para el seguimiento de la trayectoria en altura, estando directamente relacionado con T, x(T) y el retardo en el seguimiento. Estos parámetros toman valores más o menos constante para Ph constante, y a medida que aumenta Ph la respuesta se hace más rápida disminuyendo T, x(T) y el retardo.

- Pu corresponde al peso para el seguimiento de la velocidad incremental u, por lo que para Pu constante el rango de valores de u varía poco. Un valor Pu grande aumenta la influencia del error en seguimiento de u dentro de la función a optimizar, por lo que el controlador intentará minimizar su variación entorno a cero. Para valores Pu bajos, o incluso nulos, la variación de u aumenta, dejando de ser válida la aproximación de velocidad longitudinal constante ($u+U_0$ U_0), asumida en este capítulo. Por este motivo en las simulaciones de las secciones anteriores se tomo Pu=100 (ver A.5.1).

- Del análisis del rango del empuje o gas es interesante observar que, para Pu=0, el uso de dicha actuación es prácticamente nulo, corroborando que la labor principal de la actuación gas es mantener la velocidad longitudinal constante (u=0).

El resultado obtenido con Pu=0 y Ph=100 es similar al obtenido con el control LQR SISO de la altura de una aeronave en aterrizaje, en cuya dinámica se suprime la actuación gas.

Este resultado se puede comprobar en la figura 8.26, donde se ha representado la deflexión del timón de profundidad para el caso MIMO (Pu=0, Ph=100) y para el caso SISO (Ph=100). Las representaciones x(t), h(t), h(x) no se muestran aquí, por no ofrecer diferencias visibles. La simulación del caso SISO se ajusta perfectamente a la última fila de la tabla 8.1, salvo en que no existe la actuación gas, y el rango de u es [-0.0001,11.68]. En el apéndice A.5.4 se incluyen los archivos necesarios para la simulación del control SISO.

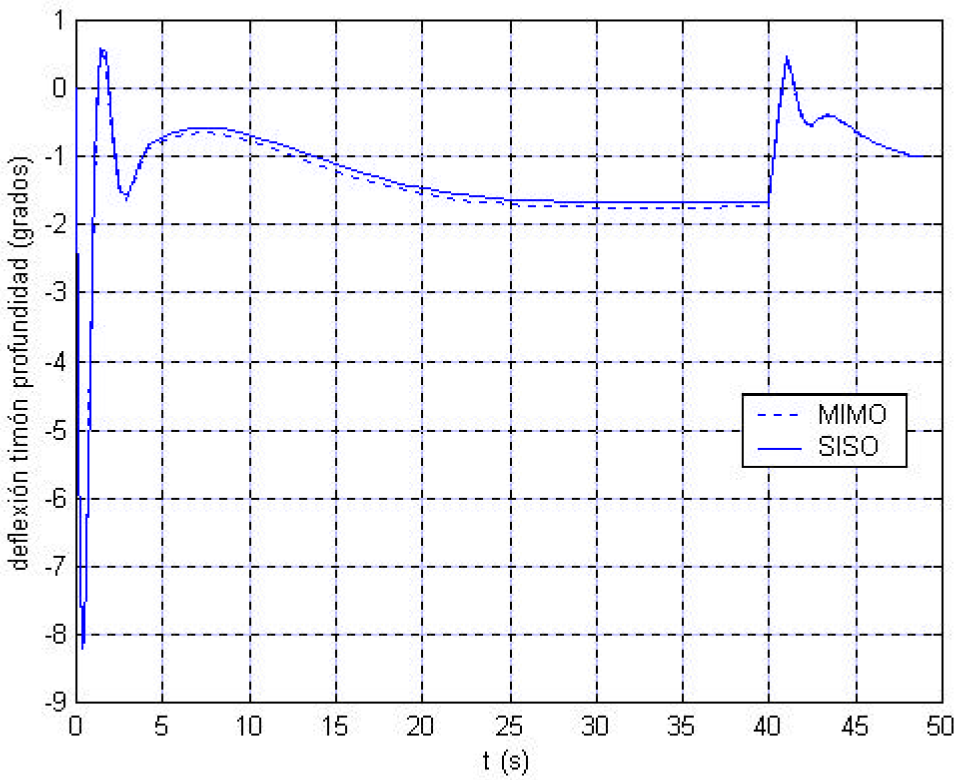


Figura 8.26

Capítulo 9

Control robusto H

9.1 INTRODUCCIÓN

Los orígenes de la teoría de control, o si se quiere, la teoría de los sistemas realimentados, marcan una estrecha relación entre la realimentación y la incertidumbre sobre la conducta dinámica del sistema a controlar. Esa incertidumbre es inherente a cualquier sistema físico o resulta de nuestra limitación en comprender o modelar la conducta del sistema. La teoría de control robusto incorpora explícitamente la incertidumbre al modelado, el análisis y la síntesis de sistemas de control. El control H queda dentro del marco de la teoría de control robusto.

Para introducirse en el control robusto H es necesario establecer una configuración previa de control estándar. Dicha configuración se muestra en la figura 9.1 donde v representa las salidas medidas, w refiere a las señales externas (ruido r, perturbaciones a entrada d_i y/o salida d_o, señales de referencia r), z es el vector de errores, y u es la señal de realimentación proporcionada por el controlador K(s).

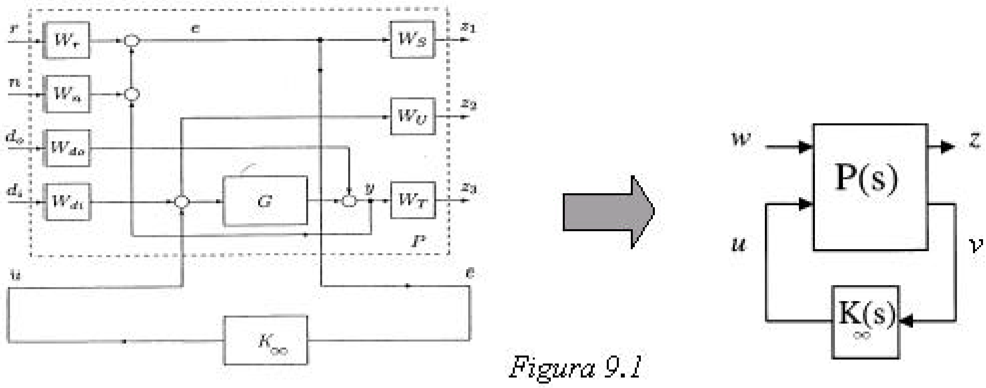


Figura 9.1

El sistema anterior puede describirse matricialmente por:

$$\begin{bmatrix} z \\ v \end{bmatrix} = P(s)\begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}\begin{bmatrix} w \\ u \end{bmatrix} \qquad u = K(s)v \qquad (9.1)$$

Atendiendo a la figura 9.1 y la ecuación 9.1 se puede establecer la función de transferencia en bucle cerrado de w a z como:

$$z = F_l(P,K)w \qquad F_l(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}$$

El objetivo del control H es encontrar todos los controladores K(s) que estabilicen de forma robusta la planta, y minimicen la norma infinita || F_l(P,K) || .

9.2 DISEÑO H MODE LADO DE LAZO

El procedimiento de diseño H loop -shaping combina la teoría de modelado de lazo clásica con la teoría de estabilización H de control robusto. H loop -shaping implica dos fases de diseño: una primera en la que se aumenta la dinámica de la planta en lazo abierto con pre y post compensadores para obtener una respuesta en frecuencia del lazo abierto deseable, y una segunda en que se estabiliza de forma robusta la planta resultante de la fase anterior.

9.2.1 Estabilización robusta

Se considerará la estabilización de una planta G, que puede ser descrita por una factorización prima a izquierdas [28] $G=M^{-1}N$. Atendiendo a la figura 9.2 se puede definir el modelo de una planta perturbada (9.2), donde $_{M,}$ $_{N}$ son funciones de transferencias estables desconocidas que representan la incertidumbre en el modelo de la planta nominal.

$$Gp=(M+\ _M)^{-1}(N+\ _N) \qquad (9.2)$$

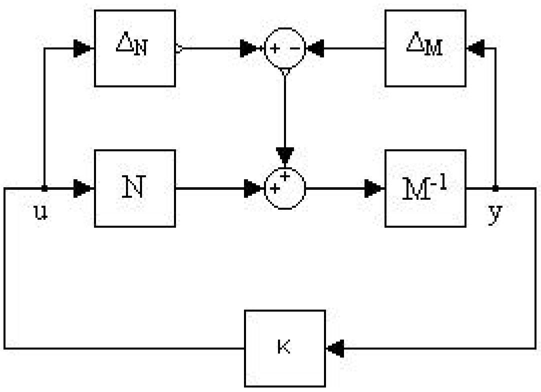


Figura 9.2

El objetivo es estabilizar toda una familia de plantas perturbadas:

$$Gp=\{(M+\ _M)^{-1}(N+\ _N): \|[\ _N \quad _M]\| < å$$

donde å>0 es el margen de estabilidad. El problema de la estabilización robusta consiste en maximizar este margen de estabilidad. El sistema realimentado de la figura 9.2 posee estabilidad robusta si y solo si el sistema nominal realimentado es estable y:

$$g = \left\| \begin{bmatrix} \mathrm{K} \\ \mathrm{I} \end{bmatrix} (\mathrm{I} - \mathrm{GK})^{-1} \mathrm{M}^{-1} \right\|_{\infty} \leq \frac{1}{e} \qquad (9.3)$$

El mínimo valor de ã posible correspondiente al máximo margen de estabilidad viene dado por Glover y Mcfarlane (1989) como:

$$g_{\min} = e^{-1}{}_{\max} = \left\{ 1 - \left\| \mathrm{N\ M} \right\|^{2}_{\mathrm{H}} \right\}^{-\frac{1}{2}} = (1 + r(\mathrm{XZ}))^{\frac{1}{2}}$$

donde para una realización de G en el espacio de estados (A, B, C, D), Z y X son la única solución definida positiva de la ecuaciones de Riccati (9.4) y (9.5) respectivamente.

$$(\mathrm{A} - \mathrm{BS}^{-1}\mathrm{D}^{\mathrm{T}}\mathrm{C})\mathrm{Z} + \mathrm{Z}(\mathrm{A} - \mathrm{BS}^{-1}\mathrm{D}^{\mathrm{T}}\mathrm{C})^{\mathrm{T}} - \mathrm{ZC}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{CZ} + \mathrm{BS}^{-1}\mathrm{B}^{\mathrm{T}} = 0 \qquad (9.4)$$

$$\mathrm{R} = \mathrm{I} + \mathrm{DD}^{\mathrm{T}}, \qquad \mathrm{S} = \mathrm{I} + \mathrm{D}^{\mathrm{T}}\mathrm{D}$$

$$(\mathrm{A} - \mathrm{BS}^{-1}\mathrm{D}^{\mathrm{T}}\mathrm{C})^{\mathrm{T}}\mathrm{X} + \mathrm{X}(\mathrm{A} - \mathrm{BS}^{-1}\mathrm{D}^{\mathrm{T}}\mathrm{C}) - \mathrm{XBS}^{-1}\mathrm{B}^{\mathrm{T}}\mathrm{X} + \mathrm{C}^{\mathrm{T}}\mathrm{R}^{-1}\mathrm{C} = 0 \qquad (9.5)$$

Un controlador K que garantiza (9.3) para un valor ã> $ã_{min}$ es:

$$\mathrm{K} \overset{s}{=} \left[\begin{array}{cc} \mathrm{A} + \mathrm{BF} + g^{2}(\mathrm{L}^{\mathrm{T}})^{-1}\mathrm{ZC}^{\mathrm{T}}(\mathrm{C} + \mathrm{DF}) & g^{2}(\mathrm{L}^{\mathrm{T}})^{-1}\mathrm{ZC}^{\mathrm{T}} \\ \mathrm{B}^{\mathrm{T}}\mathrm{X} & -\mathrm{D}^{\mathrm{T}} \end{array} \right]$$

$$\mathrm{F} = -\mathrm{S}^{-1}(\mathrm{D}^{\mathrm{T}}\mathrm{C} + \mathrm{B}^{\mathrm{T}}\mathrm{X}) \qquad \mathrm{L} = (1 - g^{2})\mathrm{I} + \mathrm{XZ}$$

En el apéndice A.6.1 se incluye el archivo coprimeunc.m en el que se realizan los cálculos necesarios para la obtención del controlador anterior.

9.2.2 Controlador con dos grados de libertad

Cuando se trata de seguir una referencia no nula el esquema de la figura 9.2 es válido siempre que se añada una matriz de escalado adecuadamente como se verá en el apartado siguiente. Sin embargo, si se desea que el seguimiento de la referencia responda a una determinada dinámica (T_{ref}) es necesario introducir un esquema de realimentación con dos grados de libertad, donde la señal de referencia y la de realimentación son tratadas de foma independiente por el controlador. Dicho esquema se muestra en la figura 9.3, donde $Gs=Ms^{-1}Ns$ corresponde a la planta en lazo abierto compensada con un prefiltro W_1.

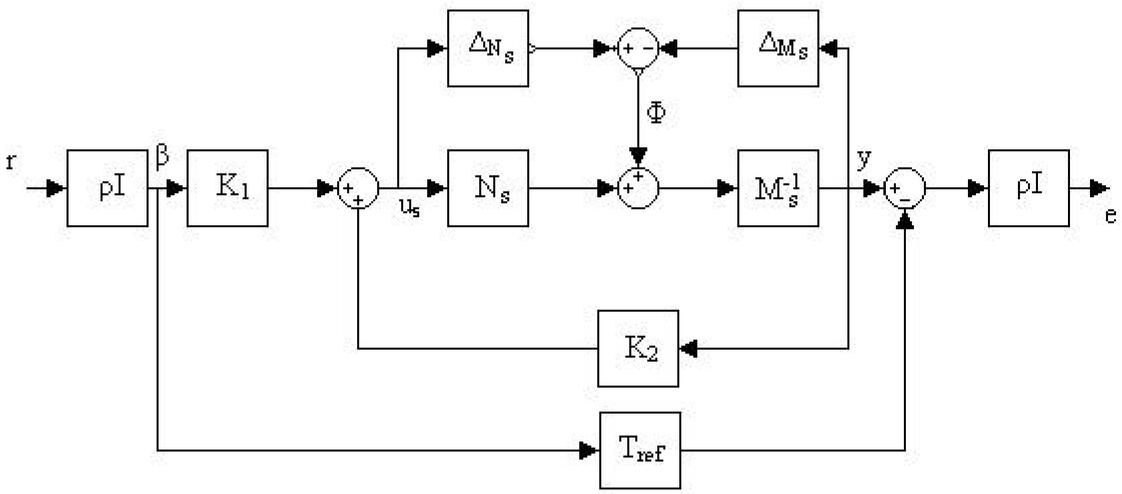


Figura 9.3

El problema de diseño consiste en encontrar el controlador estabilizador $K=[K_1\ K_2]$ para la planta compensada $G_s=GW_1$, que minimice la norma H de la función de transferencia (9.6) entre las señales $[r^T\ Ö^T]$ y $[u_s^T\ y^T\ e^T]$ definidas según la figura 9.3. La señal de control aplicada a la planta compensada es:

$$u_s = [K_1\ K_2]\begin{bmatrix} b \\ y \end{bmatrix}$$

En el apéndice A.6.2 se incluye el archivo dosDOFcontrol.m con el código necesario para el cálculo del controlador K, donde se ha tomado como parámetro de diseño ñ=1. Para mayor detalle de la resolución del problema se remite al lector a [29].

9.2.3 Escalado para error en régimen permanente nulo

Las figura 9.4 muestra el esquema de control resultante para los controladores robustos H de uno (1-DOF) y dos grados (2-DOF) de libertad aplicados a una planta compensada $G_s=GW_1$.

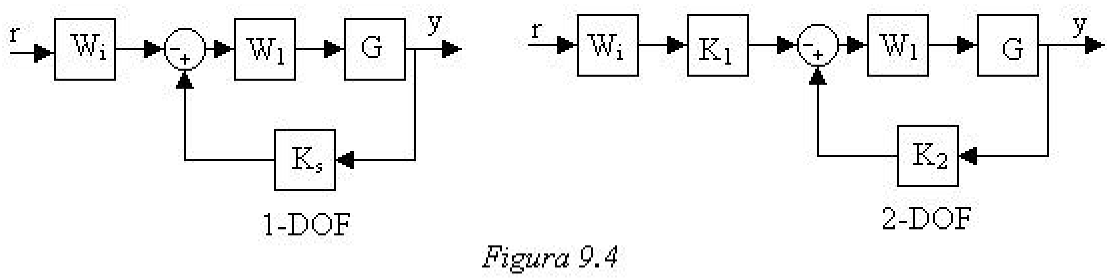


Figura 9.4

En ambos casos es necesario aplicar una matriz de escalado W_i a la referencia. Suponiendo la existencia de un integrador (1/s) en el prefiltro W_1 y/o en la planta G, es fácil obtener (véase apéndice B.2) el valor de W_i para conseguir un error en régimen permanente nulo:

.1-DOF: $W_i=K_s(0)$
.2-DOF: $W_i = [(G_{ss}(0)K_2(0))^{-1}G_{ss}(0)K_1(0)]^{-1}$, siendo $G_{ss}(0) = \lim_{s\to 0} sG_s(s)$

9.3 PROBLEMA DE CONTROL

El problema de control que se plantea coincide con el expuesto en la sección 7.2 del capítulo 7 para el proceso de aterrizaje de una aeronave. Se trata de controlar la trayectoria del aeroplano en sus dos fases de aterrizaje manteniéndose dentro de unas condiciones de seguridad dadas en (7.10). Sin embargo en este capítulo se abordará el control de una manera ligeramente distinta (inspirada en [27]), como ya se verá en los apartados siguientes.

9.3.1 Fase de planeo (Glide Control)

A partir del modelo (7.4) se construirá el modelo dinámico de la aeronave utilizado en este apartado. El primer paso es eliminar el estado h (altura) de (7.4), y modificar la matriz de salida C de forma que la salida del sistema quede definida por **y**= [u ã]. El ángulo de trayectoria ã se puede aproximar por:

$$g = q - a = q - \frac{w}{U_0}$$

con lo que el sistema queda descrito en el espacio de estados por:

$$\mathbf{X}' = \begin{bmatrix} -0.0380 & -0.0513 & 0.00152 & -0.5612 \\ 0.3130 & -0.6050 & -4.1425 & -0.0294 \\ -0.0211 & 0.157 & -0.612 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \mathbf{X} + \begin{bmatrix} 0.00005 & 0.158 \\ -0.146 & 0.031 \\ 0.459 & 0.0543 \\ 0 & 0 \end{bmatrix} . \mathbf{u}$$

$$\mathbf{X} = \begin{bmatrix} u & w & q & q \end{bmatrix}^T \qquad \mathbf{u} = [d_e \;\; d_T] \tag{9.6}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{235} & 0 & 1 \end{bmatrix}$$

Se pretende controlar dicha dinámica para obtener un ángulo de trayectoria de –3º (ã=-3), manteniéndose constante la velocidad longitudinal del avión (u=0). Este enfoque de control es distinto al adoptado en el capítulo anterior, en el que durante la fase de planeo se controlaba la altura de la aeronave para que siguiera una trayectoria rectilínea con una velocidad de descenso de 12 pies/s. Sin embargo, a velocidad 235 pies/s constante, un ángulo de trayectoria de –3º constante implica una velocidad de descenso constante de valor h'=tan(-3º)*235 -12 pies/s. Este resultado se comprobará posteriormente tras realizar las simulaciones correspondientes.

