

# CONTROL PREDICTIVO DE AVIONES

**Marina H. Murillo<sup>a</sup>, Pablo S. Rojas Fredini<sup>a</sup>, Alejandro C. Limache<sup>a</sup> y Leonardo L. Giovanini<sup>b</sup>**

<sup>a</sup>*International Center of Computational Methods in Engineering (CIMEC).  
INTEC-CONICET. Santa Fe, Argentina., <http://www.cimec.gov.ar/>*

<sup>b</sup>*Research Center for Signals, Systems and Computational Intelligence (SINC)  
Faculty of Engineering and Water Sciences, Universidad Nacional del Litoral*

**Palabras Clave:** model predictive control (MPC), aviones, simulación

**Resumen.** En este trabajo se presenta el desarrollo de un sistema de control automático basado en la técnica de control predictivo (MPC). MPC se obtiene como resultado de la minimización de una función de costo (o función objetivo) que involucra la dinámica del sistema y además permite la inclusión de las restricciones propias de la mecánica y de la física del mismo. Aplicando las entradas de control obtenidas a un modelo del avión Cessna 172, se logra que el mismo realice determinadas maniobras simples como ser variaciones de altitud. Se trabaja sobre un modelo con seis grados de libertad, el mismo es simulado con la ayuda del simulador de vuelo que se encuentran desarrollando los autores. Debido a que el modelo de un avión es un sistema no lineal, para poder implementar MPC, se trabaja con el sistema linealizado en varios puntos. Esto permite la obtención de una función objetivo cuadrática que asegura la convexidad de la misma y la existencia de algoritmos eficientes para hallar el control óptimo. Finalmente, el correcto funcionamiento del algoritmo se demuestra a través de dos maniobras de ascenso a altitudes preestablecidas.

## 1. INTRODUCCIÓN

En las últimas décadas las técnicas de control clásicas como *feedback control* (Stevens y Lewis, 2003), (Roskam, 2001) han dado lugar a técnicas de control más modernas como *model predictive control* (MPC) (Bemporad y Morari, 1999), (Kouvaritakis y Cannon, 2001), (Falcone et al., 2007) y (jun Yang et al., 2009). MPC ha ganado gran interés debido a que permite manejar las restricciones propias de los sistemas de manera más sencilla; esto es muy importante pues los sistemas físicos presentan ciertas restricciones, por ejemplo, las entradas de control y, en general, los estados deben satisfacer ciertos límites. Además MPC permite trabajar con modelos en espacio de estados multivariantes, es decir, con modelos que presentan múltiples entradas y/o múltiples salidas.

MPC se formula mediante la resolución de un problema de control *on-line*. Utilizando el estado actual se calcula una secuencia de control óptima, válida para un horizonte de control dado  $h_c$ , mediante la minimización de una función objetivo sujeta a ciertas restricciones. La entrada de control correspondiente al primer instante de muestreo se aplica al sistema y el cálculo se reinicia desplazando el horizonte de predicción  $h_p$  hacia adelante al próximo instante de muestreo. Las nuevas predicciones de las entradas de control se realizan para compensar perturbaciones no medidas, lo cual puede causar que la salida del sistema sea diferente de su predicción.

En el presente trabajo, nuestro sistema está representado por un modelo de avión Cessna 172 de seis grados de libertad. La representación en espacio de estados está formada por trece ecuaciones diferenciales que describen el comportamiento y la dinámica del avión (Stevens y Lewis, 2003). Nuestro objetivo se centra en el desarrollo de un sistema de control automático de altitud para aviones. Dicho sistema de control, además, ha sido implementado en el simulador de vuelo *Excalibur* para su evaluación. El simulador mencionado fue desarrollado en el Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC) (Limache et al., 2010) y (Limache et al., 2011). El mismo facilita el testeo de los diversos sistemas de control y permite verificar que el avión realice las maniobras deseadas volando en tiempo real. Esto último es muy importante ya que hace a la viabilidad de los desarrollos propuestos. El módulo de control se implementa como un subsistema independiente.

