

# Maniobra de aterrizaje automática de una Cessna 172P modelada en FlightGear y controlada desde un programa en C

Mario de la Rosa

Universidad de Sevilla, Camino de los descubrimientos s/n, 41092, mardeltri@gmail.com

Antonio J. Gallego

Universidad de Sevilla, Camino de los descubrimientos s/n, 41092, antgallen@gmail.com

Eduardo F. Camacho

Universidad de Sevilla, Camino de los descubrimientos s/n, 41092, efcamacho@us.es

## Resumen

*Es bien conocido que las maniobras de aterrizaje y despegue son las más complejas del vuelo de las aeronaves. En efecto, un gran porcentaje de los accidentes aéreos se da en la ejecución de estas operaciones. Si además se tiene en cuenta que la mayoría de estos accidentes se deben a errores humanos, es evidente la necesidad de un control automático que garantice la correcta ejecución de estas fases de vuelo. En este artículo se plantea una estrategia de aterrizaje sencilla y eficaz que hace uso de una estructura jerárquica de controladores PID. Para modelar la dinámica de la aeronave se empleará FlightGear, que es un simulador de vuelo de código abierto que ofrece un alto grado de realismo. El sistema de control de vuelo (Flight Control System, FCS) se incluye en un programa en C que se comunica con FlightGear mediante sockets UDP. Para llevar a cabo esta maniobra se ha desarrollado un sistema de guiado que se encarga de enviar las consignas a los controles implementados. Finalmente, cabe destacar que el aterrizaje se realizará en el aeropuerto de Sevilla con una Cessna 172P.*

**Palabras clave:** Controlador PID, Aterrizaje automático, FlightGear, Matlab, Simulink, Simulación en tiempo real

## 1 INTRODUCCIÓN

La correcta ejecución de las maniobras de aterrizaje y despegue es clave tanto en la aviación comercial como en la general. Según se indica en [2] el 12 % de los accidentes de aviones comerciales se dan en las maniobras de aterrizaje y ascenso inicial, mientras que el 49 % se dan en la aproximación final y el aterrizaje. Con estas cifras queda de manifiesto la necesidad del desarrollo de controles automáticos para garantizar la seguridad en estas fases de vuelo.

En la literatura se exponen estrategias de control cuyo desarrollo depende de los parámetros con los que se modelan la aeronave. Como ejemplo de estos algoritmos se tiene [3], que usa un control de seguimiento de modelo y control  $H_\infty$  para el control de una aeronave durante el aterrizaje. En este caso se emplea el simulador de vuelo X-Plane conectado con Matlab. De forma análoga, en [5] se expone un sistema de control de aterrizaje basado en inversión de la dinámica y control  $H_\infty$ . Finalmente en [1] se comparan tres controladores basados en el espacio de estados (MPC, LQR y LQR con planificación de ganancias), además en este artículo se trata cómo realizar la comunicación entre Matlab y FlightGear.

En este artículo se propone una estrategia de control jerárquico basada en PID y aplicada a una avioneta modelada en FlightGear. Se asume que este programa es una caja negra para el programa de control, tal y como lo sería un avión real. Esta estrategia se beneficia del alto grado de realismo proporcionado por FlightGear que usa modelos de la dinámica del vuelo realistas y altamente consolidados. Se trata de ecuaciones del movimiento no lineales junto con modelos aerodinámicos complejos.

El artículo comienza presentando una descripción de FlightGear en la sección 2. A continuación, la sección 3 incluye una descripción del sistema de guiado empleado para la generación de referencias que recibe el sistema de control, cuya estructura se detalla en la sección 4. Con el fin de validar las metodologías seguidas, la sección 5 presenta los resultados del vuelo de aterrizaje en FlightGear. Finalmente, en la sección 6 se exponen las conclusiones obtenidas.

## 2 MODELO FLIGHTGEAR

Este artículo usa FlightGear no solo como una herramienta de visualización sino también como una herramienta de modelado. FlightGear es un sim-

ulador de vuelo sofisticado, gratuito y completamente de código abierto. Este programa contiene una gran variedad de modelos de aeronaves como son: planeadores, helicópteros, cazas y aviones comerciales.