El primer paso antes de estabilizar de manera robusta el sistema (9.6) es compensar dicho sistema. En un principio se compensará el sistema aeronave con un sencillo prefiltro

(9.7). La misión de este prefiltro es conseguir un error en régimen permanente nulo con un escalado adecuado de la señal de referencia según se establece en el apartado 9.2.3.

$$W_1 = \begin{bmatrix} 1/s & 0 \\ 0 & 1/s \end{bmatrix} \qquad (9.7)$$

Si se denota por G al sistema descrito por (9.6) y por G_s al sistema compensado con el prefiltro W_1, se tendrá un sistema compensado con integradores en todas sus funciones de transferencia:

$$G_s = GW_1 = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} 1/s & 0 \\ 0 & 1/s \end{bmatrix} = \begin{bmatrix} G_{11}(s)/s & G_{12}(s)/s \\ G_{21}(s)/s & G_{22}(s)/s \end{bmatrix}$$

La figura (9.5) muestra el diagrama de Simulink el sistema compensado y realimentado. En dicho diagrama se determina de forma intermedia el estado del sistema, pues el uso de las variables de estado será necesario para determinar la altura y posición de la aeronave.

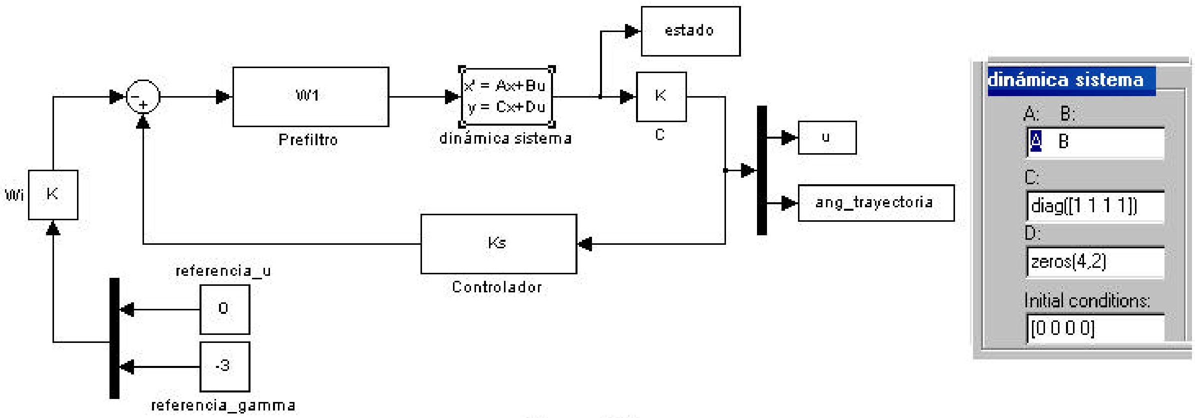


Figura 9.5

En el apéndice A.6.3 se incluye el archivo principal.m para la simulación del modelo de la figura 9.5. En dicho archivo se determina el controlador Ks por el principio de estabilización robusta expuesto en el apartado 9.2.1, haciendo uso de la función coprimeunc.m del apéndice A.6.1. El escalado Wi se determina acudiendo al apartado 9.2.3 en el caso de un controlador con un grado de libertad (1-DOF). La sucesión de tareas realizadas en A.6.3 se muestran en la figura 9.6.

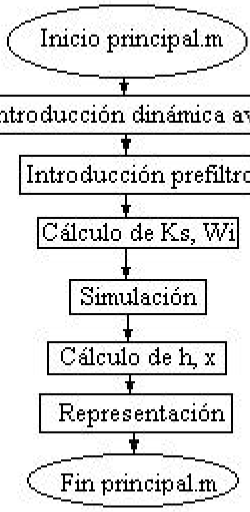


Figura 9.6

Tras la ejecución de principal.m se obtiene un valor ã$_{min}$=3.9 (å$_{max}$=0.26) y la trayectoria mostrada con trazo continuo en la figura 9.7. En la figura 9.7 se ha representado también con trazo discontinuo la línea central de planeo de referencia (GSC glide slope centerline). Se observa que la trayectoria del avión es paralela a la GSC, pues se consigue un valor ã=-3º (ver figura 9.8), sin embargo debido al retraso inherente al control la trayectoria del avión está lejos de dirigirse hacia la estación en tierra.

La figura 9.8 muestra la evolución temporal del ángulo de trayectoria del sistema realimentado. No se muestran más resultados de simulación, por ser la solución adoptada hasta este momento una solución parcial.

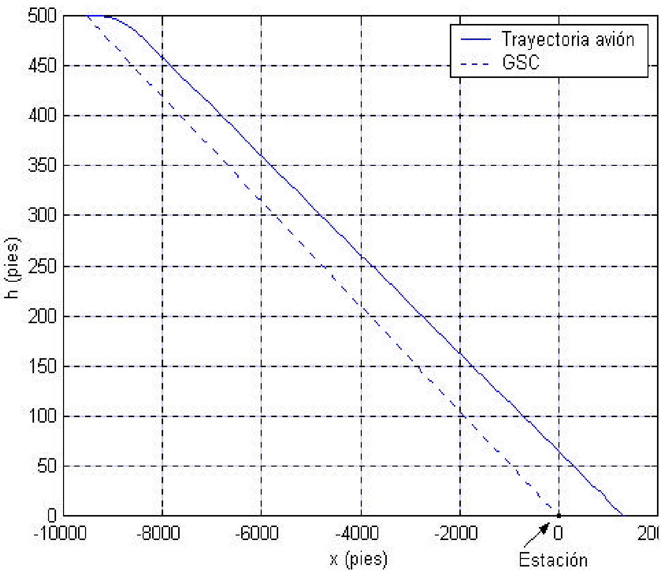


Figura 9.7

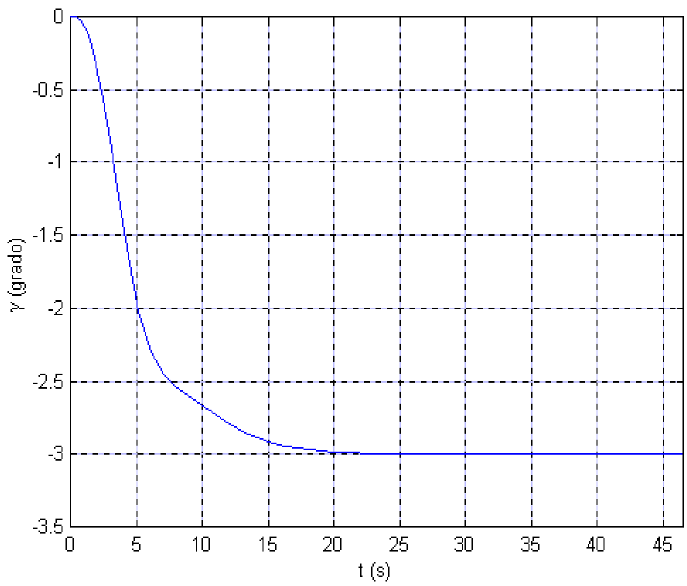
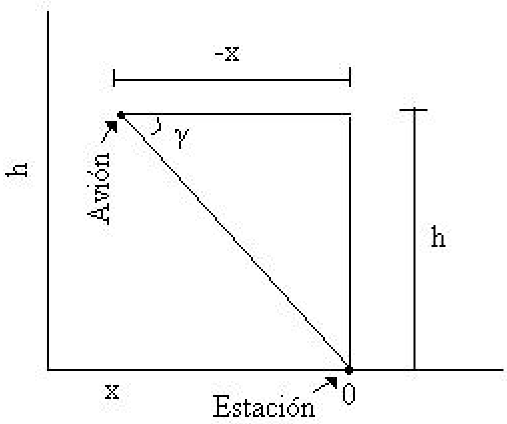


Figura 9.8

Si se tiene en cuenta el retardo en el seguimiento y se pretende que la trayectoria del avión se dirija a la estación terrestre es necesario modificar el ángulo de trayectoria de referencia según la posición de la aeronave en vuelo como se muestra en la figura 9.9.



$$\gamma = \text{atan}\left(\frac{h}{-x}\right)$$

Figura 9.9

Ahora el ángulo de trayectoria de referencia deja de ser un valor constante, y es necesario calcularlo en cada instante de simulación. Para realizar los cálculos necesarios se hace uso de una función de sistema (S-function, ver apéndice A.3.1), creándose la función medida.m ,cuyo código se incluye a continuación:

```
function [sys,x0]=medida(t,x,u,flag)

if flag==0
sys=[2 0 3 3 0 0]  %2 est.continuos [x,h];3 entradas [u,w,theta];3salidas
x0=[-9540.6 500];
elseif flag==1
   sys(1)=(u(1)+235)*cos(u(3)*pi/180)+u(2)*sin(u(3)*pi/180);   %dx/dt
   sys(2)=(u(1)+235)*sin(u(3)*pi/180)-u(2)*cos(u(3)*pi/180);   %dh/dt
elseif flag==3
   P=[x(1) x(2) atan(x(2)/x(1))*180/pi]; %salida: [x,h,gamma_referencia]
   sys=P;
end
```

La función medida.m posee 2 estados continuos [x h], cuyas derivadas temporales vienen dadas en las ecuaciones 7.7 y 7.2 respectivamente. La función medida.m tiene: 3 entradas que son 3 de las 4 variables de estado del sistema aeronave; y 3 salidas correspondientes a x, h, y ã$_{ref}$. El cálculo de x y h se realiza pues en línea dentro de la función medida.m, por lo que no será necesario usar la función de integración integra.m, como hasta ahora se había hecho.

Con las nuevas modificaciones el modelo Simulink queda modificado como muestra la figura 9.10.

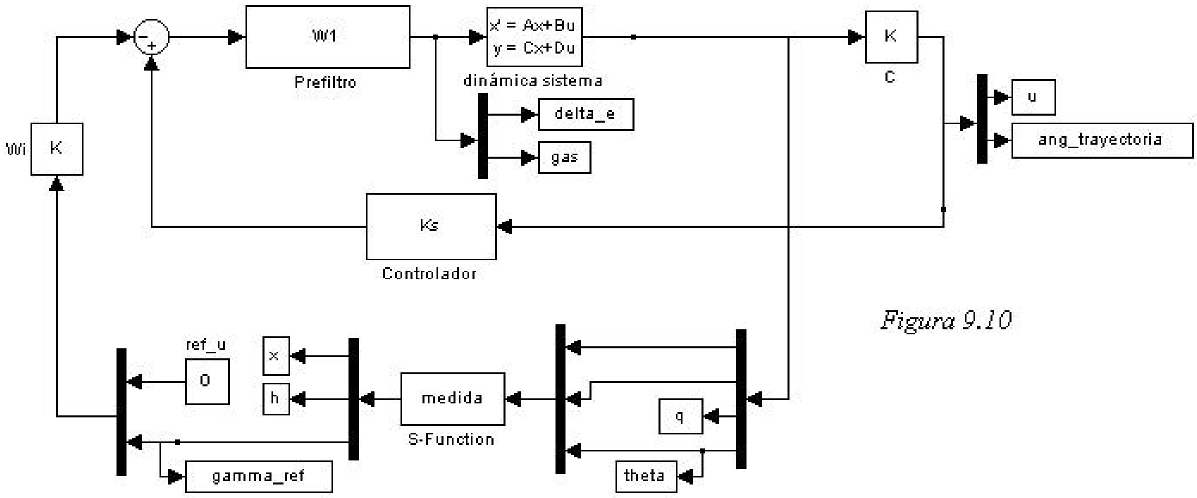


Figura 9.10

El archivo principal.m sólo ha de ser modificado eliminando las líneas para el cálculo de x y h. Tras estas modificaciones, si se ejecuta dicho archivo se obtiene el mismo valor del margen de estabilidad å=0.26 que en la simulación precedente, pues la dinámica del sistema no compensado no ha variado. Si embargo al modificarse el valor de ã$_{ref}$ (figura 9.12) la trayectoria de la aeronave se aproxima mucho más a la deseada, como se observa en la figura 9.11.

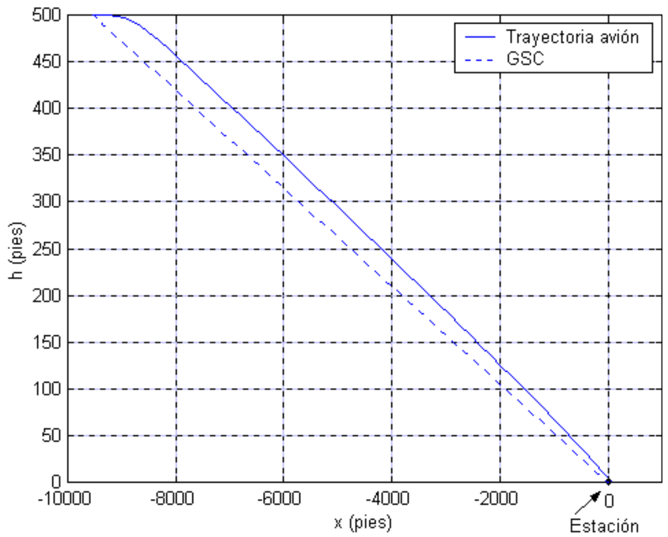


Figura 9.11

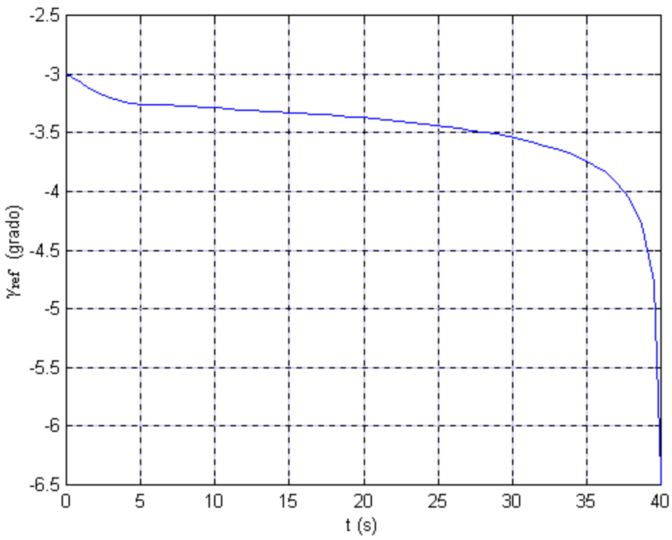


Figura 9.12

Se ha conseguido una trayectoria que converge con la GSC en la estación terrestre, sin embargo la trayectoria se separa bastante de la GSC. Para conseguir una aproximación más fiel se puede modificar la dinámica del sistema compensado G_s=GW_1 eligiendo un prefiltro W_1 adecuado. La figura 9.13 muestra con trazo discontinuo (G/s) el diagrama de Bode de una de las funciones de transferencia (G_{11}) del sistema G compensado con el prefiltro (9.7). Tras el análisis de este diagrama de Bode, se decide adoptar un nuevo filtro W_1 de la forma (9.8), consiguiéndose un sistema más rápido como se puede observar en el diagrama de Bode del sistema compensado con el nuevo filtro(figura 9.13 trazo continuo Gs).

$$W_1 = \begin{bmatrix} \frac{(10s+1)}{s} & 0 \\ 0 & \frac{(10s+1)}{s} \end{bmatrix} \qquad (9.8)$$

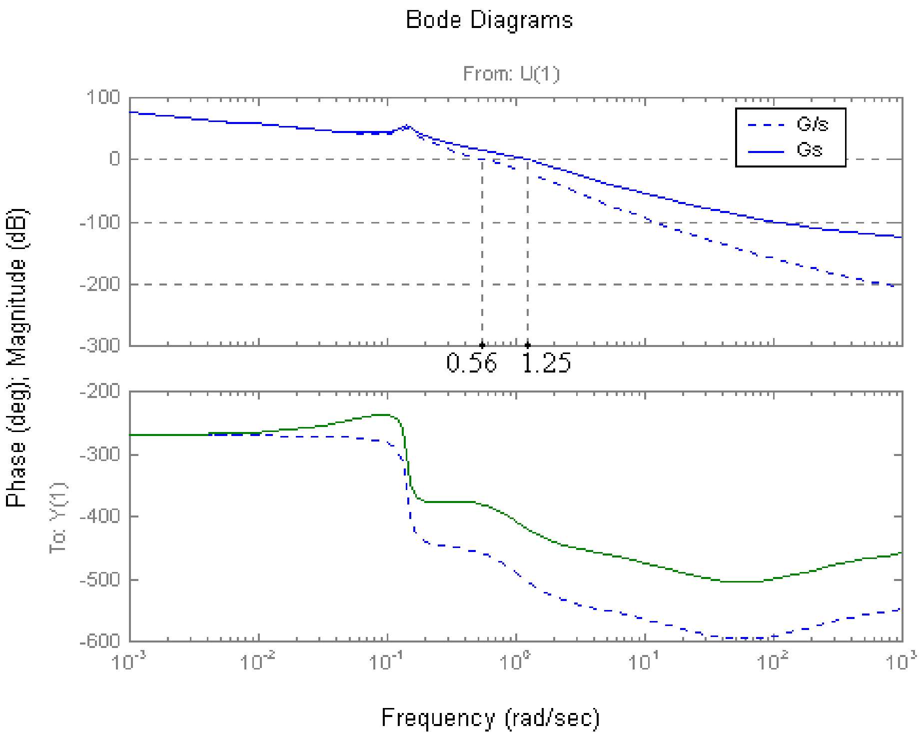


Figura 9.13

Una vez modificado el prefiltro W_1 se introduce su nueva dinámica en el archivo principal.m (apéndice A.6.3) y se ejecuta dicho archivo obteniéndose ã$_{min}$=2.17 (å$_{max}$=0.46), lo que significa un aumento del margen de estabilidad å con respecto a los casos anteriores. Por su parte la trayectoria de vuelo se aproxima mucho más a la GSC como puede observarse en la figura 9.14.

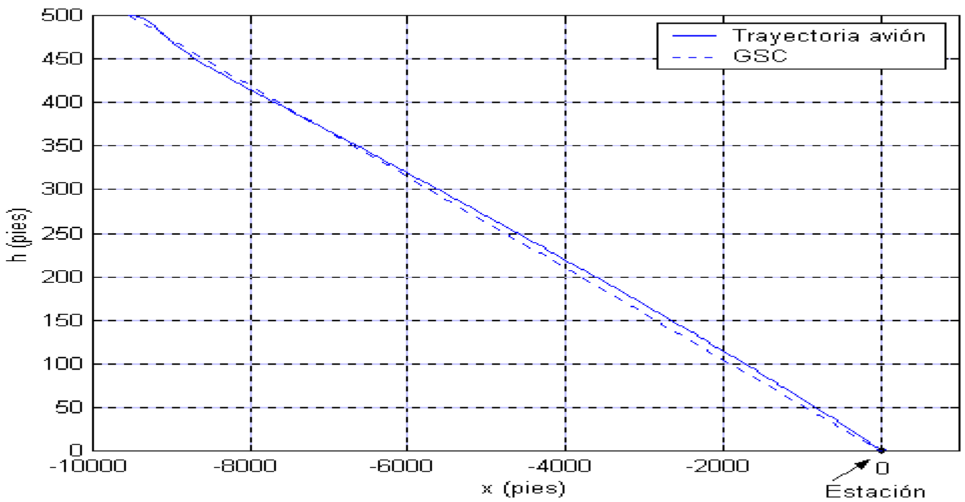


Figura 9.14

El controlador K_s y la matriz de escalado W_i resultantes se muestran en el apéndice C.1. Las figuras 9.15 a 9.20 muestran la evolución temporal de algunas variables durante la simulación. Es de especial interés observar en la figura 9.15 una velocidad de descenso de aproximadamente 12 pies/s, muy adecuada para la fase de planeo. Por otro lado se observa en la figura 9.17 como u permanece cercana a cero, lo que implica una velocidad longitudinal constante.