Para poder aplicar MPC, primero es necesario linealizar el sistema no lineal para obtener su representación en espacio de estados y luego transformarlo a su sistema discreto equivalente. La determinación de las entradas de control se realizan con una tasa de refresco igual al período de discretización  $\Delta T_s$ . En particular, se ha desarrollado un sistema de control automático de altitud basado en MPC y se ha implementado el mismo en el simulador de vuelo *Excalibur*, analizando el comportamiento del modelo del avión Cessna 172 en piloto automático para la realización de una maniobra de ascenso manteniendo velocidad constante.

## 2. MODELADO DEL AVIÓN

En particular, en este trabajo hemos implementado un modelo de avión con seis grados de libertad, el Cessna 172, con control automático de altitud basado en la técnica MPC. Para el modelo de propulsión se eligió un modelo de motor a hélice basado en (Allerton, 2009).

La representación de nuestro sistema en espacio de estados es la siguiente:

$$\left\{ \begin{array}{l} \dot{u} = r \cdot v - q \cdot w - g \cdot \sin \theta + \frac{F_x}{m} \\ \dot{v} = -r \cdot u + p \cdot w + g \cdot \sin \phi \cos \theta + \frac{F_y}{m} \\ \dot{w} = q \cdot u - p \cdot v + g \cdot \cos \phi \cos \theta + \frac{F_z}{m} \\ \dot{q}_0 = -\frac{1}{2} \cdot (p \cdot q_1 + q \cdot q_2 + r \cdot q_3) \\ \dot{q}_1 = -\frac{1}{2} \cdot (-p \cdot q_0 - r \cdot q_2 + q \cdot q_3) \\ \dot{q}_2 = -\frac{1}{2} \cdot (-q \cdot q_0 + r \cdot q_1 - p \cdot q_3) \\ \dot{q}_3 = -\frac{1}{2} \cdot (-r \cdot q_0 - q \cdot q_1 + p \cdot q_2) \\ \dot{p} = (c_1 \cdot r + c_2 \cdot p) q + c_3 \cdot l + c_4 \cdot n \\ \dot{q} = c_5 \cdot p \cdot r - c_6 \cdot (p^2 - r^2) + c_7 \cdot m \\ \dot{r} = (c_8 \cdot p - c_2 \cdot r) q + c_4 \cdot l + c_9 \cdot n \\ \dot{x}_e = u \cdot \cos \theta \cos \psi + v (\sin \phi \cos \psi \sin \theta - \cos \phi \sin \psi) + \\ + w (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{y}_e = u \cdot \cos \theta \sin \psi + v (\sin \phi \sin \psi \sin \theta + \cos \phi \cos \psi) + \\ + w (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \dot{z}_e = u \cdot \sin \theta - v \cdot \sin \phi \cos \theta - w \cdot \cos \phi \cos \theta \end{array} \right. \quad (1)$$

Donde

- $u$ ,  $v$  y  $w$  son las velocidades traslacionales en ejes cuerpo  $x_{body}$ ,  $y_{body}$  y  $w_{body}$  respectivamente
- $q_0$ ,  $q_1$ ,  $q_2$  y  $q_3$  son los cuaterniones que definen la orientación del avión con respecto al sistema coordenado tierra  $x_{earth}$ ,  $y_{earth}$  y  $z_{earth}$
- $p$ ,  $q$  y  $r$  son las velocidades angulares en los ejes cuerpo  $x_{body}$ ,  $y_{body}$  y  $w_{body}$  respectivamente
- $x_e$ ,  $y_e$  y  $z_e$  son las posiciones del avión expresadas en ejes tierra  $x_{earth}$ ,  $y_{earth}$  y  $z_{earth}$  respectivamente
- $\phi$ ,  $\theta$  y  $\psi$  son los ángulos de Euler que definen los movimientos de roll, pitch y yaw respectivamente