Actualmente FlightGear usa dos modelos de la dinámica del vuelo: JSBSIM y YAsim. Así mismo, se utiliza un modelo de terreno, TerraGear, y también se incluyen efectos atmosféricos.

JSBSIM emplea la ecuación de Newton para obtener la aceleración del centro de gravedad de la aeronave con respecto un sistema fijo a Tierra y con origen en el centro de la misma (Earth Centered Earth Fixed, ECEF):

$${}^b\dot{\mathbf{v}}_{CM/e} = \mathbf{F}_{A,T}/m - (\omega_{b/e} + 2\omega_{e/i}) \times \mathbf{v}_{CM/e}^b \quad (1)$$

donde  ${}^b\dot{\mathbf{v}}_{CM/e}$  es la derivada respecto a los ejes cuerpos de la velocidad del centro de masas respecto a los ejes ECEF,  $\mathbf{F}_{A,T}^b$  es el conjunto de fuerzas aerodinámicas y propulsivas en ejes cuerpo,  $m$  es la masa de la aeronave,  $\omega_{b/e}^b$  es la velocidad angular de los ejes cuerpo respecto a los ECEF expresada en ejes cuerpo,  $\omega_{e/i}$  es la velocidad angular de los ejes ECEF con respecto a los inerciales expresada en ejes cuerpo, y  $\mathbf{v}_{CM/e}^b$  es la velocidad del centro de masas respecto al sistema ECEF expresado en ejes cuerpo

La aceleración angular del sistema ejes cuerpo con respecto a los ejes inerciales viene dada por la siguiente relación:

$${}^b\dot{\omega}_{b/i} = (J^b)^{-1} [\mathbf{M}_{A,T}^b - \omega_{b/i}^b \times J^b \omega_{b/i}^b] \quad (2)$$

donde  ${}^b\dot{\omega}_{b/i}$  es la derivada respecto a los ejes cuerpo de la velocidad angular de los ejes cuerpo respecto a los ejes inerciales,  $J^b$  es la matriz de inercia expresada en ejes cuerpo, y  $\mathbf{M}_{A,T}^b$  son los momentos aerodinámicos y propulsivos en ejes cuerpo.

Se puede encontrar más información sobre el modelo en [9] o en [8].

### 3 SISTEMA DE GUIADO

El sistema de guiado de la aeronave es el encargado de establecer las consignas recibidas por los controladores. En la figura 1 se muestra esquemáticamente cuales son las entradas y salidas de dicho sistema. Se reciben dos entradas: la posición actual de la aeronave, que se actualiza continuamente, y los *waypoints* de paso, que pueden definirse tanto *online*, como *offline*, en este caso se empleará una planificación estratégica.

En primer lugar, en la sección 3.1, se establecerán los puntos de referencia y las consignas necesarias

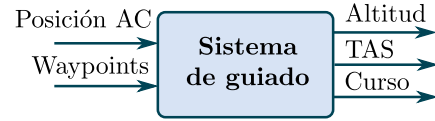


Figura 1: Sistema de guiado

para poder realizar un aterrizaje en el aeropuerto de Sevilla. Una vez definida la trayectoria, es necesario implementar un mecanismo de guiado que seleccione automáticamente el punto objetivo. El funcionamiento del mismo se tratará en detalle en la sección 3.2.

#### 3.1 Trayectoria de aproximación

La trayectoria de aproximación se definirá a partir de una serie de puntos de paso, *waypoints*, que quedan definidos por su latitud, longitud y altitud. Además, a cada uno de ellos se le ha asociado una velocidad, que será la velocidad de referencia desde dicho punto hasta el siguiente. Ver figuras Cabe destacar que las trayectorias de aproximación de los aeropuertos españoles están establecidas por ENAIRE. Dependiendo del sistema disponible para realizar la aproximación existen diferentes rutas a seguir. Por ejemplo, si se utilizase ILS (Instrumental Landing System), la trayectoria vertical a seguir sería la mostrada en la figura 2 [4]. Sin embargo, en este artículo, se realizará una aproximación visual, lo cual permitirá definir la trayectoria libremente. De esta forma se podrá mostrar el funcionamiento de los distintos controladores, haciendo así el aterrizaje más atractivo desde el punto de vista del control.