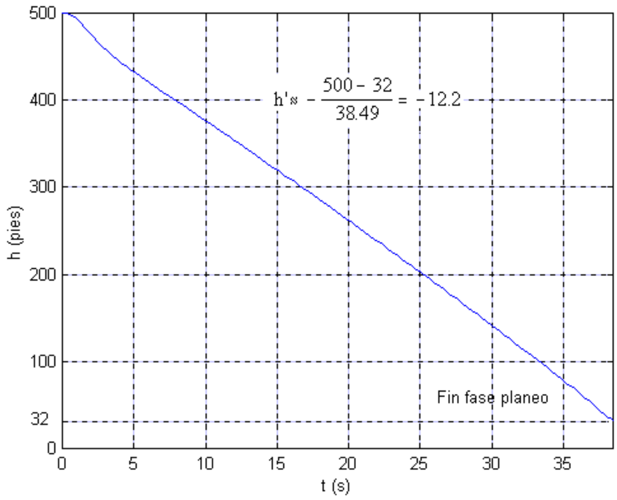


Figura 9.15

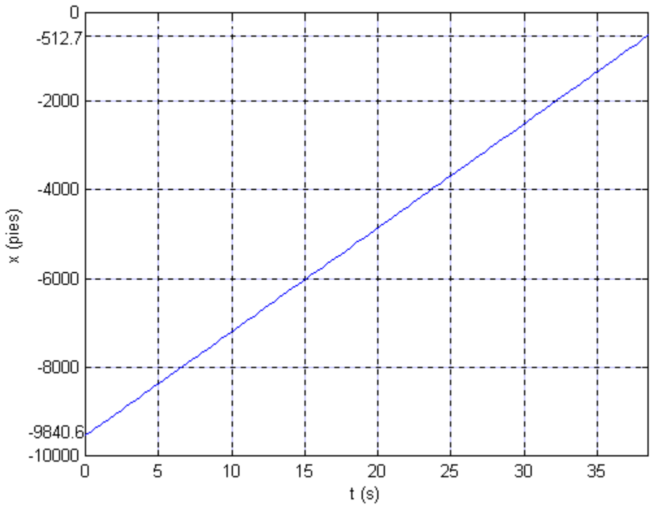


Figura 9.16

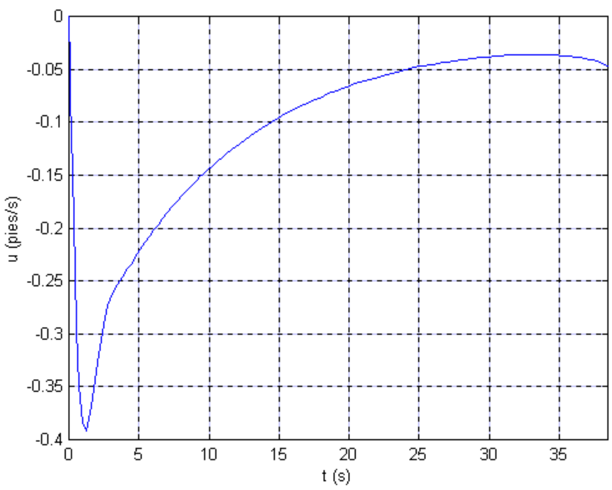


Figura 9.17

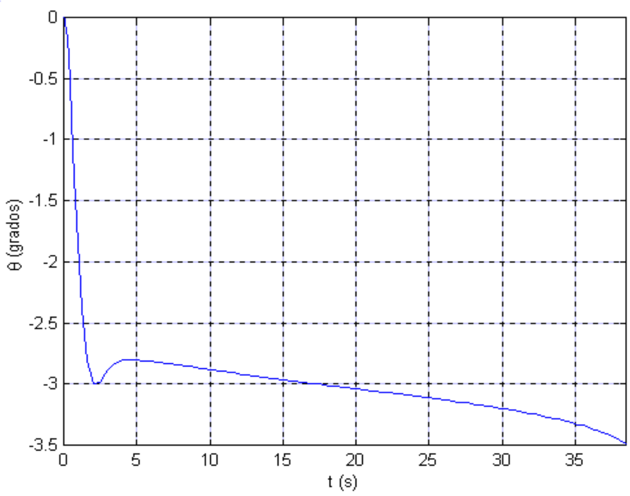


Figura 9.18

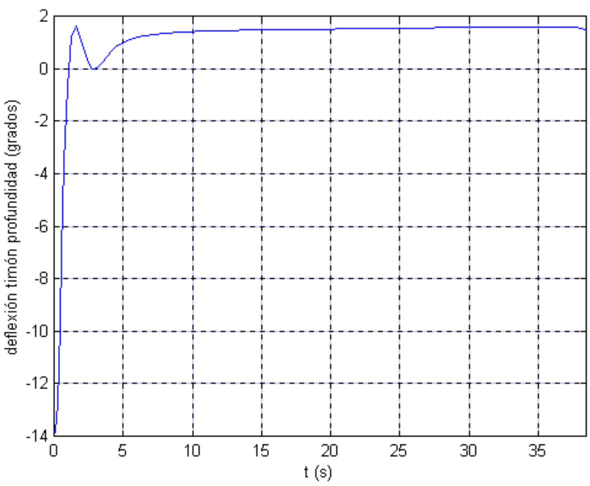


Figura 9.19

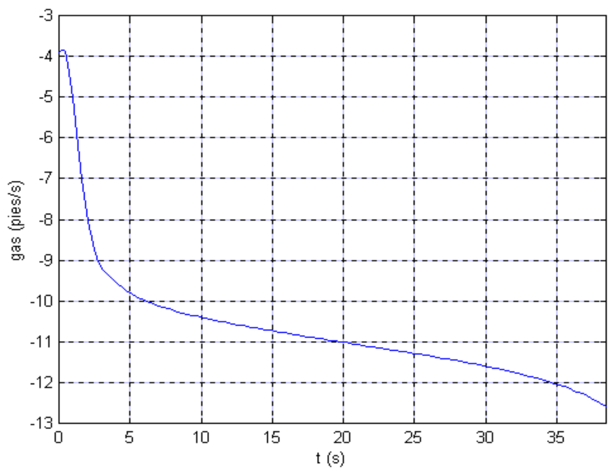


Figura 9.20

La fase de planeo terminará cuando la aeronave se encuentre a 32 pies sobre el suelo, lo que ocurre 38.49 s después de iniciada ésta. Transcurrido este tiempo se desengancha el control de planeo (glide control) dando lugar a la fase de recogida cuyo control se lleva a cabo en la siguiente sección. En (9.9) se muestran los valores de las distintas variables al finalizar la fase de planeo, que serán las condiciones iniciales para la fase de recogida.

$$(9.9)\begin{cases} h_{txon} = 32\,\text{pies} \\ x_{txon} = -512.7\,\text{pies} \\ u_{txon} = -0.0465\,\text{pies / s} \\ w_{txon} = -0.5274\,\text{pies / s} \\ q_{txon} = -0.0694\,\text{grados / s} \\ \theta_{txon} = -3.4849\,\text{grados} \end{cases} \qquad \text{txon} = 38.49\text{ s}$$

9.3.2 Fase de recogida (Flare Control)

En la fase de recogida se adoptará la dinámica del sistema dada en (7.6), con lo que se incluye en la altura en el vector de estados, el cual queda definido por:

$$\mathbf{X} = \begin{bmatrix} u & w & q & \theta & h \end{bmatrix}^T$$

Se tomará como salida del sistema la altura h, y se diseñará un control robusto H de dos grados de libertad para conseguir que h decaiga exponencialmente hasta alcanzar tierra (h=0). Por tanto, la dinámica de referencia Tref puede ser descrita por (9.10), donde se ha tomado ô=1.5 s comprobándose posteriormente los buenos resultados alcanzados.

$$h'_{ref}(t) = -\frac{1}{1.5}h_{ref}(t) \quad \leftrightarrow \quad T_{ref}(s) = \frac{1}{1.5s+1} \qquad (9.10)$$

Si se introduce como referencia en altura h_{ref} =0, el controlador K_1 del modelo de la figura 9.3 carece de efecto, siendo imposible adaptar la respuesta del sistema a T_{ref}. Para evitar esto se dará a la señal de referencia en altura el valor real de h al iniciar la fase de recogida, esto es h_{ref} = -32, y se tomará como condición de estado inicial h(0)=0 (figura 9.21, sistema auxiliar). Puesto que se está trabajando con un sistema lineal, el esfuerzo de control (actuaciones) necesario para llevar el sistema desde una altura inicial de 32 pies a una final de 0 pies es idéntico al realizado para llevar al sistema hipotéticamente desde los 0 a los –32 pies. Este resultado se ha comprobado en simulación donde se han tomado las variables medidas sobre el sistema real (con h(0)=32) como se observa en el modelo Simulink de la figura 9.21. Las demás condiciones iniciales se han obtenido de (9.9).

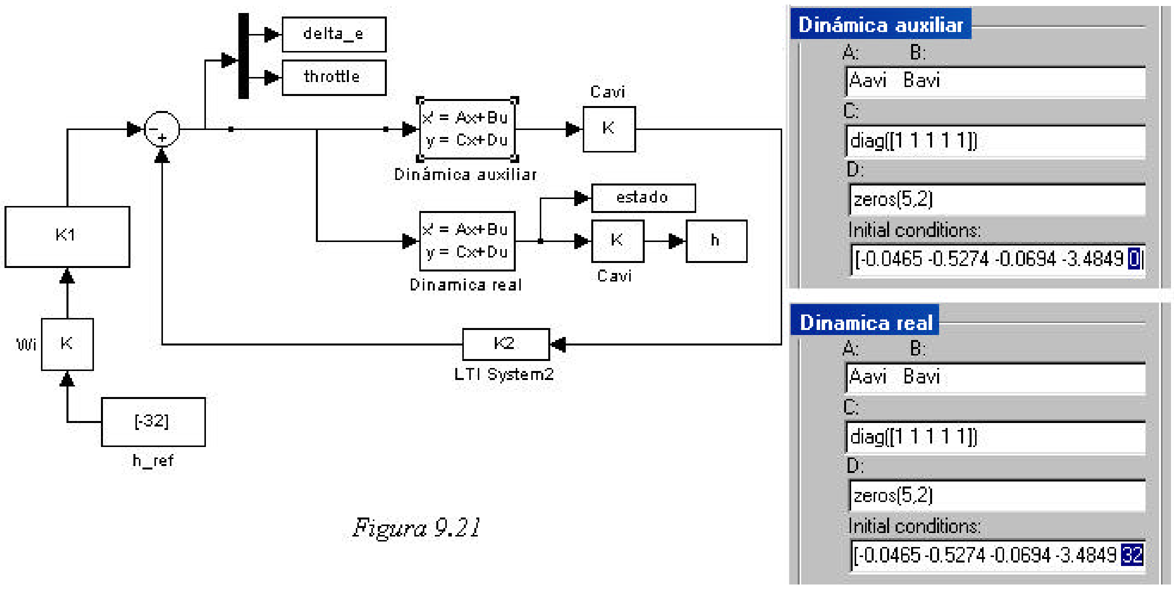


Figura 9.21

En el apéndice A.6.4 se incluye el archivo principal.m en el que se realizan las tareas indicadas en el diagrama de la figura 9.22. En dicho archivo se hace uso de dosDOFcontrol.m (apéndice A.6.2) para el cálculo del controlador de dos grados de libertad K=[K_1 K_2] y el escalado W_i. Tras la ejecución de principal.m se obtiene ã$_{min}$=4.2 (å$_{max}$=0.24), y la trayectoria mostrada en la figura 9.23. Los valores obtenidos para K_1, K_2 y W_i se muestran en el apéndice C.2.

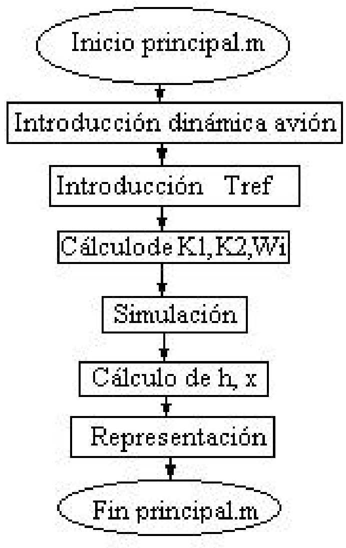


Figura 9.22

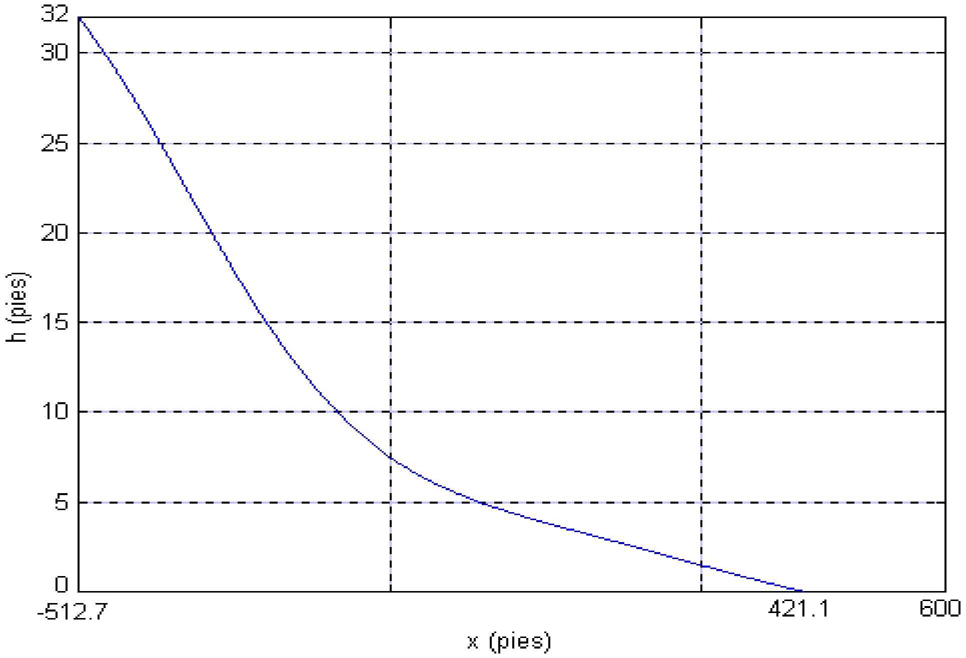


Figura 9.23

Las figuras 9.24 a 9.29 muestran los demás resultados de simulación para la fase de recogida. El avión toca tierra transcurridos 3.92 s desde el inicio de la fase de recogida, lo que hace un total de 42.41 s (t_{xon}+3.92) para la duración total del proceso de aterrizaje (fase de planeo y recogida). En la figura 9.24 se muestra con trazo continuo la evolución temporal de h, y con trazo discontinuo la evolución exponencial de referencia.