$$- C_1 = \frac{(J_y - J_z)J_z - J_{xz}^2}{J_x J_z - J_{xz}^2}$$

$$- C_2 = \frac{(J_x - J_y + J_z)J_{xz}}{J_x J_z - J_{xz}^2}$$

$$- C_3 = \frac{J_z}{J_x J_z - J_{xz}^2}$$

$$- C_4 = \frac{J_{xz}}{J_x J_z - J_{xz}^2}$$

$$- C_5 = \frac{J_z - J_x}{J_y}$$

$$- C_6 = \frac{J_{xz}}{J_y}$$

$$- C_7 = \frac{1}{J_y}$$

$$- C_8 = \frac{J_x(J_x - J_y) + J_{xz}^2}{J_x J_z - J_{xz}^2}$$

- $c_9 = \frac{J_x}{J_x J_z - J_{xz}^2}$
- $J_i$  es el momento de inercia alrededor del eje  $i$
- $J_{ij}$  es el producto cruzado de inercia
- $F_x, F_y$  y  $F_z$  son las fuerzas totales actuantes en el avión
- $l, m$  y  $n$  son los momentos totales actuantes en el avión

En la Fig. 1 puede observarse un esquema del avión modelado que facilita la comprensión y la lectura de las Ecs. (1). El sistema de ejes *cuerpo* (body) es una terna que se mueve solidariamente con el avión, el eje  $x_{\text{body}}$  se alinea apuntando a la nariz del avión, el eje  $y_{\text{body}}$  con el ala derecha del mismo y el eje  $z_{\text{body}}$  de forma tal que responda a la regla de la mano derecha. El sistema de ejes *tierra* (earth) es nuestro marco de referencia inercial y el sistema de ejes *viento* (wind) es tal que el eje  $x_{\text{wind}}$  se alinea con la dirección del viento relativo.

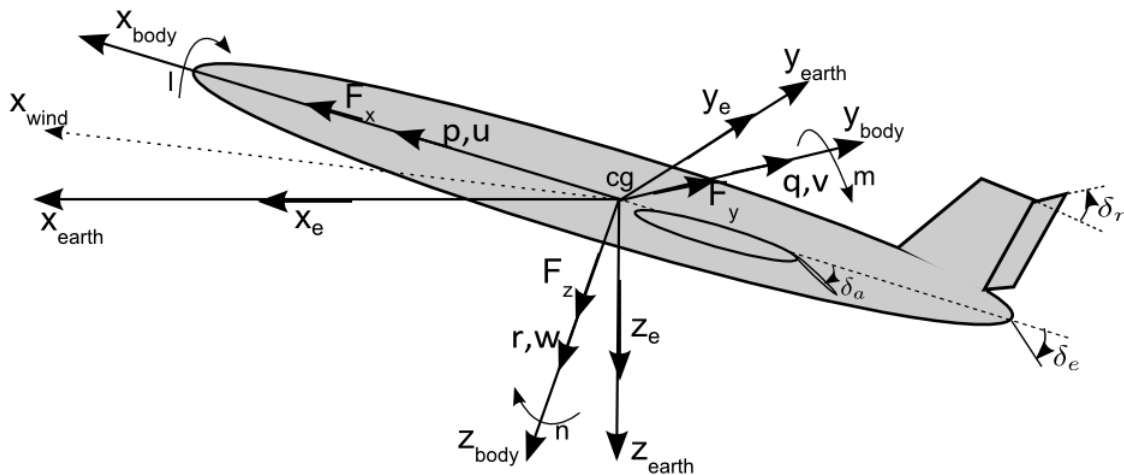


Figura 1: Esquema del avión

Donde

- $\delta_e$  es la deflexión del elevador en grados
- $\delta_a$  es la deflexión de los alerones en grados
- $\delta_r$  es la deflexión del *rudder* en grados
- $thtl$  es la posición de la columna del acelerador, cuyo valor se encuentra entre cero y uno.

Los ángulos de Euler permiten describir la orientación de un sistema de referencia móvil respecto a otro sistema de referencia fijo. El sentido de las rotaciones y el orden en que se realizan dichas rotaciones son muy importantes debido a que los ángulos no cumplen con la ley conmutativa. La secuencia de rotaciones convencionalmente utilizada para describir la actitud instantánea con respecto al marco de referencia es la siguiente:

1. Rotación alrededor del eje  $z_e$  un ángulo  $\psi$  positivo (nariz a la derecha).
2. Rotación alrededor del eje  $y_1$  un ángulo  $\theta$  positivo (nariz hacia arriba).
3. Rotación alrededor del eje  $x_2$  un ángulo  $\phi$  positivo (ala derecha hacia abajo).