Punto	Latitud[°]	Longitud[°]
Posición inicial	37.426564	-6.014983
FAF	37.4175	-5.9925
Punto A	37.417663	-5.954661
Punto B	37.417839	-5.914728
Pista	37.417991	-5.887864

Tabla 1: Lista de waypoints (latitud y longitud)

Punto	Altitud[ft]	Velocidad[knot]
Posición inicial	1000	85
FAF	1000	85
Punto A	700	80
Punto B	250	70
Pista	80	60

Tabla 2: Lista de waypoints (altitud y velocidad)

En la figura 4 se muestra el perfil de descenso que se ha establecido. Inicialmente el avión se encuentra en el punto “posición inicial” a una altitud de 1000 ft con una velocidad de 85 knot, desde aquí

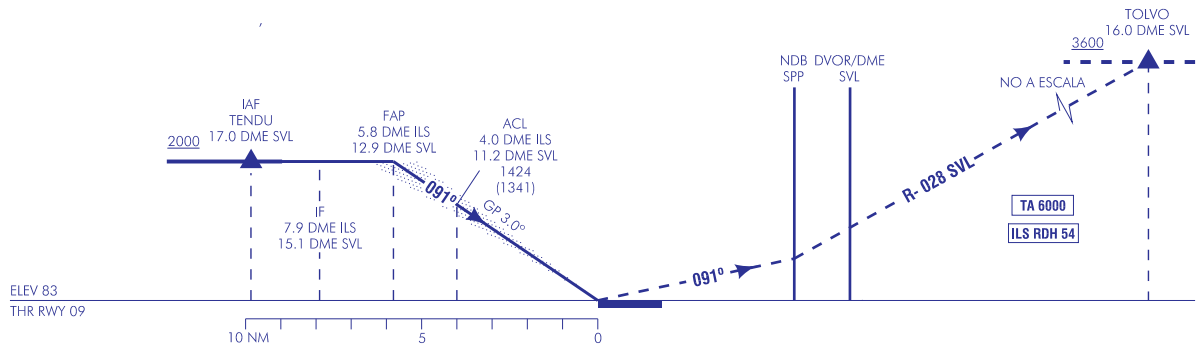


Figura 2: Senda de aproximación con ILS

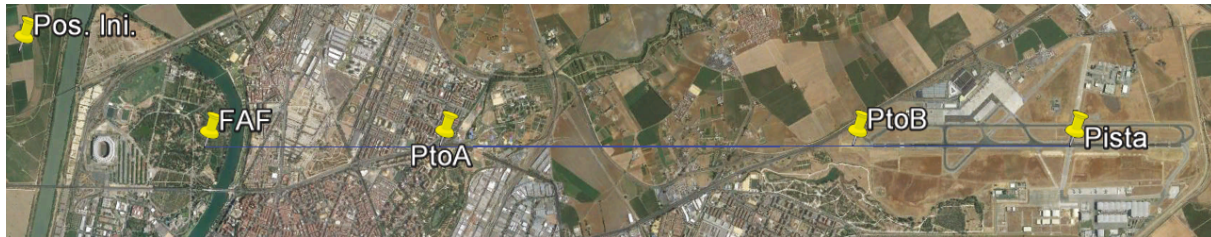


Figura 3: Puntos de paso proyectados

se dirigirá al FAF (Final Approach Fix) a una altitud de 1000 ft y una velocidad de 85 knot. El siguiente punto objetivo será el punto A, al cual se acercará a una altitud de 700 ft y una velocidad de 85 knot. Una vez alcance este punto se dirigirá al punto B a una altitud de 250ft y una velocidad de 70 knot. En el punto B cambiará el control de la velocidad aerodinámica, a partir de este momento la velocidad aerodinámica se controlará con el cabeceo y se impondrá la posición de palanca.