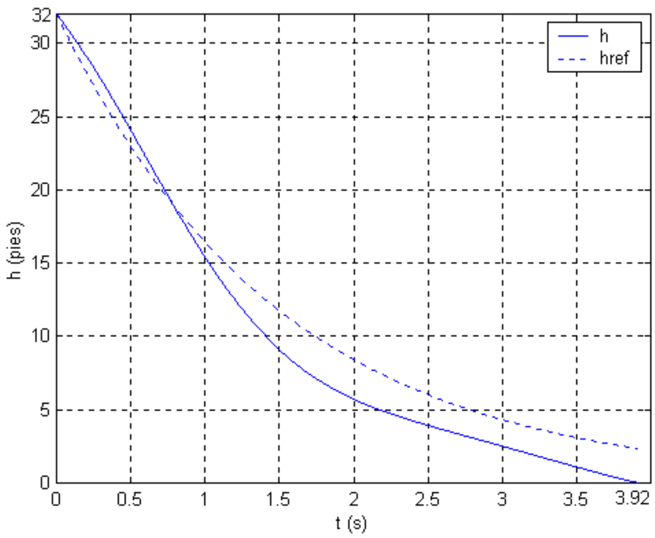


Figura 9.24

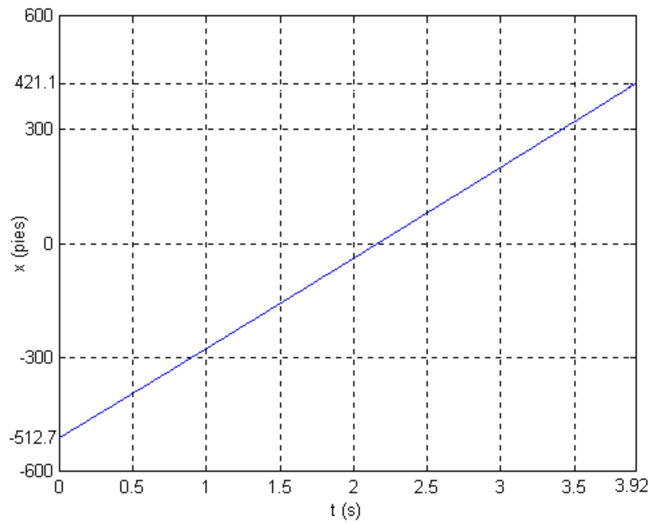


Figura 9.25

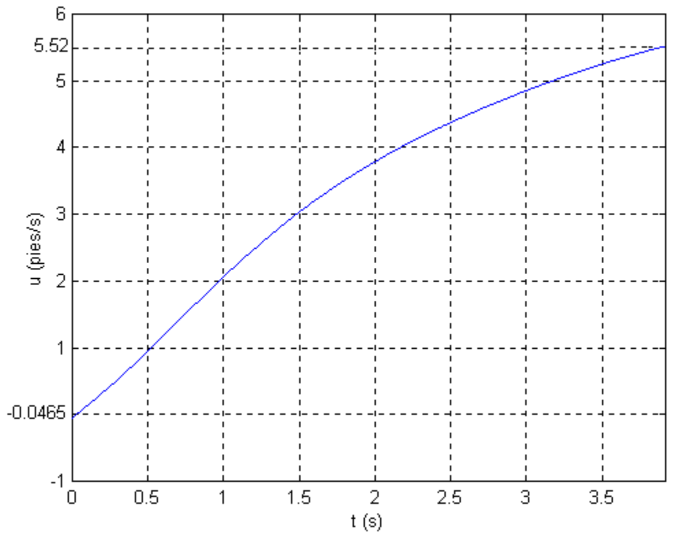


Figura 9.26

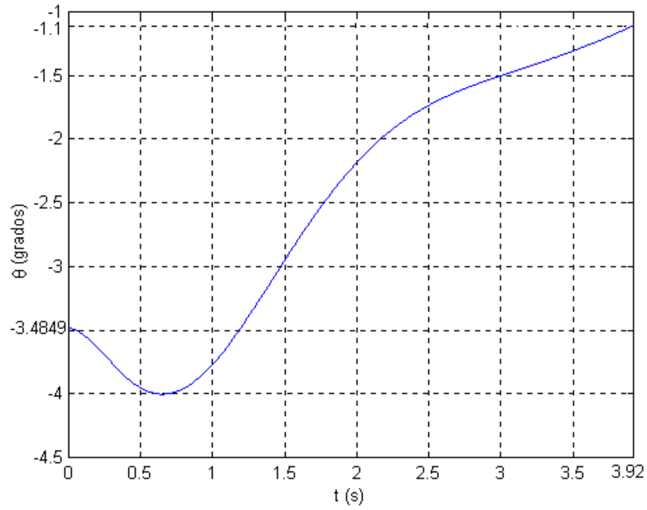


Figura 9.27

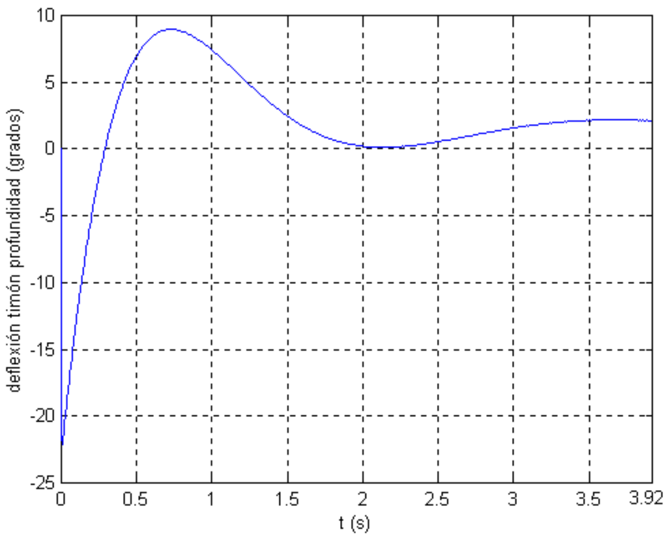


Figura 9.28

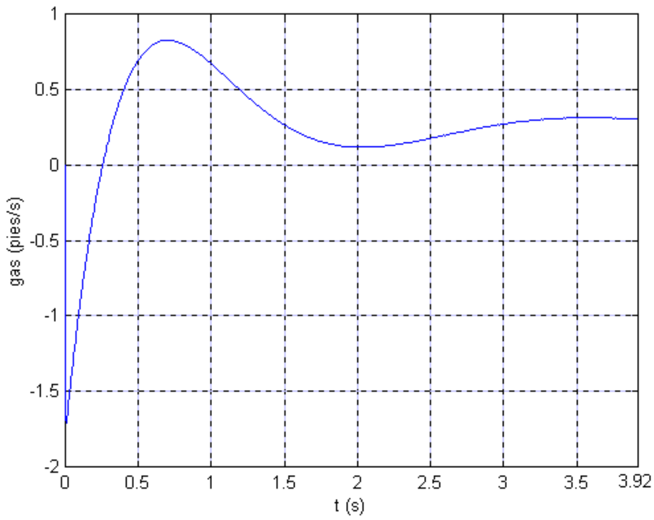


Figura 9.29

Si denota por T al instante en que la aeronave toma contacto con tierra, se pueden medir los valores dados en (9.11). Todos estos valores cumplen las condiciones para un correcto aterrizaje dadas en (7.10).

$$(xx.11)\begin{cases} \mathrm{T} = 42.41\ \mathrm{s} \\ \mathrm{x(T)} = 421.1\ \mathrm{pies} \\ \mathrm{x'(T)} \approx D\mathrm{x} / D\mathrm{t} = 240.52\ \mathrm{pies/s} \\ \mathrm{h'(T)} \approx D\mathrm{h} / D\mathrm{t} = -2.35\ \mathrm{pies/s} \\ q\,\mathrm{(T)} = -1.1^{\circ} \end{cases}$$

Capítulo 10

Conclusiones

10.1 CONTROL EN FASE CRUCERO

En esta sección se analizan las prestaciones de los distintos controladores desarrollados para la fase de vuelo crucero, poniendo de relieve el controlador LQR como el más adecuado para el sistema tratado.

10.1.1 Control LQR

El controlador LQR presenta las ventajas de estabilidad y robustez con una sencilla sintonización.

El control de la dinámica longitudinal se realizó satisfaciendo plenamente las especificaciones. Fue necesario la introducción de una red de avance adicional para disminuir las oscilaciones inducidas por el piloto.

El control de la dinámica lateral complicaba un poco la sintonización, aunque seguía obedeciendo a unas reglas empíricas muy sencillas inexistentes para el control GPC. Sobre la dinámica lateral el controlador LQR mostraba toda su potencia como control multivariable MIMO. Se consiguió una respuesta lateral bastante rápida y con poca sobreoscilación. Se observó una pequeña inversión inicial no deseada en la respuesta en derrape.

El controlador LQR presentaba recuperación completa ante perturbaciones no mantenidas, tanto si las perturbaciones eran a la entrada como a la salida. Cuando las perturbaciones se hacían mantenidas el controlador no las rechazaba plenamente.

Por último, hay que mencionar que al haberse realizado un control LQR con horizonte infinito, la eliminación total del error en régimen permanente ocurría tras un periodo de tiempo bastante grande (del orden de horas).

10.1.2 Control GPC Analítico

El controlador GPC analítico se desarrolló con dos parámetros de sintonización (horizonte de predicción y peso actuación ë), tomando el horizonte de control igual al de predicción. Aún habiendo limitado los grados de libertad, la estabilización adecuada del sistema requirió de varias simulaciones prueba-error.

En el control de la dinámica longitudinal se observó, como principal limitación, la necesidad de alejarse de la saturación de los actuadores, pues el sistema era incapaz de recuperarse de dicha saturación. Al no saturar el actuador desaparecieron las oscilaciones PIO debidas a las limitaciones de rango del actuador. Como ventaja del controlador GPC analítico o iterativo frente al control LQR de horizonte infinito, se observaba el alcance del error en régimen permanente nulo en un intervalo de tiempo bastante reducido (<10s).

Aunque cabía esperar un comportamiento del controlador GPC analítico excelente en presencia de perturbaciones, se observó la saturación del actuador ante la aparición de cualquier perturbación, ante la cual el sistema era incapaz de recuperarse.

Ante la necesidad de evitar la saturación de los actuadores se decidió desarrollar un controlador GPC con restricciones, tanto para la dinámica longitudinal como lateral.

10.1.3 Control GPC Iterativo

El controlador GPC iterativo se realizó con tres parámetros de sintonización (ë, horizonte de control, horizonte de predicción), aumentando la versatilidad con respecto al GPC analítico donde se había tomado el horizonte de control igual al de predicción.

Se desarrolló un control GPC con restricciones sobre la dinámica longitudinal con resultado positivo, observándose un rechazo completo de perturbaciones mantenidas o no, tanto a entrada como salida. El problema principal del controlador era el alto coste computacional implicado en le cálculo de la ley de control para cada intervalo de muestreo.

Por otro lado, fue imposible el diseño de un control GPC con restricciones sobre la dinámica lateral. Eso sí, partiendo del modelo general para el control GPC con restricciones y suponiendo actuadores sin limitación de rango, es decir, eliminando las restricciones correspondientes en el controlador GPC, se obtuvo una ley de control analítica (deja de ser un controlador iterativo pasando a ser un control GPC analítico sin restricciones) que proporcionaba una respuesta lateral adecuada (figura 5.7).

10.1.4 Control PID

En el capítulo 6 se intentó un control PID, aprovechando su universalidad y fácil diseño. Como resultado se obtuvo un control sobre la dinámica longitudinal incapaz de satisfaces las especificaciones de control. Por otro lado, fue imposible estabilizar la dinámica lateral con una estructura PID, pues se partió de un sistema con modo espiral divergente.

10.2 CONTROL EN FASE DE ATERRIZAJE

En los capítulos 8 y 9 se desarrollaron dos controladores distintos para el control de la fase de aterrizaje de una aeronave basándose en el proceso y dinámica de aterrizajes introducidos en el capítulo 7. Tanto el controlador basado en la teoría LQR como el basado en la teoría H dieron resultados positivos, lográndose un aterrizaje exitoso tras varias simulaciones y una elección adecuada de los parámetros de control.

Con ambos controladores se logró estabilizar el sistema, aunque cabe esperar una mayor robustez del control H frente al control LQR. Por otro lado como ya se vió en la sección 3.5, el control LQR es incapaz de rechazar perturbaciones mantenidas, mientras que el control H si puede rechazarlas.

Apéndice A

Códigos de Matlab

A.1 CONTROL LQR, FASE CRUCERO

A.1.1 Función para escalado referencia con control LQR continuo en sistemas SISO

```
function[Nbar]=rscale(A,B,C,D,K)
% Given the single-input linear system:
%       .
%      x = Ax + Bu
%      y = Cx + Du
% and the feedback matrix K,
%
% the function rscale(A,B,C,D,K) finds the scale factor N which will
% eliminate the steady-state error to a step reference
% using the schematic below:
%
%                         /---------\
%      R         +     u  | .       |
%      ---> N --->() ---->| X=Ax+Bu |--> y=Cx ---> y
%                -|       \---------/
%                 |             |
%                 |<---- K <----|
%
%8/21/96 Yanjie Sun of the University of Michigan
%        under the supervision of Prof. D. Tilbury
%
s = size(A,1);
Z = [zeros([1,s]) 1];
N = inv([A,B;C,D])*Z';
Nx = N(1:s);
Nu = N(1+s);
Nbar=Nu + K*Nx;
```

A.1.2 Control LQR continuo con dinámica longitudinal

```
                                  ---LQR_long.m---
%En este fichero se realiza la simulación de un control LQR continuo sobre el ángulo de
%cabeceo de un avión actuando sobre el timón de %profundidad (elevator).Los datos del sistema
%se han escogido de la dinámica longitudinal de un gran avión comercial propuesta en el
%"compendio de aviónica digital".


%Matrices dinámicas del avión en el eje longitudinal
format long
A=[0.0002 0.039 0 -9.81; -0.07 -0.317 250 0; 0.000088 -0.0028732
    -0.439 0; 0 0 1 0];
```

```
B=[0.44; -5.46; -1.14362; 0];
C=[0 0 0 1];                   %angulo de cabeceo
D=0;

%Analisis de estabilidad del sistema original
disp('los polos del sistema original son:');
eig(A)

%Vectores para obtener el estado completo y su uso en mod_cont.mdl, serán parámetros C y D del
%bloque dinámica longitudinal
Ce=diag([1 1 1 1]);
De=[0; 0; 0; 0];

% Elección de las matrices de ponderación cuadrática.
peso=input('introduce el peso para la matriz Q ');
Q=diag([0 0 0 peso]);
R=5;

% Resolución del problema LQR para ese criterio cuadrático.
K=lqr(A,B,Q,R);

%Analisis de estabilidad del sistema con control LQR
disp('los polos del lazo cerrado con control LQR son:');
eig(A-B*K)

%calculo del factor pa referencia no nula
Nbar=rscale(A,B,C,0,K);

%a continuacion se simulará el control lqr
sim('mod_cont');
```

Graficas

A.1.3 Representación resultados con dinámica longitudinal

```
% Representación del ángulo de cabeceo y su valor de referencia
subplot(2,1,1),plot(tout,ref_cabeceo.signals.values,'r');
legend('angulo cabeceo',4);
hold on;
subplot(2,1,1),plot(tout,cabeceo.signals.values,'r');
grid

% Representación de la señal de control ang.deflexion timón de %profundidad (elevator)
subplot(2,1,2),plot(tout,delta_e.signals.values);
legend('defleción timon profundidad',4);
grid
```

A.1.4 Control LQR discreto con dinámica longitudinal

```
                               ---LQR_long_d---
%En este fichero se realiza la simulación de un control LQR discreto sobre el ángulo de
%cabeceo de un avión actuando sobre el timón de %profundidad (elevator).Los datos del sistema
%se han escogido de la dinámica longitudinal de un gran avión comercial propuesta en el
%"compendio de aviónica digital".

%Matrices dinámicas del avión en el eje longitudinal en t.continuo
format long
A=[0.0002 0.039 0 -9.81; -0.07 -0.317 250 0; 0.000088 -0.0028732 -0.439 0; 0 0 1 0];
B=[0.44; -5.46; -1.14362; 0];
C=[0 0 0 1];                   %angulo de cabeceo
D=0;

%Vectores para obtener el estado completo y su uso en mod_cont.mdl, serán parámetros C y D del
%bloque dinámica longitudinal
He=diag([1 1 1 1]);
Je=[0; 0; 0; 0];

%Determinación del tiempo de muestreo empíricamente a partir del BW en bucle cerrado
sist_cont=ss(A,B,C,D);
```

```
BW=bww(sist_cont);
Tm=1/(60*BW);        %1/2 de lo aconsejado por regla empírica
                      %obtenida del tutorial de Matlab (univ.Michigan)

%Se pasa a la representación en el espacio de estados discreta usando %un mantenedor orden 0
[F,G,H,J] = c2dm (A,B,C,D,Tm,'zoh');

%Analisis de la controlabilidad y observabilidad del sistema
co = ctrb (F,G);           %construye matriz de controlabilidad
ob = obsv (F,H);              %construye matriz de observabilidad

Controlabilidad = rank (co);
Observabilidad = rank (ob);

if((Controlabilidad==length(F))&(Observabilidad==length(F)))

       % Elección de las matrices de ponderación cuadrática.
       peso=input('introduce el peso para la matriz Q ');
       Q=diag([0 0 0 peso]);
       R=5;

       % Resolución del problema LQR para ese criterio cuadrático en
       %tiempo discreto.
       [K] = dlqr(F,G,Q,R);

       %calculo del factor pa referencia no nula
      Nbar=mirscale_discreto(F,G,K);

       %Se piden caracteristicas de la respuesta deseada
       error=input('introduce el error deseado en reg.permanente (será
                      en % referencia)');
      t_est=input('introduce el tiempo establecimiento deseado');

      %a continuacion se simulara el control lqr
      sim('mod_disc',2*t_est)

      tend=length(ref_cabeceo.signals.values);
      tmedio=tend/2-mod(tend/2,1);    %tmedio=parte entera de(tend/2)

       if(abs(ref_cabeceo.signals.values(tmedio)-
           cabeceo.signals.values(tmedio))<ref_cabeceo.signals.values(tmedio)*error/100)
       graficas;
       else
      disp('debe relajar especificaciones o modificar matrices
               ponderación del LQR')
       end

else
       disp('el sistema no es observable y/o controlable totalmente');
end
```

A.1.5 Cálculo frecuencia de 3dB

```
%Esta función recibe un sistema y devuelve la frecuencia de ancho de banda del lazo cerrado en
%rad/seg. Esta función no es muy precisa, pero no importa, pues se aprovechará posteriormente
%para determinar un tiempo de muestreo siguiendo una regla empirica (y a cambio se gana
%velocidad de computo).

function w=bww(sistema)

[num,den]=tfdata(sistema,'v');
[numc,denc]=cloop(num,den,-1);
sist_c=tf(numc,denc);

fin=0;
w=0.1;
while(fin==0)
      if(abs(20*log10(norm(freqresp(sist_c,w)))+3)<0.5)
        fin=1;
      else
        w=w+0.05;
    end
end
```

A.1.6 Control LQR discreto con dinámica lateral

```
%En este fichero se realiza la simulación de un control LQR discreto sobre los ángulos de
%derrape y alabeo de un avión actuando sobre los %alerones (ailerons) y sobre el timón de de
%dirección (rudder).Los datos del sistema se han escogido de la dinámica lateral de un
%gran avión comercial propuesta en el "compendio de aviónica digital".

%Matrices dinámica lateral/direccional del avión
A=[-0.056 0 -1 0.03924; -1.05 -0.47 0.39 0; 0.6 -0.032 -0.115 0; 0 1 0 0];
B=[0 0.012; 0.14 0.15; 0.008 -0.48; 0 0]; %actuaciones delta_a delta_r
C=[1 0 0 0; 0 0 0 1];                        %salidas ang. derrape alabeo
D=[0 0; 0 0];


%Vectores para la obtención del estado completo y usar en mod_disc.mdl
He=diag([1 1 1 1]);
Je=[0 0; 0 0; 0 0; 0 0];

%Determinación del tiempo de muestreo empíricamente
sist_cont=ss(A,B,C,D);
BW=2;                         %gráficamente al ejecutar  "multibode.m"
Tm=1/(60*BW);           %regla empírica

%Se pasa a la representacion en el espacio de estados discreta usando %un mantenedor orden 0
[F,G,H,J] = c2dm (A,B,C,D,Tm,'zoh');

%Analisis de estabilidad del sistema original
disp('los polos del sistema original en t.discreto son:');
polos=eig(F)
disp('y sus modulos');
norm(polos(1)), norm(polos(2)), norm(polos(3)), norm(polos(4))

%Analisis de la controlabilidad y observabilidad del sistema
co = ctrb (F,G);           %construye matriz de controlabilidad
ob = obsv (F,H);              %construye matriz de observabilidad

Controlabilidad = rank (co);
Observabilidad = rank (ob);

if((Controlabilidad==length(F))&(Observabilidad==length(F)))
       % Elección de las matrices de ponderación cuadrática.
       Q=diag([80 0 0 80]);
      peso_a=input('introduce el peso para la actuacion alerones ');
       peso_r=input('introduce el peso para la actuacion timón
                                                                   dirección ');
      R=diag([peso_a peso_r]);

       % Resolución del problema LQR para ese criterio cuadrático en
       % tiempo discreto.
       [K] = dlqr(F,G,Q,R);

      %Analisis de estabilidad del sistema original
       disp('los polos del sistema con control LQR discreto son:');
       polos=eig(F-G*K)
      disp('y sus modulos');
       norm(polos(1)), norm(polos(2)), norm(polos(3)), norm(polos(4))

      %Se pide la referencia a seguir al usuario
       ref_derrape=input('introduce el ángulo de derrape deseado (serán
       ref_alabeo=input('introduce el ángulo de alabeo deseado (serán
                                                                       radianes)');

      %Se piden las caracteristicas de la respuesta deseada
       error=input('introduce el error deseado en reg.permanente (será
                                                               en % referencia)');
      t_est=input('introduce el tiempo establecimiento deseado');


       %calculo del factor pa referencia no nula
      matriz_escala=mirscale_discreto(F,G,K,ref_derrape,ref_alabeo);

      %a continuacion se simulara el control lqr
      sim('mod_disc',2*t_est)
```

```
      tend=length(ref_derrape.signals.values);
      tmedio=tend/2-mod(tend/2,1);              %parte entera (tend/2)

        if((abs(ref_alabeo.signals.values(tmedio)-
                    alabeo.signals.values(tmedio))< error*ref_alabeo.signals.values(1)/100) ...
            &(abs(ref_derrape.signals.values(tmedio)-
                    derrape.signals.values(tmedio))<
            error*ref_derrape.signals.values(1)/100))

              Graficas;
      else
             disp('Debe relajar especificaciones o modificar matrices
                                                                      LQR')
        end
else
        disp('el sistema no es observable y/o controlable totalmente');
end
```