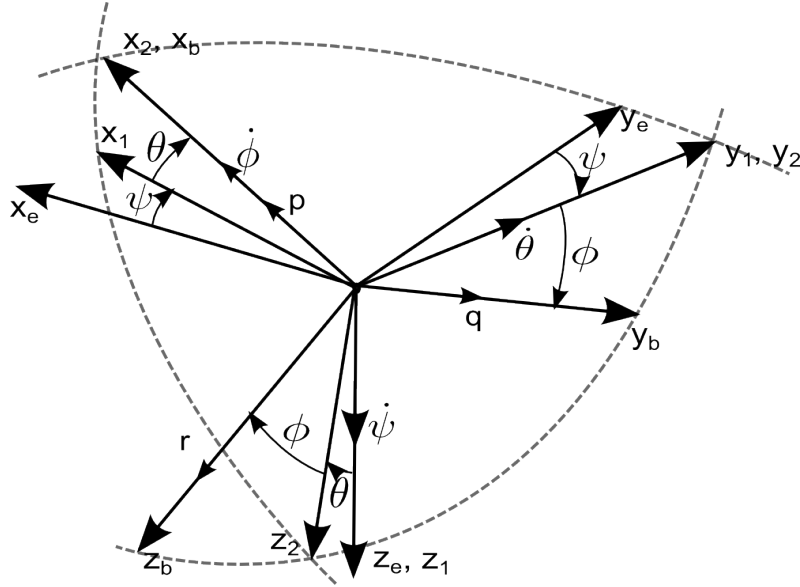


Figura 2: Ángulos de Euler

Existe otra forma de representar la orientación de un marco de referencia rotado; es la representación con cuaterniones (*quaternions*) (Ecs. 3 a 6 de la Ec. (1)). La misma evita la singularidad que presenta la representación con ángulos de Euler en la vecindad de  $\pm\theta$  y además evita la pérdida de un grado de libertad en el espacio tridimensional (*Gimbal lock*).

Para el cálculo de las fuerzas y momentos aerodinámicos se utilizó la herramienta Data Compendium (DATCOM) (USAF, 2012) provisto por la Fuerza Aérea de los Estados Unidos. Se decidió utilizar DATCOM debido a que es un compendio de información aerodinámica ampliamente validado y mundialmente conocido. Al contener información pública, permite validar diseños implementados en el simulador de vuelo. DATCOM proporciona principalmente los coeficientes aerodinámicos (y sus derivadas), y siguiendo los cálculos explicitados en (Duke et al., 1988) se pueden hallar las fuerzas y momentos resultantes.

Volviendo al sistema de ecuaciones (1), si definimos

$$\begin{aligned} \bar{f} &= [f_1 \ f_2 \ f_3 \ \cdots \ f_{13}]^T \\ \vec{x} &= [x_1 \ x_2 \ x_3 \ \cdots \ x_{13}]^T = [u \ v \ w \ q_0 \ q_1 \ q_2 \ q_3 \ p \ q \ r \ x_e \ y_e \ z_e]^T \\ \vec{u} &= [u_1 \ u_2 \ u_3 \ u_4]^T = [thtl \ \delta_e \ \delta_a \ \delta_r]^T \end{aligned}$$

donde  $f_1, f_2, \dots, f_{13}$  representan las trece funciones del lado derecho de la Ec. (1).