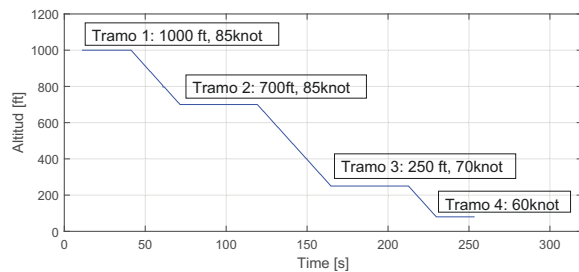


Figura 4: Perfil de descenso

### 3.2 MECANISMO DE GUIADO

En esta sección se definirá el algoritmo de guiado empleado, es decir, cómo se determinan las consignas a partir de la lista de waypoints y la posición de la aeronave.

#### 3.2.1 Selector de waypoints

El selector de waypoints será el encargado de seleccionar el waypoint objetivo. Para ello se calcula la distancia hacia el punto siguiente y cuando la misma es menor de 500 m, se selecciona el siguiente waypoint de la lista. La distancia se ha calculado como se indica a continuación:

$$d_{wp} = \sqrt{x^2 + y^2} \quad (3)$$

siendo

$$x = R_{mer}(\phi_{AC} - \phi_{wp}) \quad (4)$$

$$y = R_{norm} \cos(\phi_{wp})(\lambda_{AC} - \lambda_{wp}) \quad (5)$$

donde  $R_{mer}$  es el radio de curvatura del meridiano,  $R_{norm}$  el radio de curvatura normal,  $\phi_{AC}$  y  $\lambda_{AC}$  son la latitud y longitud de la aeronave (aircraft) y  $\phi_{wp}$  y  $\lambda_{wp}$  son la latitud y longitud del waypoint de destino.  $R_{mer}$  y  $R_{norm}$  son los radios de curvatura a lo largo de líneas de latitud y longitud constantes respectivamente. Las ecuaciones para su cálculo se pueden consultar en el capítulo 4, sección 4.1 de [7].

Una vez seleccionado el waypoint se tendrán las coordenadas del punto objetivo, la altura y la velocidad aerodinámica de referencia.

Por último cabe destacar que esta referencia de altitud no es la que recibe directamente el controlador. Esto se debe a que la tasa de variación de la referencia está limitada con el objetivo de conseguir una respuesta del controlador más suave y precisa.

### 3.2.2 Referencia de curso

La referencia de curso se debe calcular en función de la posición de la aeronave en cada momento. En primer lugar se implementó un algoritmo que calculaba el curso determinando la dirección de la recta que une la aeronave y el punto objetivo, es decir:

$$\psi = \text{atan} \left( \frac{R_{norm} \cos(\phi_{AC})(\lambda_{wp} - \lambda_{AC})}{R_{mer}(\phi_{wp} - \phi_{AC})} \right) \quad (6)$$

Sin embargo este método no proporcionaba un resultado aceptable ya que pequeños errores en el curso hacían que la dirección de aproximación a la pista no fuese la que se buscaba. Además, cuando la aeronave se iba acercando al *waypoint* objetivo, el curso comenzaba a variar de forma notable provocando que la avioneta oscilase al llegar a cada *waypoint*.

Este problema fue solventado implementando un método de seguimiento basado en campos vectoriales [6]. El objetivo de este algoritmo es asociar a la aeronave un vector de curso de referencia en función de la posición en la que se encuentra. Este método se describe en detalle a continuación.

En primer lugar se determina el curso de la recta que se desea seguir:

$$\psi_{12} = \text{atan} \left( \frac{y_2 - y_1}{x_2 - x_1} \right) \quad (7)$$

siendo  $x_2$  e  $y_2$  las coordenadas del punto objetivo y  $x_1$  e  $y_1$  las del *waypoint* anterior. En segundo lugar se determina el error lateral:

$$e_{lat} = (x_1 - x) \sin(\psi_{12}) + (y - y_1) \cos(\psi_{12}) \quad (8)$$

siendo  $x$  e  $y$  las coordenadas de la aeronave.