A.1.7 Representación resultados con dinámica lateral

```
% Representación de ángulo de alabeo, derrape y referencias
subplot(3,1,1),plot(tout,ref_alabeo.signals.values,'r');
hold on;
subplot(3,1,1),plot(tout,ref_derrape.signals.values,'b');
subplot(3,1,1),plot(tout,alabeo.signals.values,'r');
subplot(3,1,1),plot(tout,derrape.signals.values,'b');
title('ángulo de derrape (azul) y ángulo de alabeo (rojo)')
grid

% Representación de las señales de control ang.deflexion alerones %timon direccion (rudder)
subplot(3,1,2),plot(tout,delta_a.signals.values);
legend('deflexión alerones',4);
grid
subplot(3,1,3),plot(tout,delta_r.signals.values);
legend('deflexión timon direccion',4);
grid
```

A.1.8 Función para escalado referencia con control LQR discreto con dinámica lateral

```
                       ---mirscale_lateral.m---
%Esta función calcula la matriz de escala a aplicar a la referencia bidimensional formada por
%el ángulo de derrape y el de alabeo para un %control LQR (con vector ganancia K).Se considera
%una dinámica lateral %con dos actuaciones,2 salidas (angulo derrape y alabeo) y
%4 estados --> A(4x4);B(4x2);K(4x2);C=[1 0 0 0;0 0 0 1].
%ref1 es la referencia en derrape, ref2 es la referencia en alabeo.
%La matriz devuelta se aplicará en el modelo Simulink (mod_cont.mdl).

function salida=mirscale(A,B,K,ref1,ref2);

if((ref2==0)&(ref1==0))

   salida=[0 0; 0 0];

elseif(ref1==0)    %se trata de controlar pues el ángulo de alabeo.

S=solve('A(1,2)*x+A(1,3)*y+A(1,4)*ref2+(B(1,1)*N+B(1,2)*M)*ref2=B(1,1)*(K(1,2)*x+K(1,3)*y+K(1,
4)*ref2)+B(1,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*ref2)',...

'A(2,2)*x+A(2,3)*y+A(2,4)*ref2+(B(2,1)*N+B(2,2)*M)*ref2=x+B(2,1)*(K(1,2)*x+K(1,3)*y+K(1,4)*ref
2)+B(2,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*ref2)',...

'A(3,2)*x+A(3,3)*y+A(3,4)*ref2+(B(3,1)*N+B(3,2)*M)*ref2=y+B(3,1)*(K(1,2)*x+K(1,3)*y+K(1,4)*ref
2)+B(3,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*ref2)',...

'A(4,2)*x+A(4,3)*y+A(4,4)*ref2+(B(4,1)*N+B(4,2)*M)*ref2=ref2+B(4,1)*(K(1,2)*x+K(1,3)*y+K(1,4)*
ref2)+B(4,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*ref2)','M,N,x,y');
```

```
    salida=[0 eval(S.N); 0 eval(S.M)];

elseif(ref2==0)     %se trata de controlar el ángulo de derrape.

S=solve('A(1,1)*ref1+A(1,2)*x+A(1,3)*y+(B(1,1)*N+B(1,2)*M)*ref1=ref1+B(1,1)*(K(1,1)*ref1+K(1,2
)*x+K(1,3)*y)+B(1,2)*(K(2,1)*ref1+K(2,2)*x+K(2,3)*y)',...

'A(2,1)*ref1+A(2,2)*x+A(2,3)*y+(B(2,1)*N+B(2,2)*M)*ref1=x+B(2,1)*(K(1,1)*ref1+K(1,2)*x+K(1,3)*
y)+B(2,2)*(K(2,1)*ref1+K(2,2)*x+K(2,3)*y)',...

'A(3,1)*ref1+A(3,2)*x+A(3,3)*y+(B(3,1)*N+B(3,2)*M)*ref1=y+B(3,1)*(K(1,1)*ref1+K(1,2)*x+K(1,3)*
y)+B(3,2)*(K(2,1)*ref1+K(2,2)*x+K(2,3)*y)',...

'A(4,1)*ref1+A(4,2)*x+A(4,3)*y+(B(4,1)*N+B(4,2)*M)*ref1=B(4,1)*(K(1,1)*ref1+K(1,2)*x+K(1,3)*y)
+B(4,2)*(K(2,1)*ref1+K(2,2)*x+K(2,3)*y)','M,N,x,y');

    salida=[eval(S.N) 0; eval(S.M) 0];

else

z=ref2/ref1;

S=solve('A(1,1)+A(1,2)*x+A(1,3)*y+A(1,4)*z+B(1,1)*N+B(1,2)*M*z=1+B(1,1)*(K(1,1)+K(1,2)*x+K(1,3
)*y+K(1,4)*z)+B(1,2)*(K(2,1)+K(2,2)*x+K(2,3)*y+K(2,4)*z)',...

'A(2,1)+A(2,2)*x+A(2,3)*y+A(2,4)*z+B(2,1)*N+B(2,2)*M*z=x+B(2,1)*(K(1,1)+K(1,2)*x+K(1,3)*y+K(1,
4)*z)+B(2,2)*(K(2,1)+K(2,2)*x+K(2,3)*y+K(2,4)*z)',...

'A(3,1)+A(3,2)*x+A(3,3)*y+A(3,4)*z+B(3,1)*N+B(3,2)*M*z=y+B(3,1)*(K(1,1)+K(1,2)*x+K(1,3)*y+K(1,
4)*z)+B(3,2)*(K(2,1)+K(2,2)*x+K(2,3)*y+K(2,4)*z)',...

'A(4,1)+A(4,2)*x+A(4,3)*y+A(4,4)*z+B(4,1)*N+B(4,2)*M*z=z+B(4,1)*(K(1,1)+K(1,2)*x+K(1,3)*y+K(1,
4)*z)+B(4,2)*(K(2,1)+K(2,2)*x+K(2,3)*y+K(2,4)*z)','M,N,x,y');

   salida=[eval(S.N) 0; 0 eval(S.M)];

end
```

A.2 CONTROL GPC ANALÍTICO

A.2.1 subrutina gpc.m

```
%                      --- gpc.m ---
%subrutina para el cálculo de la predicción óptima GPC

function [g,gp,f,fb]=gpc(a,b,d,n);
%[g,gp,f,fb]=gpc(den,num,d,N);
%den, num son el denominador y numerador de la función de %transferencia del sistema a
controlar

g=zeros(n,n);
a=[a,0]-[0,a];
na=max(size(a));
nb=max(size(b));
f=zeros(n+d,na-1);
gm=zeros(n+d,d+n+nb-1);
f(1,:)=-a(2:na);
e=1;
gm(1,1:nb)=b;
for j=2:n+d
e=[e,f(j-1,1)];
if na > 2
f(j,:)=[f(j-1,2:na-1),0]-f(j-1,1)* a(2:na);
else
f(j,:)=-f(j-1,1)* a(2);
end;
gm(j,1:j+nb-1)=conv(e,b);
end;
for j=1:n
g(j,1:j)=gm(j+d,j:-1:1);
if (d+nb) > 1
gp(j,:)=gm(j+d,j+1:1:d+nb+j-1);
else
```

```
gp=0;
end;
fb(j,:)=f(d+j,1:1:na-1);
end;
```

A.2.2 Controlador GPC con dinámica longitudinal pura del avión

```
%                      ---principal.m---
%Calculo del controlador GPC basado en función de transferencia

%Dinámica longitudinal discreta del avión (dada en (2.44))
format long;
a1=-3.994;
a2=5.982;
a3=-3.982;
a4=0.994;
b0=-3.655e-5;
b1=3.651e-5;
b2=3.646e-5;
b3=-3.642e-5;
num=[b0 b1 b2 b3];
den=[1 a1 a2 a3 a4];

%Introducción de los parámetros del GPC
numero=input('introduce un valor que se adjudicara a N2,Nu y N');
lambda=input('introduce un valor para lambda');

N1=1;              %solo a título informativo se presupone
N2=numero;     %solo a título informativo se presupone
Nu=numero;     %solo a título informativo se presupone
N=numero;
d=0;

%Llamada a la función que realiza la predicción.
[g,gp,f,fb]=gpc(den,num,d,N);

%Cálculo vectores para construcción señal de control
h=-inv(g'*g+lambda*eye(N))*g';
kr=-h(1,:);
ku=h*gp;
ku=ku(1,:);
ky=h*fb;
ky=ky(1,:);

%Simulación del control GPC
sim('mod_suave');

%representación de la salida (ángulo de cabeceo)
subplot(2,1,1),plot(tout,salida.signals.values),hold on;

%represención de la señal de control (la deflexion del %t.profundidad)
subplot(2,1,2),plot(tout,control.signals.values),hold on;
```

A.2.3 Controlador GPC con dinámica longitudinal incluyendo actuador

```
%                      ---principal.m---
%Calculo del controlador GPC considerando actuador real

%Matrices dinámicas del avión en el eje longitudinal
format long
A=[0.0002 0.039 0 -9.81; -0.07 -0.317 250 0; 0.000088 -0.0028732 -0.439 0; 0 0 1 0];
B=[0.44; -5.46; -1.14362; 0];
C=[0 0 0 1];                    %ángulo de cabeceo
D=0;

sistema=ss(A,B,C,D);
sistema_tf=tf(sistema);
sistemad=c2d(sistema_tf,0.008);
```

```
%dinámica actuador
actuador_cont=tf([37],[1 37]);
actuador_disc=c2d(actuador_cont,0.008);

%dinámica total
sistema_total=sistemad*actuador_disc;
[num,den]=tfdata(sistema_total,'v');
num=num(2:6);                %eliminar un cero de la izquierda

%se piden parámetros para GPC. En adelante igual que sin actuador
numero=input('introduce un valor que se adjudicara a N2,Nu y N');
lambda=input('introduce un valor para lambda');

N1=1;              %solo a título informativo se presupone
N2=numero;     %solo a título informativo se presupone
Nu=numero;     %solo a título informativo se presupone
N=numero;
d=0;

%Llamada a la función que realiza la predicción.
[g,gp,f,fb]=gpc(den,num,d,N);

%Cálculo vectores para construcción señal de control
h=-inv(g'*g+lambda*eye(N))*g';
kr=-h(1,:);
ku=h*gp;
ku=ku(1,:);
ky=h*fb;
ky=ky(1,:);

%Simulación del control GPC
sim('mod_suave');

%representación de la salida (ángulo de cabeceo)
subplot(2,1,1),plot(tout,salida.signals.values),hold on;

%representación de la señal de control (deflexion del %t.profundidad)
subplot(2,1,2),plot(tout,control.signals.values),hold on;
```

A.3 CONTROL GPC ITERATIVO

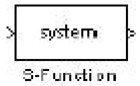
A.3.1 Descripción de las S-functions

Implementación de S-Functions en Simulink

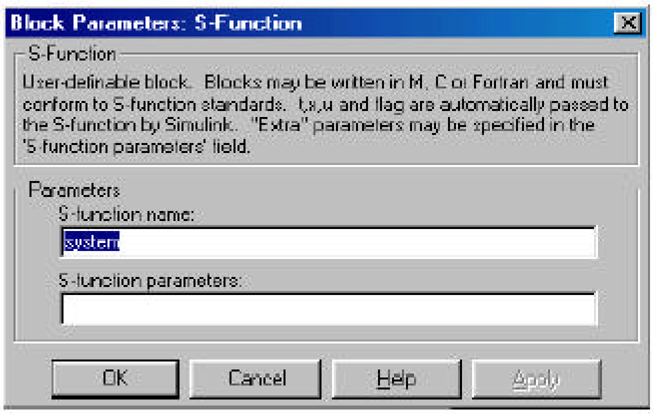
Simulink es una poderosa herramienta que permite, a través de modelos en forma de diagrama de bloques, simular y analizar prácticamente cualquier sistema dinámico sea lineal o no. Dichos modelos pueden ser de naturaleza continua, discreta o mixta y para construirlos contamos con librerías dotadas de diferentes bloques como ser: bloques no-lineales, fuentes de señal, funciones transferencias (FT's), etc. El usuario tiene también la posibilidad de definir bloques propios, lo cual será tema del presente trabajo. Otra de las virtudes de Simulink es que al estar construido sobre Matlab, permite procesar los resultados de las simulaciones con todo su poder de cálculo. La importancia de esto reside en las llamadas "toolboxes", las cuales agrupan varias funciones que permiten atacar diversos problemas en forma directa. Estas librerías no solo representan un conjunto de funciones sino el gran esfuerzo de los más reconocidos investigadores en las diferentes áreas del control.

¿Qué son las S-Functions ?

Las llamadas S-Functions (Funciones de Sistemas) extienden la capacidad de simulación en Simulink. Estas funciones permiten incluir algoritmos propios en modelos Simulink y pueden estar escritos en lenguaje C o en lenguaje Matlab, con la salvedad de que deben poseer una estructura determinada. Se podría decir que una S-Function es un algoritmo que describe un sistema dinámico determinado. Para incorporarlas a los modelos Simulink desarrollados en este informe, se debe utilizar el bloque homónimo que se encuentra en una de las librerías.



Dicho bloque posee una ventana de parámetros en la cual se puede especificar el nombre de la S-function (nombre del archivo) asociada a este bloque. Este es el nombre del archivo .m ó .c que contiene el código a ejecutar:



El bloque S-Function permite además el paso de parámetros al algoritmo lo cual facilita la creación de Máscaras; una facilidad de Simulink para el ingreso eficiente de parámetros al subsistema. En general, deberían utilizarse S-Functions en algunos de los siguientes casos:

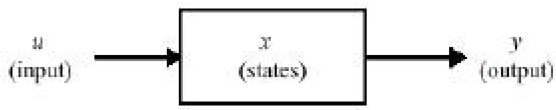
- Animaciones Gráficas
- Descripción matemática de sistemas
- Creación de bloques específicos

Funcionamiento de las S-Functions

En Simulink, cada bloque posee:

- Un vector de estados (x)
- Un vector de entrada (u)
- Un vector de salida (y)

Como se puede observar en la siguiente figura:



Cada uno de estos bloques representa un sistema físico, el cual posee una descripción matemática en la forma matricial de Ecuaciones de Estado. Esto es:

$$\dot{x} = A.x + B.u$$
$$y = C.x + D.u$$

Estos bloques pueden ser de naturaleza discreta, continua o una combinación de ambas.En la simulación, Simulink hace repetidas llamadas en ciertos instantes a cada uno de los bloques que componen el sistema con el fin de actualizar estados, calcular salidas, etc. Llamadas adicionales se efectúan al principio y al final de la simulación con el objeto de realizar tareas de inicialización y finalización de la simulación.

En cada simulación, se inicializa el sistema. Esto significa inicializar cada bloque que lo compone como así también las S-Functions. Luego, se ingresa al lazo de simulación durante el cual cada paso es conocido como paso de simulación. En cada paso de simulación, Simulink ejecuta el bloque SFunction. El ciclo continua hasta que la simulación se completa. Si el sistema no posee estados, el ciclo de ejecución se reduce al cálculo de salidas e inicialización. Las llamadas efectuadas por Simulink a las S-Functions resuelven diversas tareas como ser:

Inicialización: Previo a el primer ciclo de simulación, Simulink inicializa la S-Function. En esta etapa se ajusta el número de entradas y salidas, el período de muestreo, el espacio de memoria reservado y además se calculan algunas constantes que permanecen invariables en el resto de la simulación.

Calculo de salidas: Al final de esta etapa son válidas, durante el timestep en cuestión, las salidas del bloque.

Todas estas subrutinas se deben ejecutar en momentos determinados de la simulación, para lo cual se implementan como subrutinas de la S-Function. Para determinar en que etapa del ciclo de simulación se encuentra, Simulink pasa un “flag” a la función de sistema mediante el cual se llama a la subrutina adecuada. El usuario debe tener en cuenta que los valores que debe retornar cada subrutina varían con cada “flag” que pasa Simulink.

La siguiente tabla muestra cómo nombrar las subrutinas y el “flag” asociado a cada una de ellas:

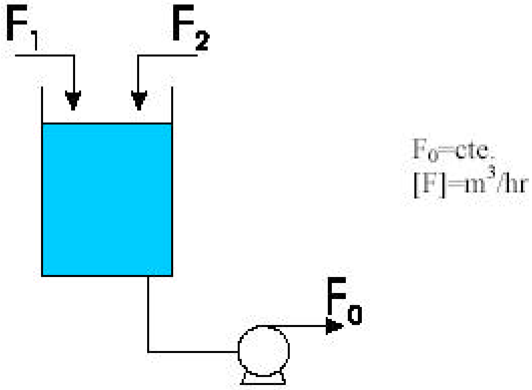
Etapa de Simulación	*Rutina*	*Flag*
Inicialización	MdlInitializaSizes	0
Cálculo de las Salidas	MdlOutputs	3
Actualizar Estados Discretos	MdlUpdate	2
Cálculo de Derivadas	MdlDerivatives	1
Fin de Simulación	MdlTerminate	9

El nombre de la Rutina asociada a cada “flag” puede variar quedando a libertad del autor siempre y cuando se respeten los “flags”. Sin embargo, es recomendable tomar el “template” que ofrece Matlab de manera de evitar errores en cuanto a los argumentos que recibe cada subrutina y los valores que se deben retornar.

Las S-Functions permiten además combinar sistemas discretos y continuos, tomar tiempos de muestreo arbitrarios, etc.

Ejemplo sencillo de uso de las S-Functions con sistema en tiempo continuo

Lo que se hará en este ejemplo es calcular la variación con respecto al tiempo de la altura en un tanque, en el cual hay dos flujos de entrada y un flujo de salida, el cual se mantiene constante por una bomba. El sistema está descripto en el siguiente diagrama:

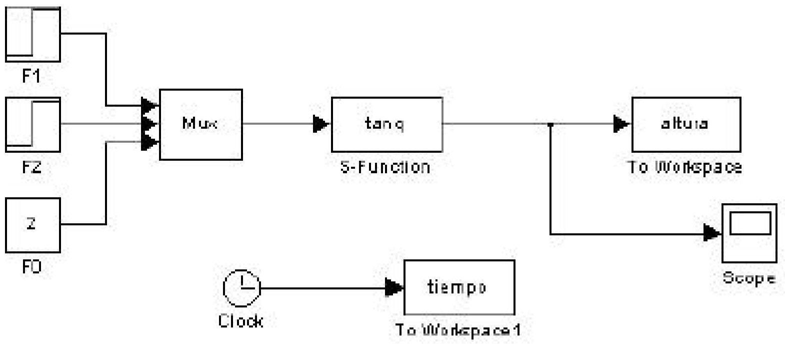


La variación de volumen en un instante de tiempo, en forma discreta, es: (V/ . t)=F1(t) + F2(t) – F0

Que es igual a la variación de la altura multiplicado por el área: A(h/ t), ya que el área no varía. O en forma diferencial: A(dh/dt)=F1(t) + F2(t) – F0.

Se simulará con una s-function la evolución de la altura en el tanque a medida que se producen cambios en los caudales F1 y F2.

Para resolver este problema se utiliza la toolbox de MatLab llamada Simulink, donde se crea el siguiente sistema:



Donde:

• en F1, F2 y F0 sirven para definir los flujos de entrada y salida. En este caso los flujos de entrada poseen una variación tipo escalón en un cierto tiempo: F1: comienza con el valor de 1 y cambia al valor de 1,7 a los 12 segundos F2: comienza con el valor de 1 y cambia al valor de 1,3 a los 7 segundos El valor de F0 se mantiene constante a lo largo del tiempo, su valor es de 2 m^3/hr

• en tanq se define cual es el nombre de la s-function que se usará, en la cual se determinan la ecuación diferencial y algunos valores, como la altura inicial, altura máxima(altura del tanque), área del tanque, etc.