Desarrollando (1) en serie de Taylor de primer orden alrededor de un punto de operación  $(\vec{x}_e, \vec{u}_e)$  (Maciejowski, 2002)

$$\dot{\vec{x}} = \bar{f}(\vec{x}, \vec{u}) \approx \bar{f}(\vec{x}_e, \vec{u}_e) + \frac{\partial \bar{f}}{\partial \vec{x}}|_{(\vec{x}_e, \vec{u}_e)} \cdot (\vec{x} - \vec{x}_e) + \frac{\partial \bar{f}}{\partial \vec{u}}|_{(\vec{x}_e, \vec{u}_e)} \cdot (\vec{u} - \vec{u}_e) \quad (2)$$

donde

$$\frac{\partial \bar{f}}{\partial \bar{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_{13}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_{13}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{13}}{\partial x_1} & \frac{\partial f_{13}}{\partial x_2} & \dots & \frac{\partial f_{13}}{\partial x_{13}} \end{bmatrix} \quad \text{y} \quad \frac{\partial \bar{f}}{\partial \bar{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_4} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \dots & \frac{\partial f_2}{\partial u_4} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{13}}{\partial u_1} & \frac{\partial f_{13}}{\partial u_2} & \dots & \frac{\partial f_{13}}{\partial u_4} \end{bmatrix}$$

El punto de linealización  $(\vec{x}_e, \vec{u}_e)$  es un punto arbitrario, físicamente realizable, definido por un estado de vuelo  $\vec{x}_e$  en el que la posición de los controles están especificadas por el vector de control  $\vec{u}_e$ . Al ser un punto arbitrario,  $f(\vec{x}_e, \vec{u}_e)$  toma un valor distinto de cero. Como dicho valor es un valor fijo que puede calcularse para cada  $(\vec{x}_e, \vec{u}_e)$ , podemos considerarlo como una perturbación determinística que afecta a nuestro sistema. Entonces, si llamamos

$$A = \frac{\partial \bar{f}}{\partial \bar{x}}, \quad B_u = \frac{\partial \bar{f}}{\partial \bar{u}}, \quad \vec{d} = \bar{f}(\vec{x}_e, \vec{u}_e), \quad \delta \vec{x} = (\vec{x} - \vec{x}_e) \quad \text{y} \quad \delta \vec{u} = (\vec{u} - \vec{u}_e)$$

Podemos reescribir (2) como

$$\delta \dot{\vec{x}} = A|_{(\vec{x}_e, \vec{u}_e)} \cdot \delta \vec{x} + B_u|_{(\vec{x}_e, \vec{u}_e)} \cdot \delta \vec{u} + \vec{d} \quad (3)$$

Para poder aplicar control predictivo, necesitamos primero discretizar (3). Utilizando un bloqueador de orden cero obtenemos

$$\delta \vec{x}(k+1) = A_d|_{(\vec{x}_e, \vec{u}_e)} \cdot \delta \vec{x}(k) + B_{ud}|_{(\vec{x}_e, \vec{u}_e)} \cdot \delta \vec{u}(k) + B_{dd} \cdot \vec{d}(k) \quad (4)$$

Donde  $B_{dd}$  es la matriz identidad discreta y el valor de la variable  $\delta \vec{x}(k)$  se refiere a la variable de estado evaluada en el instante de tiempo  $t = k \cdot \Delta T_s$ , siendo  $\Delta T_s$  el período de discretización. En general, nos interesa minimizar las variaciones en la entrada de control (Maciejowski, 2002), (Rossiter, 2003). Si definimos

$$\delta \vec{u}(k) = \delta \vec{u}(k-1) + \Delta \delta \vec{u}(k) \quad (5)$$

Reemplazando (5) en (4) y escribiendo en forma matricial nos queda

$$\begin{bmatrix} \delta \vec{x}(k+1) \\ \delta \vec{u}(k) \end{bmatrix} = \begin{bmatrix} A_d & B_{ud} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta \vec{x}(k) \\ \delta \vec{u}(k-1) \end{bmatrix} + \begin{bmatrix} B_{ud} \\ \mathbf{I} \end{bmatrix} \Delta \delta \vec{u}(k) + \begin{bmatrix} B_{dd} \\ \mathbf{0} \end{bmatrix} \vec{d}(k) \quad (6)$$