Las coordenadas cartesianas se han obtenido a partir de la latitud y la longitud tomando como referencia el ARP (*Airport Reference Point*) del aeropuerto de Sevilla.

$$x = R_{mer}(\phi - \phi_{ARP}) \quad (9)$$

$$y = R_{norm}(\lambda - \lambda_{ARP}) \quad (10)$$

Por último, la consigna de curso se calcula empleando la siguiente expresión:

$$\psi = \psi_{12} - \psi_{\infty} \frac{2}{\pi} \text{atan}(K_{path} e_{lat}) \quad (11)$$

siendo  $\psi_{\infty}$  y  $K_{path}$  dos variables de diseño. La primera hace referencia al valor del curso de aproximación a la recta cuando la aeronave se encuentra muy alejada de la misma y  $K_{path}$  define la suavidad del acercamiento a la recta.

## 4 ESTRUCTURA DE CONTROL

Para controlar la aeronave se ha planteado una estructura de control jerárquica que cuenta con dos capas: el controlador de alto nivel y el de bajo. El primero recibe las consignas del sistema de guiado es decir, la altitud, el curso y la velocidad aerodinámica. El segundo es el encargado de garantizar el seguimiento de las referencias que el controlador de alto nivel define: cabeceo y alabeo.

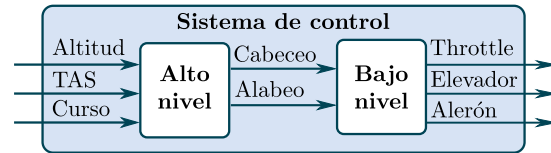


Figura 5: Sistema de control

En todos los controladores se ha implementado una *anti-windup* para evitar que el término integral aumente indefinidamente.

### 4.1 CONTROLADOR DE ALTO NIVEL

El controlador de alto nivel incluye tres controladores: el control de altitud, el control de curso y el control de velocidad aerodinámica. A continuación se detalla cada uno de ellos.

#### 4.1.1 Control de altitud

El control de altitud recibe la referencia de altitud y envía al controlador de bajo nivel la referencia de cabeceo. El método utilizado para implementar el PI ha sido de implementación directa tal y como se muestra en la siguiente relación:

$$\theta_{ref} = \theta_{trim} + k_p(e_k + \frac{T}{T_i} I_k) \quad (12)$$

siendo  $I_k$  el valor del término integral que se calcula como se indica a continuación:

$$I_k = I_{k-1} + e_k \quad (13)$$

Como se observa, se ha incluido un término de trimado con el objetivo de reducir el esfuerzo del término integral. Este valor,  $\theta_{trim}$ , se calcula en función de la velocidad de referencia.

$$\theta_{trim} = 0.002 \cdot TAS_{ref}^2 - 0.472 \cdot TAS_{ref} + 27.72 \quad (14)$$

Los coeficientes de esta función cuadrática se han obtenido ajustando la función a los valores obtenidos en varias simulaciones para cruceros a diferentes velocidades.

#### 4.1.2 Control de curso

El control de curso se realiza con un PI que recibe las señales del navegador y mantiene el curso

impuesto enviando una consigna de alabeo. El controlador implementado es un PI con implementación directa:

$$\phi_{ref} = k_p(e_k + \frac{T}{T_i} I_k) \quad (15)$$

Siendo  $I_k$  el valor del término integral que se calcula como se indica a continuación:

$$I_k = I_{k-1} + e_k \quad (16)$$

#### 4.1.3 Control de TAS

Para el control de la velocidad aerodinámica verdadera (*True Airspeed*, TAS) se han diseñado dos controladores: uno para controlar la velocidad a través de la posición de palanca (*throttle*) y otra a partir del cabeceo.