• en altura se define cual es el valor de la variable que interesa (la variable de salida)

• con clock se determina la variación del tiempo, el cual es guardado en la variable tiempo

• y con scope, se grafica la variable analizada(altura) en función del tiempo

La s-function utilizada es la siguiente:

```
tanq.m :

function [sys,x0]=tanq(t,x,u,flag)

A=2

if flag==0
   sys=[1 0 1 3 0 0]
   x0=.3
elseif flag==1
   sys=(u(1)+u(2)-u(3))/A
elseif flag==3
   if x>0 & x<4
      sys(1)=x(1)
   elseif x>4
      sys=4
   else
      sys=0
   end
else
   sys=[]
end
end
end
```

En su sintaxis se observa que dado un tiempo(t), una variable a analizar(x), en este caso la altura, ciertas variables determinadas (u), F1, F2 y F0, y cierta variable flag, la cual comienza con el valor 0 y va aumentando a medida que avanza el programa, se obtienen las variables sys y x0, con las cuales se calculan los valores de la altura en cada momento. Cuando flag es igual a 0, se define:

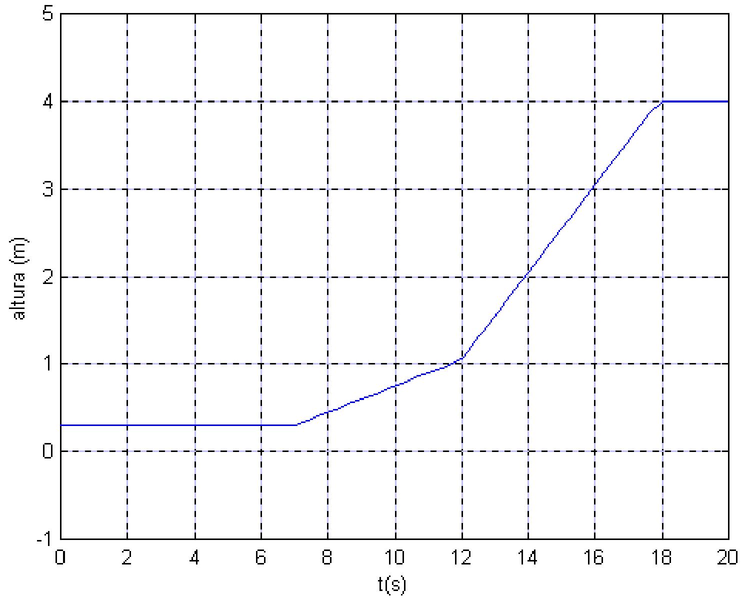
- la altura inicial: x0=.3 (la altura inicial será de 30 cm)

- la variable sys, donde el primer valor es el valor de estados continuos, el segundo, el valor de estados discretos, el tercero, el número de salidas y el cuarto el número de entradas. El valor sys(5)=0 es un valor reservado. Por otro lado sys(6)=0 indica la desactivación de la alimentación directa, es decir que no se usan las entradas u en el bloque correspondiente a flag=3;

Cuando flag es igual a 1: se determina la ecuación diferencial, donde sys es la derivada de la altura con respecto al tiempo.

Cuando flag es igual a 3, se asigna el valor de la altura calculada, sólo cuando el resultado no exceda la altura del tanque(4 m) y no disminuya de 0, o sea, que el resultado no sea negativo. En caso de exceso, se le asignará a la altura un valor igual al de la altura del tanque, y en caso de que la altura sea negativa, la altura se tomará igual a 0.

Al realizar la simulación, el resultado obtenido es el siguiente:



A.3.2 Control GPC iterativo con dinámica longitudinal

```
        --- principal.m ---

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  DEFINICION DEL MODELO DE PREDICCION %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Matrices dinámicas del avión en el eje longitudinal
format long
A=[0.0002 0.039 0 -9.81; -0.07 -0.317 250 0; 0.000088 -0.0028732 -0.439 0; 0 0 1 0];
B=[0.44; -5.46; -1.14362; 0];
C=[0 0 0 1];                    %ángulo de cabeceo
D=0;
Tm=0.06; % tiempo de muestreo

sistema=ss(A,B,C,D);
sistema_tf=tf(sistema);
avion_d=c2d(sistema_tf,Tm);

%dinámica actuador
actuador_cont=tf([37],[1 37]);
actuador_d=c2d(actuador_cont,Tm);

%dinámica piloto sin retardo (se añade en Simulink)
s=tf('s');
piloto_cont=0.74*(0.27*s+1);
piloto_d=c2d(piloto_cont,Tm,'tustin');

%dinámica del filtro paso baja para suavizar acción piloto
s=tf('s');
filtro_cont=(37.58/27)/(0.5*s+1)^2;
filtro_d=c2d(filtro_cont,Tm);

%dinámica total
sistema_total=avion_d*actuador_d*piloto_d*filtro_d;
warning off;
[num,den]=tfdata(d2c(sistema_total),'v');
warning on;

% parametros necesarios para la definición del modelo
n=1; % numero de salidas
m=1; % numero de entradas
```

```
R=0;
N=num;
D=den;

% transforación del modelo en continua al modelo CARIMA
% discretizado
[NA,NB]=g2car(N,D,R,n,m,Tm);

% Conversión  del formato estándar al formato por instantes
% Formulacion entrada/salida
[Ap,Bp]=convsist(NA,NB,n);

%retardo inducido por la planta
d=1;

%se suma el retardo introducido por piloto
d=d+5;         %5*0.06=0.3seg

A=Ap;
B=[zeros(n,m*d),Bp];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Definición de los parametros del controlador  %%%
%%%  predictivo                                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


N1=d+1; %  comienzo del horizonte de prediccion
N2=d+50; % final del horizonte de prediccion
N3=d+20; % horizonte de control

% matriz de ponderacion del error
R=eye(N2-N1+1);
% matriz de ponderacion del esfuerzo de control
Q=eye(N3); %*lambda,actualizándose en cada paso en GPC3sf


[n,a]=size(A);
[n,b]=size(B);
na=a/n-1;
nb=b/m-1;

% vectores de señales del controlador
du=zeros(N3*m,1);
wk=zeros(n*(N2-N1+1),1);
y=zeros(n,1);


% matriz de restricciones
GR=zeros((4*m*N3+3*n*(N2-N1+1)),m*N3);

% matrices constantes
D=zeros(nb*m,(nb+1)*m);
for i=1:nb
    D(((i-1)*m+1):i*m,((i-1)*m+1):i*m)=eye(m);
    D(((i-1)*m+1):i*m,(i*m+1):(i+1)*m)=-eye(m);
end;

for i=1:N3
    for j=1:i
        T((i-1)*m+1:i*m,(j-1)*m+1:j*m)=eye(m);
    end;
end;


Re=zeros((N2-N1+1)*n);
Qe=zeros(N3*m);

for i=1:n
    Re((i-1)*(N2-N1+1)+1:i*(N2-N1+1),(i-1)*(N2-N1+1)+1:i*(N2-N1+1))=R;
end;
for i=1:m
    Qe((i-1)*N3+1:i*N3,(i-1)*N3+1:i*N3)=Q;
end;
```

```
%Cálculo predicción: g es GN123, gp es GPN12, f es FN12
[GN123,GpN12,FN12]=mgpc(A,B,m,N1,N2,N3);

% generación de la matriz de restricciones
GR=[T;-T;eye(m*N3);-eye(m*N3);GN123;-GN123];

par.na=na;
par.nb=nb;
par.m=m;
par.wk=wk;
par.FN12=FN12;
par.n=n;
par.y=y;
par.GN123=GN123;
par.du=du;
par.GR=GR;
par.GpN12=GpN12;
par.Qe=Qe;
par.N=N;
par.N1=N1;
par.Re=Re;
par.N2=N2;
par.N3=N3;
par.T=T;
par.D=D;
```

A.3.3 Control GPC iterativo con dinámica lateral

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  DEFINICION DEL MODELO DE PREDICCION %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Matrices dinámica lateral/direccional del avión
A=[-0.056 0 -1 0.03924; -1.05 -0.47 0.39 0; 0.6 -0.032 -0.115 0; 0 1 0 0];
B=[0 0.012; 0.14 0.15; 0.008 -0.48; 0 0]; %actuaciones delta_a,delta_r
C=[1 0 0 0; 0 0 0 1];             %salidas: ángulo de derrape y alabeo
D=[0 0; 0 0];
Tm=0.05; % tiempo de muestreo
n=2;
m=2;

avion=ss(A,B,C,D);
avion_tf=tf(avion);
avion_d=c2d(avion_tf,Tm);

%Definición f.t actuadores.(Añadir saturación en Simulink)
actuador_cont=tf([1],[0.0169 0.121 1]);    actuador_d=c2d(actuador_cont,Tm);

%Definición función transferencia piloto sin retardo
s=tf('s');
aux=1.5/(0.13*s+1);
piloto_d=c2d(aux,Tm);

%dinámica total
sistema_total(1,1)=avion_d(1,1)*piloto_d*actuador_d;
sistema_total(1,2)=avion_d(1,2)*piloto_d*actuador_d;
sistema_total(2,1)=avion_d(2,1)*piloto_d*actuador_d;
sistema_total(2,2)=avion_d(2,2)*piloto_d*actuador_d;
warning off;

[n11,d11]=tfdata(d2c(sistema_total(1,1)),'v');
[n12,d12]=tfdata(d2c(sistema_total(1,2)),'v');
[n21,d21]=tfdata(d2c(sistema_total(2,1)),'v');
[n22,d22]=tfdata(d2c(sistema_total(2,2)),'v');
warning on;


N=[n11 n12;n21 n22];
D=[d11 d12;d21 d22];
```

```
R=zeros(2);

% Se Transforma el modelo continuo al modelo CARIMA discretizado
[NA,NB]=g2car(N,D,R,n,m,Tm);

% Conversión del formato estándar al formato por instantes
% Formulacion entrada/salida
[Ap,Bp]=convsist(NA,NB,n);

%retardo inducido por la planta
d=1;

%Se suma el retardo introducido por piloto
d=d+3;         %3*0.05=0.15seg

A=Ap;
B=[zeros(n,m*d),Bp];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Definición de los parametros del controlador  %%%
%%%  predictivo                                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N1=d+1; %  comienzo del horizonte de prediccion
N2=d+30; % final del horizonte de prediccion
N3=d+25; % horizonte de control

%a partir de aquí, igual que en A.3.1
% matriz de ponderacion del error
R=eye(N2-N1+1);
% matriz de ponderacion del esfuerzo de control
Q=eye(N3); %*lambda,actualizándose en cada paso en GPC3sf


[n,a]=size(A);
[n,b]=size(B);
na=a/n-1;
nb=b/m-1;

% vectores de señales del controlador
du=zeros(N3*m,1);
wk=zeros(n*(N2-N1+1),1);
y=zeros(n,1);


% matriz de restricciones
GR=zeros((4*m*N3+3*n*(N2-N1+1)),m*N3);

% matrices constantes
D=zeros(nb*m,(nb+1)*m);
for i=1:nb
    D(((i-1)*m+1):i*m,((i-1)*m+1):i*m)=eye(m);
    D(((i-1)*m+1):i*m,(i*m+1):(i+1)*m)=-eye(m);
end;

for i=1:N3
    for j=1:i
        T((i-1)*m+1:i*m,(j-1)*m+1:j*m)=eye(m);
    end;
end;


Re=zeros((N2-N1+1)*n);
Qe=zeros(N3*m);

for i=1:n
    Re((i-1)*(N2-N1+1)+1:i*(N2-N1+1),(i-1)*(N2-N1+1)+1:i*(N2-N1+1))=R;
end;
for i=1:m
    Qe((i-1)*N3+1:i*N3,(i-1)*N3+1:i*N3)=Q;
end;

%Cálculo predicción: g es GN123, gp es GPN12, f es FN12
[GN123,GpN12,FN12]=mgpc(A,B,m,N1,N2,N3);
```

```
% generación de la matriz de restricciones
GR=[T;-T;eye(m*N3);-eye(m*N3);GN123;-GN123];

par.na=na;
par.nb=nb;
par.m=m;
par.wk=wk;
par.FN12=FN12;
par.n=n;
par.y=y;
par.GN123=GN123;
par.du=du;
par.GR=GR;
par.GpN12=GpN12;
par.Qe=Qe;
par.N=N;
par.N1=N1;
par.Re=Re;
par.N2=N2;
par.N3=N3;
par.T=T;
par.D=D;
```

A.3.4 g2car.m

```
function [NA,NB]=G2car(N,D,R,n,m,Tm)
% funcion que transforma una matriz de transferencia de un sistema MIMO
% al modelo CARIMA del mismo.
% Llamada a la funcion
%                          [NA,NB]=G2car(N,D,R,n,m,Tm)
%
% Parametros:
%       N-> matriz de los polinomios de numeradores de G(s)
%       D-> matriz de los polinomios de denominadores de G(s)
%       R -> matriz con los retardos de G(s)
%       n -> numero de salidas al sistema
%       m -> numero de entradas al sistema
%       Tm -> tiempo de muestreo
% Devuelve:
%       NA -> matriz A del modelo CARIMA
%       NB -> matriz B del modelo CARIMA ( con los retardos incluidos)



[a,b]=size(N);
nn=floor(b/m); % nn= numero de terminos de los polinomios del numerador
[a,b]=size(D);
nd=floor(b/m); % nn= numero de terminos de los polinomios del denominador
ndn=nd;      % idem a nn del polinomio discreto
ndd=nd;        % idem a nd del polinomio discreto

% matrices de definición del modelo en discreto
ND=zeros(n,m*ndn);
DD=zeros(n,m*ndd);
RD=zeros(n,m);

% transformamos los polinomios en s en polinomios en z
% y rellenamos las matrices ND,DD,y RD

dmax=0;
for i=1:n;
   for j=1:m;
      numc=N(i,(j-1)*nn+1:j*nn);
      denc=D(i,(j-1)*nd+1:j*nd);
      [numd,dend]=c2dm(numc,denc,Tm,'zoh');
      if length(numd)<ndn
         numd=[numd,zeros(1,ndn-length(numd))];
      end
      ND(i,(j-1)*ndn+1:j*ndn)=numd;
      if length(dend)<ndd
         dend=[dend,zeros(1,ndd-length(dend))];
      end
      DD(i,(j-1)*ndd+1:j*ndd)=dend;
```

```
      RD(i,j)=round(R(i,j)/Tm);
      if dmax<RD(i,j)
         dmax=RD(i,j);
      end
   end
end


% Añadimos los retardos como ceros de la matriz de numeradores

NRD=zeros(n,m*(dmax+ndn));
for i=1:n
   for j=1:m
      NRD(i,(j-1)*(ndn+dmax)+1:(j-1)*(ndn+dmax)+RD(i,j)+ndn)=[zeros(1,RD(i,j)),ND(i,(j-
1)*ndn+1:j*ndn)];
   end
end

% calculamos el orden de los polinomios de A y B
na=ndd*m;
A=zeros(n,n*na);
nb=(m-1)*ndd+ndn+dmax;
B=zeros(n,m*nb);
for i=1:n
   aux=[1];
   for j=1:m
      aux=conv(aux,DD(i,(j-1)*ndd+1:j*ndd));
   end;
   A(i,(i-1)*na+1:(i-1)*na+length(aux))=aux;
end

for i=1:n
   for j=1:m
      aux=[1];
      for k=1:m
         if( k ~= j)
            aux=conv(aux,DD(i,(k-1)*ndd+1:k*ndd));
         end
      end
      aux=conv(aux,NRD(i,(j-1)*(ndd+dmax)+1:j*(ndd+dmax)));
      B(i,(j-1)*nb+1:(j-1)*nb+length(aux))=aux;
   end
end


% Eliminacion de los ceros a la izq de los polinomios

nzmin=na;
for i=1:n
   nz=0;
   flag=0;
   for j=na:-1:1
      if (A(i,(i-1)*na+j)==0)&(flag==0)
         nz=nz+1;
      else
         flag=1;
      end
   end
   if(nz<nzmin)
      nzmin=nz;
   end;
end

nna=na-nzmin;
NA=zeros(n,n*nna);
for i=1:n
   NA(i,(i-1)*nna+1:i*nna)=A(i,(i-1)*na+1:(i-1)*na+nna);
end;

nzmin=nb;
for i=1:n
   for j=1:m
      nz=0;
      flag=0;
      for k=nb:-1:1
```

```
         if (B(i,(j-1)*nb+k)==0)&(flag==0)
            nz=nz+1;
         else
            flag=1;
         end
      end
      if(nz<nzmin)
         nzmin=nz;
      end;
   end
end

nnb=nb-nzmin-1;
NB=zeros(n,m*nnb);
for i=1:n
   for j=1:m
      NB(i,(j-1)*nnb+1:j*nnb)=B(i,(j-1)*nb+2:(j-1)*nb+nnb+1);
   end
end;

return
```

A.3.5 GPC3sf.m

```
function [sys,x0,str,ts] = GPC3sf(t,x,u,flag,struc)

switch flag,

case 0
   [sys,x0,str,ts]=mdlInitializeSizes(struc);
case { 1,4,9}
    sys=[];
 case 3
    sys=mdlOutputs(t,x,u,struc);
case 2
    sys=mdlUpdate(t,x,u,struc);
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end

function [sys,x0,str,ts]=mdlInitializeSizes(struc)
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  =(struc.na+1)*struc.n+(struc.nb+1)*struc.m ;
sizes.NumOutputs     = struc.m;
sizes.NumInputs      = 2*struc.n+(4*struc.m+2*struc.n+4+1) ;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
%para yk y uk
x0=zeros(sizes.NumDiscStates,1);
ts  = [-1 0];
str=[];

function sys=mdlUpdate(t,x,u,struc)
m=struc.m;
wk=struc.wk;
FN12=struc.FN12;
n=struc.n;
y=struc.y;
GN123=struc.GN123;
du=struc.du;
na=struc.na;
GR=struc.GR;
nb=struc.nb;
GpN12=struc.GpN12;
Qe=struc.Qe;
N=struc.N;
N1=struc.N1;
Re=struc.Re;
N2=struc.N2;
N3=struc.N3;
```

```
T=struc.T;
D=struc.D;
%lambda=struc.lambda;

%%%%%%%%%%%%%%%%%%%%actualizo yk uk y%%%%%%%%%
yk=x(1:(na+1)*n);
uk=x((na+1)*n+1:(na+1)*n+(nb+1)*m);
y=u(1:n);
ref=u(n+1:2*n);

%Actualización de las restricciones
RestrDU=u(2*n+1);
dumax=u(2*n+2:2*n+m+1);
dumin=u(2*n+m+2:2*n+2*m+1);

RestrU=u(2*n+2*m+2);
umax=u(2*n+2*m+3:2*n+3*m+2);
umin=u(2*n+3*m+3:2*n+4*m+2);

RestrY=u(2*n+4*m+3);
ymax=u(2*n+4*m+4:2*n+4*m+n+3);
ymin=u(2*n+4*m+n+4:2*n+4*m+2*n+3);

Restr=u(2*n+4*m+2*n+4);

lambda=u(2*n+4*m+2*n+5)