Finalmente, podemos expresar (6) como

$$\delta \vec{\tilde{x}}(k+1) = \tilde{A} \cdot \delta \vec{\tilde{x}}(k) + \tilde{B}_u \cdot \Delta \delta \vec{u}(k) + \tilde{B}_d \cdot \vec{d}(k) \quad (7)$$

donde

$$\delta \vec{\tilde{x}}(k+1) = \begin{bmatrix} \delta \vec{x}(k+1) \\ \delta \vec{u}(k) \end{bmatrix} \quad \tilde{A} = \begin{bmatrix} A_d & B_{ud} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \tilde{B}_u = \begin{bmatrix} B_{ud} \\ \mathbf{I} \end{bmatrix} \quad \tilde{B}_d = \begin{bmatrix} B_{dd} \\ \mathbf{0} \end{bmatrix}$$

$\mathbf{I}$  es la matriz identidad y  $\mathbf{0}$  es una matriz de ceros.

Asumiendo que podemos medir el vector de estados completo, es decir que nuestra Ec. de salida es

$$\delta \vec{y}(k) = \tilde{C} \cdot \delta \vec{\tilde{x}}(k) \quad (8)$$

Con  $\tilde{C} = [C \ \mathbf{0}]$  y  $C$  es la matriz identidad. Luego, podemos realizar la predicción iterando el modelo descrito por (7)

$$\begin{aligned}
\delta\vec{x}(k+1) &= \tilde{A} \cdot \delta\vec{x}(k) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k) + \tilde{B}_d \cdot \vec{d}(k) \\
\delta\vec{x}(k+2) &= \tilde{A} \cdot \delta\vec{x}(k+1) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k+1) + \tilde{B}_d \cdot \vec{d}(k+1) \\
&= \tilde{A}^2 \cdot \delta\vec{x}(k) + \tilde{A}\tilde{B}_u \cdot \Delta\delta\vec{u}(k) + \tilde{A}\tilde{B}_d \cdot \vec{d}(k) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k+1) \\
&\quad + \tilde{B}_d \cdot \vec{d}(k+1) \\
\delta\vec{x}(k+3) &= \tilde{A} \cdot \delta\vec{x}(k+2) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k+2) + \tilde{B}_d \cdot \vec{d}(k+2) \\
&= \tilde{A}^3 \cdot \delta\vec{x}(k) + \tilde{A}^2\tilde{B}_u \cdot \Delta\delta\vec{u}(k) + \tilde{A}^2\tilde{B}_d \cdot \vec{d}(k) + \tilde{A}\tilde{B}_u \cdot \Delta\delta\vec{u}(k+1) \\
&\quad + \tilde{A}\tilde{B}_d \cdot \vec{d}(k+1) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k+2) + \tilde{B}_d \cdot \vec{d}(k+2) \\
&\quad \vdots \\
\delta\vec{x}(k+h_p) &= \tilde{A} \cdot \delta\vec{x}(k+h_p-1) + \tilde{B}_u \cdot \Delta\delta\vec{u}(k+h_p-1) + \tilde{B}_d \cdot \vec{d}(k+h_p-1) \\
&= \tilde{A}^{h_p} \cdot \delta\vec{x}(k) + \tilde{A}^{h_p-1}\tilde{B}_u \cdot \Delta\delta\vec{u}(k) + \tilde{A}^{h_p-1}\tilde{B}_d \cdot \vec{d}(k) \\
&\quad + \tilde{A}^{h_p-2}\tilde{B}_u \cdot \Delta\delta\vec{u}(k+1) + \tilde{A}^{h_p-2}\tilde{B}_d \cdot \vec{d}(k+1) + \dots \\
&\quad + \tilde{A}^{h_p-h_c}\tilde{B}_u \cdot \Delta\delta\vec{u}(k+h_c-1) + \tilde{A}^{h_p-h_c}\tilde{B}_d \cdot \vec{d}(k+h_c-1) \\
&\quad + \dots + \tilde{B}_d \cdot \vec{d}(k+h_p-1)
\end{aligned} \tag{9}$$