##### Control de TAS con throttle

Se ha implementado un PI para controlar la velocidad aerodinámica con la posición de palanca. Las entrada de este controlador es la consigna de velocidad del sistema de navegación y la salida es la posición de palanca establecida. Se ha empleado el método de implementación directa tal y como se indica a continuación:

$$\delta_{T,ref} = k_p(e_k + \frac{T}{T_i} I_k) \quad (17)$$

Siendo  $\delta_{T,ref}$  la posición de palanca de referencia e  $I_k$  el valor del término integral que se calcula como se indica en la siguiente expresión:

$$I_k = I_{k-1} + e_k \quad (18)$$

Además, para un mejor ajuste y relajar el trabajo del integrador se ha calculado una función de trimado tal que para cada velocidad de referencia se obtiene una posición de palanca determinada.

$$\delta_{T,trim} = 5 \cdot 10^{-5} TAS_{ref}^2 - 0.0061 \cdot TAS_{ref} + 0.89 \quad (19)$$

Estos coeficientes también se han calculado ajustando los valores obtenidos en simulaciones de vuelo en crucero para diferentes velocidades.

##### Control de TAS con cabeceo

Con el objetivo de controlar la velocidad de descenso en la última fase del aterrizaje se ha implementado este controlador que recibe la consigna de velocidad aerodinámica y establece un cabeceo que permite alcanzar esa referencia. Se ha implementado un PI con el método de implementación directa:

$$\theta_{ref} = \theta_{trim} + k_p(e_k + \frac{T}{T_i} I_k) \quad (20)$$

siendo  $I_k$  el valor del término integral que se calcula como se indica a continuación:  $I_k = I_{k-1} + e_k$  y  $\theta_{trim}$  es el valor del cabeceo de trimado para cada velocidad de referencia con un valor de *throttle* de 0.1.

$$\theta_{trim} = -0.001TAS_{ref}^2 - 0.05TAS_{ref} + 7.5 \quad (21)$$

## 4.2 Controlador de bajo nivel

En esta sección se tratará el control de bajo nivel que es el encargado de controlar las consignas de alabeo y cabeceo.

### 4.2.1 Controlador de cabeceo

El controlador de cabeceo recibe las consignas del control de altitud o del control de velocidad aerodinámica. Cuando el modo de vuelo es aterrizaje se utiliza este último mientras que cuando se vuela en crucero se reciben las consignas del control de altitud. Para controlar el cabeceo se ha implementado un PI con el método de implementación directa:

$$\delta_e = k_p(e_k + \frac{T}{T_i} I_k) \quad (22)$$

siendo  $I_k$  el valor del término integral que se calcula como se indica a continuación:  $I_k = I_{k-1} + e_k$

### 4.2.2 Controlador de alabeo

El controlador de roll recibe las señal de referencia del bloque de control de curso. Para controlarlo también se ha empleado un PI con el método de implementación directa:

$$\delta_a = k_p(e_k + \frac{T}{T_i} I_k) \quad (23)$$

siendo  $I_k$  el valor del término integral que se calcula como se indica a continuación:  $I_k = I_{k-1} + e_k$

## 5 RESULTADOS DE SIMULACIÓN

Es esta sección se incluyen y comentan los resultados de la simulación durante el aterrizaje de la Cessna 172P en el aeropuerto de Sevilla. Hay que tener en cuenta, que FlightGear tiene un modelo del contacto con el terreno, por lo tanto es necesario realizar la toma de contacto lo suficientemente suave como para que la aeronave permanezca intacta. Con la trayectoria implementada se consigue que la avioneta aterrice correctamente y sin sufrir daños.