%%%%%%%%%%%%%%%%%%%%%%un paso%%%%%%%%%%%%%%%%%
%En la variable y se almacenan las salidas leidas
% de la planta

    for i=na+1:-1:2;
        yk((i-1)*n+1:i*n)=yk((i-2)*n+1:(i-1)*n);
    end;
    yk(1:n)=y;

    duk=D*uk;

    for i=1:N2-N1+1
            wk((i-1)*n+1:i*n)=ref;
    end;

   %[t,yk',uk']
   %cálculo de la respuesta libre

    f=GpN12*duk+FN12*yk;

    % Actualización de las restricciones de las entradas

    for i=1:N3
        Cumax((i-1)*m+1:i*m)=umax-uk(1:m);
        Cumin((i-1)*m+1:i*m)=uk(1:m)-umin;
        Cdumax((i-1)*m+1:i*m)=dumax;
        Cdumin((i-1)*m+1:i*m)=-dumin;
    end;

    % Actualización de las restricciones de las salidas

    for i=1:(N2-N1+1)
        Cymax((i-1)*n+1:i*n)=ymax-f((i-1)*n+1:i*n);
        Cymin((i-1)*n+1:i*n)=f((i-1)*n+1:i*n)-ymin;
    end;



%C=[Cumax';Cumin';Cdumax';Cdumin';Cymax';Cymin'];

    % Obtencion de la ley de control
    for i=1:m
        Qe((i-1)*N3+1:i*N3,(i-1)*N3+1:i*N3)=lambda*eye(N3);
    end;

    H=2*(GN123'*Re*GN123+Qe);
    p=2*GN123'*Re*(f-wk);
```

```
    if(Restr==1)
      if (RestrU==1)
         GR=[T;-T];
         C=[Cumax';Cumin'];
      else
         GR=[];
         C=[];
      end

      if (RestrDU==1)
         GR=[GR;eye(m*N3);-eye(m*N3)];
         C=[C;Cdumax';Cdumin'];
      end

      if(RestrY==1)
         GR=[GR;GN123;-GN123];
         C=[C;Cymax';Cymin'];
      end

      [du,J,exitflag]=quadprog(H,p,GR,C);

   end

   if ((Restr==0)|(exitflag<0))
       %Si la solución no es factible, se aplica la solución
       % sin restricciones
       disp('Conmutando a la solución sin restricciones');
       du=-1*inv(H)*p;
   end

    for i=(nb+1):-1:2;
        uk((i-1)*m+1:i*m)=uk((i-2)*m+1:(i-1)*m);
    end;
    uk(1:m)=du(1:m)+uk(1:m);

   %%%%%%%%%%%%%%%%%%%%%%%%fin un paso%%%%%%%%%%%%%%%%%%%%%%%%
   state(1:(na+1)*n)=yk;
   state((na+1)*n+1:(na+1)*n+(nb+1)*m)=uk;

   struc.yk=yk;
   struc.uk=uk;

   sys=state;


function sys=mdlOutputs(t,x,u,struc)
P=x((struc.na+1)*struc.n+1:(struc.na+1)*struc.n+struc.m);
sys=P;
```

A.3.6 Matconv.m

```
function [C]=matconv(A,B,nb)

% realiza la convolucion de dos matrices polinomiales C=A*B
% siendo
%   A: una matriz polinomial de n*m
%   B: una matriz polinomial de m*p
%   nb: el grado del polinomio de B
%
% llamada   [C]=matconv(A,B,nb)

[n,nta]=size(A);
[m,ntb]=size(B);
na=nta/m-1;
p=ntb/(nb+1);


C=zeros(n,p*(na+nb+1));

for i=1:na+nb+1;
    for j=1:na+1;
        for k=1:nb+1;
            if((j+k-1)==i)
```

```
                C(:,(i-1)*p+1:i*p)=C(:,(i-1)*p+1:i*p)+A(:,(j-1)*m+1:j*m)*B(:,(k-1)*p+1:k*p);
            end;
        end;
    end;
end;


%for i=0:na+nb;
%   for j=0:na;
%       for k=0:nb;
%           if((j+k)==i)
%
    C(:,(i*n+1):((i+1)*n))=C(:,(i*n+1):((i+1)*n))+A(:,(j*n+1):((j+1)*n)).*B(:,(k*n+1):((k+1)*n
));
%           end;
%       end;
%   end;
%end;
return;
```

A.3.7 Mgpc.m

```
function [GN123,GpN12,FN12]=mgpc(A,B,m,N1,N2,N3);

% gpc multivariable
% Resolucion de la ecuacion diofantica para la obtencion
% de las matrices de respuesta libre y forzada
% llamada   [GN123,GpN12,FN12]=mgpc(A,B,m,N1,N2,N3);



[n,a]=size(A);
[n,b]=size(B);
na=a/n-1;
nb=b/m-1;

% Definicion de las matrices que se van a usar
% matriz polin EN2*B
GN2=zeros(n,n*N2);
% matriz polin resultado de la Ec diofántica de grado N2-1
EN2=zeros(n,n*N2);
% vector de matrices polin. de grado na
GpN12=zeros((N2-N1+1)*n,m*nb);
% vector de N2 matrices polin. de grado na
FN2=zeros(N2*n,n*(na+1));
FN12=zeros((N2-N1+1)*n,n*(na+1));
% matriz de respuesta forzada
GN123=zeros((N2-N1+1)*n,N3*m);



Ap=zeros(n,n*(na+2));
D=zeros(n,2*n);
% Resolucion de la ecuacion diofantica

% calculamos A viruguilla
P(:,1:n)=eye(n);
P(:,n+1:2*n)=-eye(n);
% multiplicamos A * delta
Ap=matconv(P,A,na);

% Resolucion de la ecuacion diofantica

FN2(1:n,:)=-Ap(:,n+1:(n*(na+2)));

EN2(:,1:n)=eye(n);

for i=2:N2;
    EN2(:,(i-1)*n+1:i*n)=FN2((i-2)*n+1:(i-1)*n,1:n);
    for j=1:na;
        FN2(((i-1)*n+1):i*n,((j-1)*n+1):j*n)=FN2(((i-2)*n+1):(i-1)*n,(j*n+1):(j+1)*n)-FN2((i-
2)*n+1:(i-1)*n,1:n)*Ap(:,(j*n+1):(j+1)*n);
```

```
    end;
    FN2(((i-1)*n+1):i*n,(na*n+1):(na+1)*n)=-FN2((i-2)*n+1:(i-
1)*n,1:n)*Ap(:,((na+1)*n+1):(na+2)*n);
end;

% Generación de las matrices GN123 y GpN12

for i=N1:N2;
    Gi=matconv(EN2(:,1:i*n),B,nb);
    if(nb+i>1)
    GpN12(((i-N1)*n+1):(i-N1+1)*n,:)=Gi(:,(i*m+1):(nb+i)*m);
    end;
    for j=1:N3
        if(j<=i)
            GN123(((i-N1)*n+1):(i-N1+1)*n,((j-1)*m+1):j*m)=Gi(:,((i-j)*m+1):(i-j+1)*m);
        end;
    end;
end;



FN12=FN2((N1-1)*n+1:N2*n,:);

return;
```

A.3.8 Convist.m

```
function [CA,CB]=convsist(A,B,m)

% Llamada [CA,CB]=convsist(A,B,m)

sa=size(A);
sb=size(B);
n=sa(1);
na=sa(2)/n;
nb=sb(2)/m;

CA=zeros(n,na*n);
CB=zeros(n,m*nb);

for i=1:n;
    for j=1:na;
        CA(i,n*(j-1)+i)=A(i,(i-1)*na+j);
    end;
end;

for i=1:n;
    for j=1:m;
        for k=1:nb;
            CB(i,m*(k-1)+j)=B(i,(j-1)*nb+k);
        end;
    end;
end;

return;
```

A.4 CONTROL PID

A.4.1 Definición dinámica longitudinal

```
%      --- Definición sistema para uso en pid.mdl ---

%Matrices dinámicas del avión en el eje longitudinal
format long
A=[0.0002 0.039 0 -9.81; -0.07 -0.317 250 0; 0.000088 -0.0028732 -0.439 0; 0 0 1 0];
B=[0.44; -5.46; -1.14362; 0];
C=[0 0 0 1];                    %ángulo de cabeceo
D=0;
```

```
T=0.008;      %periodo de muestreo

%dinámica longitudinal del avión
avion_cont=ss(A,B,C,D);
avion_d=c2d(avion_cont,T);

%dinámica actuador sin saturación (se añade en pid.mdl)
actuador_cont=tf([37],[1 37]);
actuador_d=c2d(actuador_cont,T);

%dinámica piloto sin retardo (se añade en pid.mdl)
s=tf('s');
piloto_cont=0.74*(0.27*s+1);
piloto_d=c2d(piloto_cont,T,'tustin');

%dinámica del filtro
filtro_cont=1/(0.2*s+1)^2;
filtro_d=c2d(filtro_cont,T);
```

A.4.2 Cálculo de la MGR

```
%Usando las matrices dadas en el compendio de aviónica digital para la dinámica lateral de un
%gran avión comercial se desacoplará el ángulo de alabeo y derrape en función de las
%deflexiones de los alerones (delta_a) y del timón de profundidad (delta_r)

%Matrices dinámicas del avión en el eje transversal
format long
A=[-0.056 0 -1 0.03924;-1.05 -0.47 0.39 0;0.6 -0.032 -0.115 0;0 1 0 0]
B=[0 0.012; 0.14 0.15; 0.008 -0.48; 0 0]; %actuacion delta_a y delta_r
C=[1 0 0 0; 0 0 0 1];             %salidas: ángulo de derrape y alabeo
D=[0 0; 0 0];

%Definición del sistema en el espacio de estados con estas matrices
sist_ss=ss(A,B,C,D);
set(sist_ss,'inputname',{'delta_a','delta_r'});  %def. de actuaciones
set(sist_ss,'outputname',{'derrape','alabeo'});  %def. de salidas

%Paso a la descripción externa con las 4 f.t correspondientes
sist_tf=tf(sist_ss);
sistema1=sist_tf(1,1);  %f.t de delta_a (entrada) a derrape  (salida).
sistema2=sist_tf(2,1);  %f.t de delta_a (entrada) a alabeo  (salida).
sistema3=sist_tf(1,2);  %f.t de delta_r (entrada) a derrape  (salida).
sistema4=sist_tf(2,2);  %f.t de delta_r (entrada) a alabeo  (salida).

%Construcción de la matriz (Kp)de ganancia del sistema en estado estacionario
Kp=zeros(2);
Kp(1,1)=freqresp(sistema1,0);  %delta_a a derrape
Kp(1,2)=freqresp(sistema3,0);  %delta_r a derrape
Kp(2,1)=freqresp(sistema2,0);  %delta_a a alabeo
Kp(2,2)=freqresp(sistema4,0);  %delta_r a alabeo

%Construcción de la Matriz de Ganancia Relativa (MGR)
MGR=Kp.*(inv(Kp))';        %.* es producto elemento a elemento
disp('la Matriz de Ganancia Relativa es: ');
MGR

%Paso a tiempo discreto las f.t para usar en Simulink
tm=0.008;
sistema1d=c2d(sistema1,tm);
sistema2d=c2d(sistema2,tm);
sistema3d=c2d(sistema3,tm);
sistema4d=c2d(sistema4,tm);
```

A.5 CONTROL LQR ATERRIZAJE

A.5.1 principal.m

```
                           %---principal.m---%

%Introducción de la dinámica de la aeronave en aterrizaje
%El vector de estados será [u w q theta h]'
A=[-0.038  -0.0513  0.00152            -32.2*pi/180*cos(-3*pi/180) 0;
   0.313   -0.605   -0.041-pi/180*235   32.2*pi/180*sin(-3*pi/180) 0;
  -0.0211   0.157   -0.612              0                          0;
  0         0        1                  0                          0;
  0        -1        0                  235*pi/180                 0];
B=[0.00005 0.158; -0.146 0.031; 0.459 0.0543; 0 0; 0 0];         %entrada=elevator,propulsión

%Introducción vectores C y D para realimentación de estados completa
C=diag(ones(5,1));
D=zeros(5,2);

%resolución del problema LQR
Q=zeros(5);
Q(1,1)=100;
Q(5,5)=20;
R=diag([5 5]);

KLQR=lqr(A,B,Q,R);

%cálculo del escalado para referencia no nula
Kescalado=mirscale_continuo(A,B,KLQR);

%simulamos control LQR
tsim=49;                %tiempo de simulación
txon=input('introduce el instante de conmutación planeo-recogida ');
sim('mod_principal',tsim)

%recopilación de variables para representación gráfica con vector de tiempos tout
u=estado(:,1);
w=estado(:,2);
q=estado(:,3);
theta=estado(:,4);
h=estado(:,5);

%definición derivada temporal del downrange
xderaprox=235+w.*(theta*pi/180);

%definición del downrange
x=integra(xderaprox,tout,-9540.6);
```

A.5.2 mirscale_continuo.m

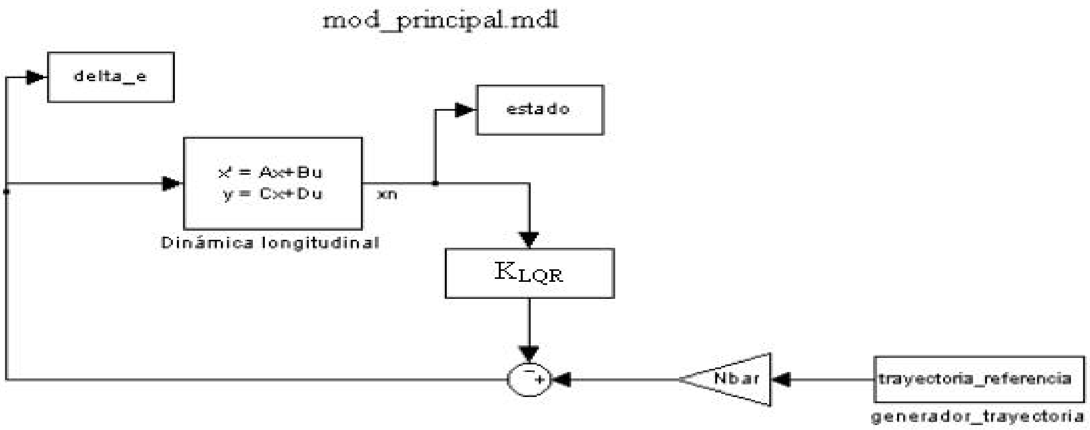
```
                          %---mirscale_continuo.m---%

function salida=mirscale_continuo(A,B,K);

   S=solve('A(1,2)*x+A(1,3)*y+A(1,4)*z+A(1,5)+(B(1,1)*N+B(1,2)*M)=B(1,1)*(K(1,2)*x+K(1,3)*y
           +K(1,4)*z+K(1,5))+B(1,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*z+K(2,5))',...

           'A(2,2)*x+A(2,3)*y+A(2,4)*z+A(2,5)+(B(2,1)*N+B(2,2)*M)=B(2,1)*(K(1,2)*x+K(1,3)*y
           +K(1,4)*z+K(1,5))+B(2,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*z+K(2,5))',...

           'A(3,2)*x+A(3,3)*y+A(3,4)*z+A(3,5)+(B(3,1)*N+B(3,2)*M)=B(3,1)*(K(1,2)*x+K(1,3)*y
           +K(1,4)*z+K(1,5))+B(3,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*z+K(2,5))',...

           'A(4,2)*x+A(4,3)*y+A(4,4)*z+A(4,5)+(B(4,1)*N+B(4,2)*M)=B(4,1)*(K(1,2)*x+K(1,3)*y
           +K(1,4)*z+K(1,5))+B(4,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*z+K(2,5))',...

           'A(5,2)*x+A(5,3)*y+A(5,4)*z+A(5,5)+(B(5,1)*N+B(5,2)*M)=B(5,1)*(K(1,2)*x+K(1,3)*y
           +K(1,4)*z+K(1,5))+B(5,2)*(K(2,2)*x+K(2,3)*y+K(2,4)*z+K(2,5))','M,N,x,y,z');

   salida=[0 eval(S.N); 0 eval(S.M)];
```

A.5.3 integra.m

```
                                  %---integra.m---%

function x=integra(xder,tsim,ini)

longitud=length(xder);
suma=ini;
x(1)=suma;

for i=2:1:longitud
    x(i)=suma+(xder(i)+xder(i-1))/2*(tsim(i)-tsim(i-1));
      suma=x(i);
end
```

A.5.4 Control SISO en aterrizaje

```
                                  %---principal.m---%

%Introducción de la dinámica de la aeronave en aterrizaje
%El vector de estados será [u w q theta h]'
A=[-0.038  -0.0513  0.00152          -32.2*pi/180*cos(-3*pi/180) 0;
   0.313   -0.605   -0.041-pi/180*235  32.2*pi/180*sin(-3*pi/180) 0;
  -0.0211   0.157   -0.612             0                          0;
  0         0        1                 0                          0;
  0        -1        0                 235*pi/180                 0];

B=[0.00005; -0.146; 0.459; 0; 0];  %Entrada: deflexión timón de profundidad

%Introducción vectores C y D para realimentación de estados completa
C=diag(ones(5,1));
D=zeros(5,1);

%resolución del problema LQR
Q=zeros(5);
Q(5,5)=100;
R=5;

KLQR=lqr(A,B,Q,R);

%calculo factor escala para referencia no nula
Nbar=rscale(A,B,[0 0 0 0 1],0,KLQR);                %rscale definida en A.1.1

%simulamos control LQR
tsim=49;            %tiempo de simulación
txon=input('introduce el instante de conmutación planeo-recogida ');

sim('mod_principal',tsim)

u=estado(:,1);
w=estado(:,2);
q=estado(:,3);
theta=estado(:,4);
h=estado(:,5);

%definición derivada temporal del downrange
xderaprox=235+w.*(theta*pi/180);

%definición del downrange
x=integra(xderaprox,tout,ini);                  %integra definida en A.5.3
```



A.6 CONTROL ROBUSTO H

A.6.1 coprimeunc.m

```
function [N,M,K,gammin]=coprimeunc(a,b,c,d,gamrel)
%
%Finds the controller which optimally "robustifies" a given shaped plant
%in terms of tolerating maximun coprime uncertainty.
%
%INPUTS:
%a,b,c,d: State-space descripcion of (shaped) plant.
%gamrel: gamma used is gamrel*gammin (typical gamrel=1.1)
%
%OUTPUTS:
%N,M left coprime factorization of (shaped) plant Gs, Gs=inv(M)N
%K: "Robustifying" controller (positive feedback).