Finalmente, reescribiendo (9) en forma matricial y en notación más compacta, las predicciones de los estados pueden calcularse como

$$\delta\vec{X} = \Gamma \cdot \delta\vec{x}(k) + H_u \cdot \Delta\delta\vec{U} + H_d \cdot \vec{D} \tag{10}$$

donde

$$\begin{aligned}
\delta\vec{X} &= \begin{bmatrix} \delta\vec{x}(k+1) \\ \delta\vec{x}(k+2) \\ \delta\vec{x}(k+3) \\ \vdots \\ \delta\vec{x}(k+h_p) \end{bmatrix} & \Delta\delta\vec{U} &= \begin{bmatrix} \Delta\delta\vec{u}(k) \\ \Delta\delta\vec{u}(k+1) \\ \Delta\delta\vec{u}(k+2) \\ \vdots \\ \Delta\delta\vec{u}(k+h_c-1) \end{bmatrix} \\
\Gamma &= \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \tilde{A}^3 \\ \vdots \\ \tilde{A}^{h_p} \end{bmatrix} & H_u &= \begin{bmatrix} \tilde{B}_u & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \tilde{A}_t\tilde{B}_u & \tilde{B}_u & \mathbf{0} & \dots & \mathbf{0} \\ \tilde{A}_t^2\tilde{B}_u & \tilde{A}_t\tilde{B}_u & \tilde{B}_u & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_t^{h_p-1}\tilde{B}_u & \tilde{A}_t^{h_p-2}\tilde{B}_u & \tilde{A}_t^{h_p-3}\tilde{B}_u & \dots & \tilde{A}_t^{h_p-h_c}\tilde{B}_u \end{bmatrix} \\
\vec{D}(k) &= \begin{bmatrix} \vec{d}(k) \\ \vec{d}(k+1) \\ \vec{d}(k+2) \\ \vdots \\ \vec{d}(k+h_p-1) \end{bmatrix} & H_d &= \begin{bmatrix} \tilde{B}_d & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \tilde{A}_t\tilde{B}_d & \tilde{B}_d & \mathbf{0} & \dots & \mathbf{0} \\ \tilde{A}_t^2\tilde{B}_d & \tilde{A}_t\tilde{B}_d & \tilde{B}_d & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_t^{h_p-1}\tilde{B}_d & \tilde{A}_t^{h_p-2}\tilde{B}_d & \tilde{A}_t^{h_p-3}\tilde{B}_d & \dots & \tilde{B}_d \end{bmatrix}
\end{aligned}$$

### 3. CONTROL AUTOMÁTICO DE VELOCIDAD

#### 3.1. Función Objetivo Cuadrática

Como hemos dicho anteriormente, la técnica de control MPC se basa en la predicción del comportamiento futuro del sistema mediante la determinación de una secuencia de controles

óptima. Para poder realizar la predicción es necesario adoptar un criterio que permita reflejar nuestros objetivos; por ello se utiliza una función costo  $J$  que penaliza las diferencias entre la respuesta deseada y la predicha, y las variaciones en las entradas de control.

Definiendo nuestro vector de *setpoints* ampliado como  $\delta\vec{X}_{sp} = \vec{X}_{sp} - \vec{X}_e$  con  $\vec{X}_e = [\vec{X}_e \ \vec{U}_e]^T$ , definimos nuestra función costo como

$$J = (\delta\vec{X}_{sp} - \delta\vec{X})^T \cdot \tilde{Q} \cdot (\delta\vec{X}_{sp} - \delta\vec{X}) + \Delta\delta\vec{U}^T \cdot R \cdot \Delta\delta\vec{U} \quad (11)$$

donde  $\tilde{Q} = \tilde{C}^T \cdot Q \cdot \tilde{C}$

$Q$  es una matriz de pesos positiva semidefinida y  $R$  es una matriz de pesos positiva definida.