En la figura 6 se muestra el seguimiento de la referencia de velocidad aerodinámica (*True Airspeed*,

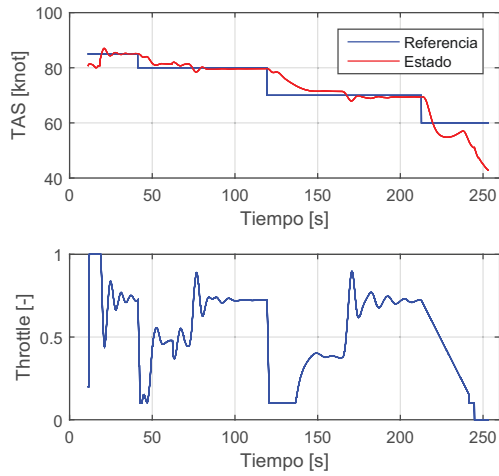


Figura 6: Seguimiento de la referencia de velocidad

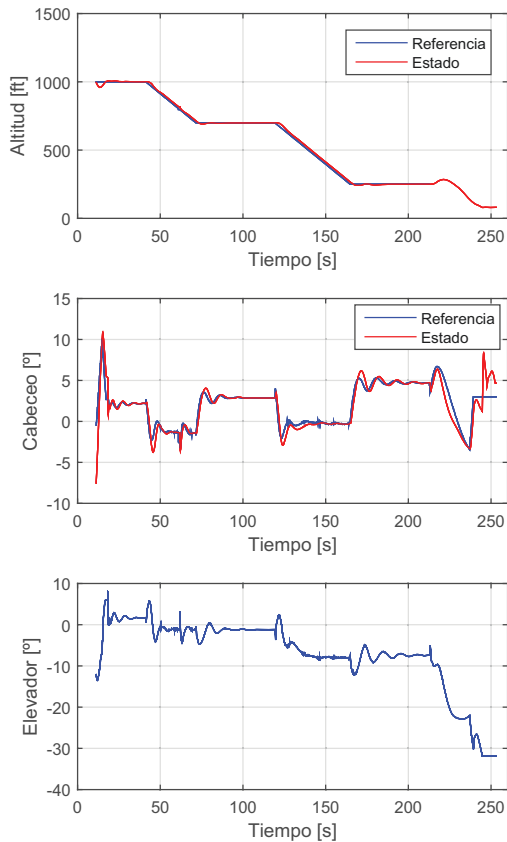


Figura 7: Seguimiento de la referencia de altitud durante el aterrizaje

TAS). Se observa que el seguimiento de la velocidad es adecuado, con sobreoscilaciones del orden de  $2 \text{ knot} \equiv 3.7 \text{ km/h}$ , que se van atenuando a medida que el sistema alcanza el régimen permanente. Cabe destacar que el cuarto tramo de la referencia en velocidad no se sigue con la variación de la posición de palanca si no con la variación del

cabeceo. En efecto, en la figura 7 se observa que no existe referencia de altitud en este tramo temporal.

Con respecto al control de la altitud, el seguimiento es muy satisfactorio con sobreoscilaciones de tan solo  $8 \text{ ft} \equiv 2.5 \text{ m}$ .

El control del curso (ver figura 8) también se sigue de forma suave, aunque se observan ciertas oscilaciones, cuyo origen se debe a la dinámica del sistema. En efecto, en avionetas con un solo motor de hélice en el morro, la dinámica lateral está acoplada con la dinámica longitudinal, concretamente los cambios de posición de palanca provocan un alabeo. Esto es debido a la resistencia aerodinámica de la hélice, que aumenta al aumentar la velocidad de la misma. Esta resistencia provoca un momento que es transmitido al fuselaje