S=eye(size(d'*d))+d'*d;
R=eye(size(d*d'))+d*d';
A1=a-b*inv(S)*d'*c;
Q1=c'*inv(R)*c;
R1=b*inv(S)*b';
[x1, x2, fail, reig_min]=ric_schr([A1 -R1; -Q1 -A1']); X=x2/x1;
[x1, x2, fail, reig_min]=ric_schr([A1' -Q1; -R1 -A1]); Z=x2/x1;
%gamma optima:
gammin=sqrt(1+max(eig(X*Z)))
%Se usa gamma mayor .....
gam=gamrel*gammin;
L=(1-gam*gam)*eye(size(X*Z))+X*Z;
F=-inv(S)*(d'*c+b'*X);
Ac=a+b*F+gam*gam*inv(L')*Z*c'*(c+d*F);
Bc=gam*gam*inv(L')*Z*c';
Cc=b'*X;
Dc=-d';

K=ss(Ac,Bc,Dc,Dc);        %controlador

%para descomponer Gs en su factorizacion
H=-(b*d'+Z*c')*inv(R);
A=a+H*c;
Bn=b+H*d; Bm=H;
C=inv(sqrt(R))*c;
Dn=inv(sqrt(R))*d; Dm=inv(sqrt(R));
N=ss(A,Bn,C,Dn);
M=ss(A,Bm,C,Dm);                       %Gs=inv(M)*N;
```

A.6.2 dosDOFcontrol.m

```
%En este archivo se calculan el controlador H-inf robusto de dos grados de libertad
%K=[K1 K2], para una planta compensada Gs y una dinámica de referencia deseada Tref.
```

```
%También se calcula la matriz Wi de escalado de la señal de referencia para conseguir un
%error nulo en régimen permanente

[As,Bs,Cs,Ds]=unpck(Gs);
[Ar,Br,Cr,Dr]=unpck(Tref);
[nr,nr]=size(Ar); [lr,mr]=size(Dr);
[ns,ns]=size(As); [ls,ms]=size(Ds);
Rs=eye(ls)+Ds*Ds'; Ss=eye(ms)+Ds'*Ds;
A1=(As-Bs*inv(Ss)*Ds'*Cs);
R1=Cs'*inv(Rs)*Cs; Q1=Bs*inv(Ss)*Bs';
[Z1,Z2,fail,reig_min]=ric_schr([A1' -R1; -Q1 -A1]); Zs=Z2/Z1;

rho=1;
A=daug(As,Ar);
B1=[zeros(ns,mr) ((Bs*Ds')+(Zs*Cs'))*inv(sqrt(Rs));
   Br zeros(nr,ls)];
B2=[Bs;zeros(nr,ms)];
C1=[zeros(ms,ns+nr);Cs zeros(ls,nr);rho*Cs -rho*rho*Cr];
C2=[zeros(mr,ns+nr);Cs zeros(ls,nr)];
D11=[zeros(ms,mr+ls);zeros(ls,mr) sqrt(Rs); -rho*rho*Dr rho*sqrt(Rs)];
D12=[eye(ms);Ds;rho*Ds];
D21=[rho*eye(mr) zeros(mr,ls);zeros(ls,mr) sqrt(Rs)];
D22=[zeros(mr,ms);Ds];
B=[B1 B2]; C=[C1;C2]; D=[D11 D12;D21 D22];
P=pck(A,B,C,D);

[l1,m2]=size(D12); [l2,m1]=size(D21);
nmeas=l2; ncon=m2; gmin=1; gmax=5; gtol=0.01;
[K,Gnclp,gam]=hinfsyn(P,nmeas,ncon,gmin,gmax,gtol);


[Ak,Bk,Ck,Dk]=unpck(K);
K=ss(Ak,Bk,Ck,Dk);
K=tf(K);

%Para el caso de un sistema con 2 entradas y una salida se tendrá
K1=K(1:2,1);
K2=K(1:2,2);

K10=dcgain(K1);
K20=dcgain(K2);

%Calculo del escalado de la referencia
Gs=tf(ss(As,Bs,Cs,Ds));
s=tf('s');
Gss0=dcgain(s*Gs);
Wi=inv(inv(Gss0*K20)*Gss0*K10);
```

A.6.3 Principal.m

```
%Introducción dinámica aeronave (9.6)
%El vector de estado es [u w q theta]'
A=[-0.038  -0.0513  0.00152           -32.2*pi/180*cos(-3*pi/180) ;
   0.313   -0.605   -0.041-pi/180*235   32.2*pi/180*sin(-3*pi/180) ;
  -0.0211   0.157   -0.612              0                          ;
  0         0        1                  0                           ];
B=[0.00005 0.158; -0.146 0.031; 0.459 0.0543; 0 0]; %entrada = [elevator throttle]'
C=[1 0 0 0;0 -1/235 0 1];      %salida =[u ángulo trayectoria]'
D=zeros(2,2);

%Prefiltro
s=tf('s');
W1=tf(ss(A,B,C,D));
W1(1,1)=1/s;
W1(1,2)=0;
W1(2,1)=0;
W1(2,2)=1/s;
%Planta compensada
Gs=ss(tf(ss(A,B,C,D))*W1);

%Cálculo del controlador robusto
[N,M,Ks,gammin]=coprimeunc(Gs.a,Gs.b,Gs.c,Gs.d,1.1);
Ks0=dcgain(Ks);
Wi=Ks0;     %Escalado de la referencia
```

```
%simulación
tsim=46.5;
sim('modelo',tsim);

%recopilación de variables para representación gráfica con vector de tiempos tout
u=estado(:,1);
w=estado(:,2);
q=estado(:,3);
theta=estado(:,4);

%definición derivada temporal del downrange
xder=(u+235).*cos(theta*pi/180)+w.*sin(theta*pi/180);

%defición velocidad de descenso
hder=(u+235).*sin(theta*pi/180)-w.*cos(theta*pi/180);

%definición del downrange
x=integra(xder,tout,-9540.6);    %función integra definida en A.5.3

%definición de la altura
h=integra(hder,tout,500);

GSC=-tan(3*pi/180)*(9540.6+x)+500;     %glide center line

plot(x,h,x,GSC);
grid;
```

A.6.4 principal.m

```
%Introducción dinámica aeronave (7.6)
%El vector de estado es [u w q theta h]'
Aavi=[-0.038  -0.0513  0.00152            -32.2*pi/180*cos(-3*pi/180) 0;
   0.313   -0.605   -0.041-pi/180*235   32.2*pi/180*sin(-3*pi/180) 0;
  -0.0211   0.157   -0.612             0                          0;
  0         0        1                 0                          0;
  0        -1        0                 235*pi/180                 0];
Bavi=[0.00005 0.158; -0.146 0.031; 0.459 0.0543; 0 0; 0 0]; %entrada = [elevator throttle]'
Cavi=[0 0 0 0 1];      %salida = h
Davi=zeros(1,2);
Gs=pck(Aavi,Bavi,Cavi,Davi);

%Tref
s=tf('s');
Tref=ss(1/(1.5*s+1));
Tref=pck(Tref.a,Tref.b,Tref.c,Tref.d);

%Cálculo del controlador robusto de 2 grados de libertad
dosDOFcontrol;

%simulación
tsim=7;
sim('modelo',tsim)

%recopilación de variables para representación gráfica con vector de tiempos tout
u=estado(:,1);
w=estado(:,2);
q=estado(:,3);
theta=estado(:,4);
h=estado(:,5);

%definición derivada temporal del downrange
xderaprox=(235+u).*cos(theta*pi/180)+w.*sin(theta*pi/180);

%definición del downrange
x=integra(xderaprox,tout,-512.7);

plot(x,h);
grid;
```

Apéndice B

Desarrollos y notas aclaratorias

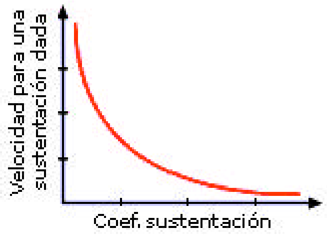
B.1 RELACIÓN ENTRE ÁNGULO DE ATAQUE Y VELOCIDAD

La fórmula de la sustentación para una aeronave es **L=CL*q*S,** donde **CL** es el coeficiente de sustentación, directamente proporcional al ángulo de ataque; **q** la presión aerodinámica (***1/2dv²*** donde ***d*** es la densidad y ***v*** la velocidad del viento relativo) y **S** la superficie alar. Como en vuelo normal la sustentación es siempre muy cercana al peso y puesto que la superficie alar es invariable (salvo que se extiendan flaps), la fórmula anterior podría escribirse:

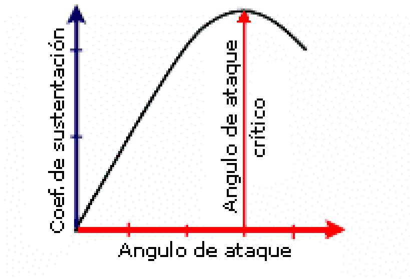
Sustentación (**L**) = Coeficiente de sustentación (**CL**)* 1/2dv² (**q**)

La igualdad reflejada en esta fórmula pone de relieve que:

1. En la sustentación total producida **L** los principales ingredientes son la velocidad y el ángulo de ataque, relacionados de forma que,
2. Para mantener una misma cantidad de sustentación, si la velocidad **v** disminuye, el coeficiente de sustentación **CL** (que depende del ángulo de ataque) debe incrementarse y viceversa, tal como muestra el gráfico de la siguiente figura:



A continuación se muestra un gráfico donde se observa que el coeficiente de sustentación (**CL**) es una función sencilla del ángulo de ataque, y como este coeficiente va aumentando con el ángulo de ataque hasta un punto (ángulo de ataque crítico) a partir del cual comienza a disminuir. Cada perfil tiene su propio ángulo de ataque crítico.



Puesto que los dos gráficos anteriores tienen en común el coeficiente de sustentación, combinando ambos vemos la correspondencia existente entre velocidades y ángulos de ataque, tal como se muestra en la siguiente figura; a mayor coeficiente de sustentación mayor ángulo de ataque y menor velocidad; cuando este coeficiente ha alcanzado su máximo la velocidad está en el mínimo; este mínimo es la velocidad de pérdida (Vs).

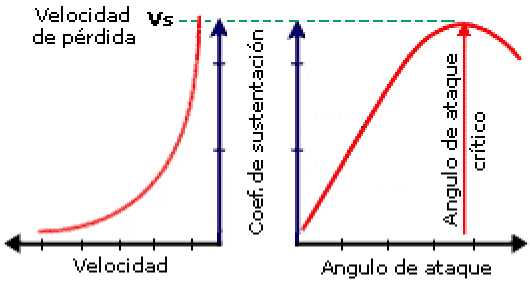


Fig.1.7.7 - Velocidad vs. Ángulo de ataque.

B.2 CÁLCULO DEL ESCALADO DE LOS CONTROLADORES DE LA FIGURA 9.4

2-DOF

$$y = G_s(K_2 y - K_1 W_i r) \;\rightarrow\; \text{reg. permanente}\begin{cases} s \rightarrow 0 \\ y = r \end{cases} \; G_s(0)K_1(0)W_i = (G_s(0)K_2(0) - I) \leftrightarrow$$
$$W_i = (G_s(0)K_1(0))^{-1}(G_s(0)K_2(0) - I) \leftrightarrow \; W_i = [(G_s(0)K_2(0) - I)^{-1} G_s(0)K_1(0)]^{-1}$$

Si se supone ahora la existencia de un integrador en la planta compensada G_s, y se denota $G_{ss}(0) = \lim_{s \to 0} sG_s(s)$ se puede concluir:

$$W_i = [(G_{ss}(0)K_2(0))^{-1} G_{ss}(0)K_1(0)]^{-1}$$

1-DOF

Si se sustituye en la expresión anterior $K_1(0)$ por I, y $K_2(0)$ por $K_s(0)$ se obtiene:

$$W_i = [K_s(0)^{-1} G_{ss}(0)^{-1} G_{ss}(0)]^{-1} = K_s(0)$$

Apéndice C

Resultados adicionales

C.1 GLIDE CONTROL

$$K_s = \begin{bmatrix} K_{s11} & K_{s12} \\ K_{s21} & K_{s21} \end{bmatrix}$$

$$K_{s11}(s) = \frac{2.756\, s^9 + 13.56\, s^8 + 42.72\, s^7 + 62.03\, s^6 + 53.84\, s^5 + 23.35\, s^4 + 4.383\, s^3 + 0.6066\, s^2 + 0.06069\, s + 0.002539}{s^{10} + 21.42\, s^9 + 165.5\, s^8 + 546.8\, s^7 + 794.2\, s^6 + 671.5\, s^5 + 278.5\, s^4 + 51.51\, s^3 + 7.084\, s^2 + 0.6966\, s + 0.02845}$$

$$K_{s12}(s) = \frac{-22.47\, s^9 - 161.8\, s^8 - 389.6\, s^7 - 508.7\, s^6 - 380.9\, s^5 - 157\, s^4 - 27.44\, s^3 - 3.789\, s^2 - 0.3586\, s - 0.01305}{s^{10} + 21.42\, s^9 + 165.5\, s^8 + 546.8\, s^7 + 794.2\, s^6 + 671.5\, s^5 + 278.5\, s^4 + 51.51\, s^3 + 7.084\, s^2 + 0.6966\, s + 0.02845}$$

$$K_{s21}(s) = \frac{-2.885\, s^9 - 46.75\, s^8 - 252.1\, s^7 - 415.7\, s^6 - 387.7\, s^5 - 168.7\, s^4 - 30.89\, s^3 - 4.293\, s^2 - 0.421\, s - 0.01668}{s^{10} + 21.42\, s^9 + 165.5\, s^8 + 546.8\, s^7 + 794.2\, s^6 + 671.5\, s^5 + 278.5\, s^4 + 51.51\, s^3 + 7.084\, s^2 + 0.6966\, s + 0.02845}$$

$$K_{s22}(s) = \frac{2.33\, s^9 + 20.4\, s^8 + 80.3\, s^7 + 130.4\, s^6 + 114.9\, s^5 + 49.65\, s^4 + 4.753\, s^3 + 0.7381\, s^2 + 0.02865\, s - 0.003702}{s^{10} + 21.42\, s^9 + 165.5\, s^8 + 546.8\, s^7 + 794.2\, s^6 + 671.5\, s^5 + 278.5\, s^4 + 51.51\, s^3 + 7.084\, s^2 + 0.6966\, s + 0.02845}$$

$$W_i = \begin{bmatrix} 0.0892 & -0.4588 \\ -0.5861 & -0.1301 \end{bmatrix}$$

C.2 FLARE CONTROL

$$K_1 = \begin{bmatrix} K_{1a} \\ K_{1b} \end{bmatrix}$$

$$K_{1a}(s) = \frac{212.6\, s^5 + 717.1\, s^4 + 1272\, s^3 + 1185\, s^2 + 304.8\, s + 34.19}{s^6 + 944.1\, s^5 + 5637\, s^4 + 1.608e004\, s^3 + 1.222e004\, s^2 + 2934\, s + 300.6}$$

$$K_{1b}(s) = \frac{16.53\, s^5 + 47.58\, s^4 + 68.06\, s^3 + 36.4\, s^2 + 8.267\, s + 0.8786}{s^6 + 944.1\, s^5 + 5637\, s^4 + 1.608e004\, s^3 + 1.222e004\, s^2 + 2934\, s + 300.6}$$

$$K_2 = \begin{bmatrix} K_{2a} \\ K_{2b} \end{bmatrix}$$

$$K_{2a}(s) = \frac{-3723\, s^4 - 6842\, s^3 - 6440\, s^2 - 1481\, s - 160.2}{s^5 + 943.4\, s^4 + 5008\, s^3 + 1.275e004\, s^2 + 3724\, s + 451}$$

$$K_{2b}(s) = \frac{-289.4\, s^4 - 436\, s^3 - 335\, s^2 - 59.48\, s - 6.693}{s^5 + 943.4\, s^4 + 5008\, s^3 + 1.275e004\, s^2 + 3724\, s + 451}$$

$$W_i = -3.1383$$

Bibliografía

[1] Gero D. Aviation Disasters. The world's major civil airliner crashed since 1950. Somerset, 1993.

[2] Informe del Grupo de Personalidades creado por el Comisario Busquin. "La aeronáutica europea: una visión para el 2020", ISBN 92-894-0559-7. 2001

[3] José C. Meizoso Fernández – Carlos Meizoso Muñoz."Compendio de Aviónica Digital. Sistemas de Control y Guiado", tomo 2. Bellisco, 2000

[4] M.A. Muñoz. http://inicia.es/de/vuelo/indice.html

[5] Brogan, W. (1991). Modern Control Theory, Prentice Hall.

[6] Bruce, K.R., and Gangsaas, D., 1984, Improvement of the 767 lateral autopilot using optimal control design techniques.

[7] Gangsaas et al. (1986)

[8] Thompson, C.M., Coleman, E.E., and Blight, J.D., 1987, Integral LQG controller design for a fighter aircraft.

[9] Ulybyshev, Y., "Long-Term Formation Keeping of Satellite Constellation Using Linear-Quadratic Controller," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 109–115.

[10] Manuales de Matlab para control (CTM, Control Tutorials for Matlab). Universidad de Michigan. http://www.engin.umich.edu/group/ctm/

[11] Application of FDI to a boeing-747, Budapest University

[12] Pilot Analysis Using Multivariable Methods, Virginia Politecnic Institute

[13] A'Harrah, Ralph C. "An Alternate Control Scheme for Alleviating Aircraft Pilot Coupling", AIAA Guidance, Navigation and Control Conference, 3 :1194-1201 (1994).

[14] Sistema de Control de Servos para una Aeronave sin Piloto. Tesis de maestría, Jorge Sandoval L., IPN-México/IAJ-Ucrania 1999

[15] John Michael Phillips. Variable Strategy Model of the Human Operator, Virginia Politechnic Institute. July 27, 2000.

[16] E. F. Camacho y C. Bordons, Model Predictive Control. London: Springer-Verlag,1999

[17] D. W. Clarke, C. Mohtadi y P. S. Tuffs, "Generalized Predictive Control. Part I. The Basic Algorithmn". Automática, vol. 23, nº 2, pp. 137-148, 1987.

[18] D. W. Clarke, C. Mohtadi y P. S. Tuffs, "Generalized Predictive Control. Part II. The Basic Algorithmn". Automática, vol. 23, nº 2, pp. 137-148, 1987.

[19] Pilot Analysis Using Multivariable Methods, Virginia Politecnic Institute

[20] Jan M. Maciejowski. Predictive Control with Constraints. Paperback, 2002

[21] http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/sfg/sfun.html

[22] Katsuhiko Ogata, "Sistemas de Control en Tiempo Discreto". Ed. Prentice Hall, 1996.

[23] Y. Takahashi, C. S. Chan, and D. M. Auslander, "Parameter Tuning of Linear DDC Algorithms," ASME paper #70-WA/AUT-16, 1970. Also published as: "Parametereinstellung bei linearen DDC-Algorithmen," Regelungstecknik und Prozess-Datenverarbeitung, 6, 237-244, 1971.

[24] Gangsaas, D., et al. Aplication of modern sintesis to aircraft control: Three case studies. IEEE transactions on Automatic Control, AC-31, 11,1986.

[25] Youji Iiguni, Osaka University. An intelligent Landing System Based on a Human Skill Model. IEEE transactions on Aerospace and Electronic Systems vol. 34, nº 3, July 1998.

[26] Jih-Gau Juang, Hao-Hsiang Chang y Kai-Chung Cheng. Intelligent Landing Control Using Linearized Inverse Aircraft Model. Procedings of the American Control Conference Anchorage, AK May 8-10,2002.

[27] Yoshimasa Ochi y Kimio Kanai. Automatic Aproach and Landing for Propulsión Controlled Aircraft. Proceedings of the IEEE. Internacional Conference on Control Applications, Kohala Coast-Island of Hawai'i,USA, agosto 1999.

[28] Sigurd Skogestad y Ian Postlethwaite. Multivariable Feedback Control, Analysis and Design. 116-118.

[29] Sigurd Skogestad y Ian Postlethwaite. Multivariable Feedback Control, Analysis and Design. 385-389.

[30] Tischler, Mark B. A. "Advances in Aircraft Flight Control". Taylor & Francis July, 1996 Library Binding