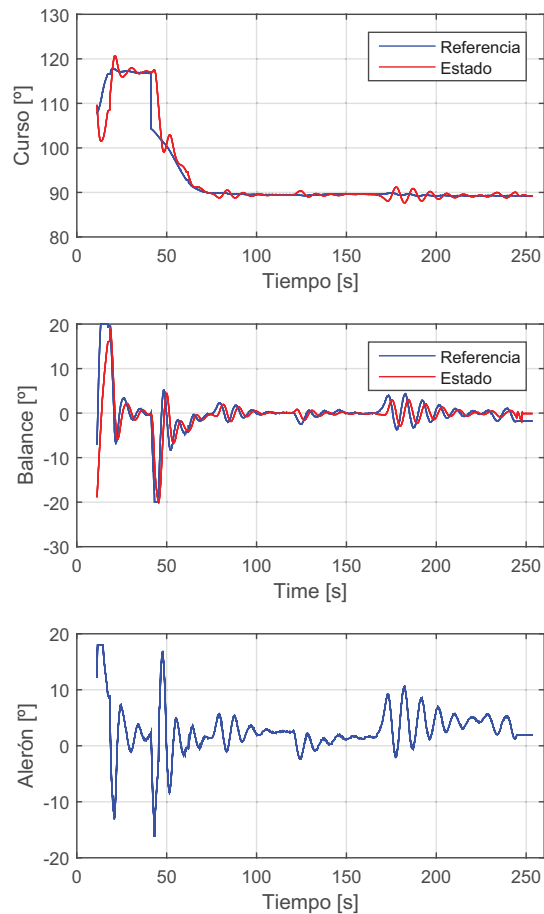


Figura 8: Seguimiento de la referencia de curso durante el aterrizaje



## 6 CONCLUSIONES

En este artículo se ha implementado la maniobra de aterrizaje para la avioneta Cessna 172P con un controlador PID y un modelo dinámico de FlightGear.

Con respecto al uso del modelo de FlightGear se puede decir que aporta claras ventajas frente al desarrollo del modelo matemático. Además de que se trata de un simulador consolidado y que aporta buenos resultados, con su uso se evita tener que calcular los parámetros aerodinámicos y propulsivos. Por otra parte, una gran ventaja consiste en poder cambiar fácilmente el modelo de la aeronave, lo cual permite comprobar la flexibilidad del control diseñado. Además se puede verificar la robustez del mismo incluyendo viento en la simulación.

El controlador PID ha mostrado buenos resultados teniendo en cuenta la simplicidad de esta metodología de control. Además, el hecho de no conocer los parámetros de la aeronave en cuestión, hace que este controlador sea un muy buen candidato para el problema planteado. En las simulaciones se observa que para futuras revisiones del control convendría incluir en el controlador el acoplamiento existente entre el balance y la posición de palanca del motor.

### Agradecimientos

Los autores quieren agradecer a la junta de Andalucía y al Ministerio de Educación por financiar el trabajo mediante el proyecto Control Predictivo de Microrredes Reconfigurables con Almacenamiento Híbrido y Móvil (DPI2016-78338-R).

### Referencias

- [1] Aschauer, G., Schirrer, A., Kozek, M., (2015) Co-Simulation of Matlab and FlightGear for Identification and Control of Aircraft, International Federation of Automatic Control (IFAC), pp. 067-072.
- [2] Boeing, (2015). Statistical Summary of Commercial Jet Airplane Accidents. Worldwide Operations — 1959-2015.
- [3] Çetin, E., Kutay, A. T., (2016) Automatic landing flare control design by model-following control and flight test on X-Plane Flight Simulator, International Conference on Mechanical and Aerospace Engineering (ICMAE), pp. 416-420.
- [4] ENAIRE, (2017) Cartas de aproximación de los aeródromos españoles: <http://www.enaire.es/csee/>

Satellite/navegacion-aerea/es/Page/1078418725163/?other=1083158950596&other2=1083857760137#ancla350

- [5] Lungu, M., Lungu, R., Tutunea, D., (2016) Control of Aircraft Landing using the Dynamic Inversion and the H-inf Control, IEEE. 17th International Carpathian Control Conference (ICCC), pp. 461-466.
- [6] Nelson, D. R., Barver, D. B., McLain, T. W., and Beard, R. W., (2007) Vector field path following for miniature air vehicles, IEE Trans. Robot, vol. 23, no. 3, pp. 519-529.
- [7] Robert M. Rogers, (2003) Applied mathematics in integrated navigation systems, second edition.
- [8] Stevens, B. L., Lewis, F. L., Johnson, E. N., (2016) Aircraft Control and Simulation, Third Edition, Wiley.
- [9] S. Berndt, Jon, & the JSBSim Development Team, (2011) JSBSim Reference Manual. <http://jsbsim.sourceforge.net/JSBSimReferenceManual.pdf>