Reemplazando (10) en (11), operando y descartando los términos no dependientes de  $\Delta\delta\vec{U}$ , llegamos a que la función objetivo puede escribirse como

$$J = \Delta\delta\vec{U}^T \cdot (R + H_u^T \cdot \tilde{Q} \cdot H_u) \cdot \Delta\delta\vec{U} + 2(\delta\vec{x}(k)^T \cdot \Gamma^T - \delta\vec{X}_{sp}^T + \vec{D}^T \cdot H_d^T) \cdot \tilde{Q} \cdot H_u \cdot \Delta\delta\vec{U} \quad (12)$$

De (12), podemos ver que el problema a resolver se reduce a minimizar una función objetivo cuadrática en  $\Delta\delta\vec{U}$ .

### 3.2. Ejemplo: Control de altitud basado en MPC para el modelo de avión Cessna 172

A continuación presentaremos como ejemplo dos maniobras de ascenso. Se pretende hallar las entradas de control óptimas que permitan realizar un ascenso a una altitud deseada, a velocidad constante, de manera autónoma. Las maniobras están definidas por dos condiciones iniciales distintas: en la primera el avión se encuentra volando en vuelo recto y nivelado y en la segunda el avión se encuentra volando en equilibrio con una dada inclinación lateral (ángulo de *roll* distinto de cero).

Asumimos que el algoritmo de control se encuentra en una unidad de control embebida dentro del avión. La unidad de control se ejecuta con una tasa de refresco igual al período de discretización  $\Delta T_s$ . Por lo tanto, como se observa en la Fig. 3, los controles del avión se mantienen constantes e iguales a  $u_k$  hasta que es refrescado a un nuevo valor igual a  $u_{k+1}$ . Como el sistema en cuestión se trata de un sistema no lineal, cada vez que la unidad de control computa los controles óptimos el modelo del avión se linealiza de acuerdo a Ec. (2).

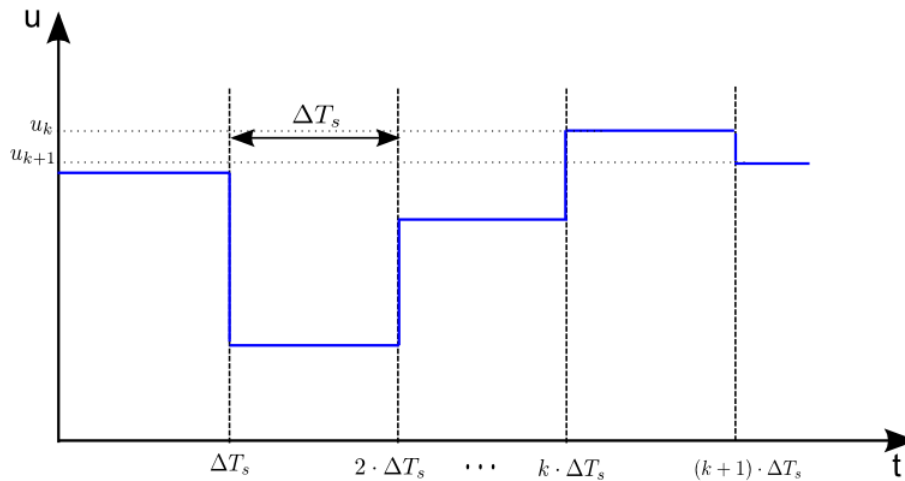


Figura 3: Esquema tiempo de discretización y bloqueador de orden cero



Consideramos que el avión se encuentra volando en una situación de equilibrio a una altura  $h = 1200m$ , con una velocidad de  $v_t = 45 \frac{m}{s}$  y deseamos realizar una maniobra de ascenso a  $h = 1525m$  manteniendo la velocidad constante.

Para las simulaciones utilizamos un período de discretización  $\Delta T_s = 0,5$  segundos, horizonte de predicción  $h_p = 20$  y horizonte de control  $h_u = 15$ . Los valores de ambos horizontes han sido obtenidos empíricamente observando que más allá de los valores utilizados, el aumento de los horizontes mencionados no producía cambios significativos en la respuesta del sistema. La matriz de peso de estados es una matriz cuyos valores distintos de cero se ubican en los siguientes lugares:  $Q(0, 0) = 1000$ ,  $Q(4, 4) = 10$ ,  $Q(6, 6) = 10$ ,  $Q(12, 12) = 100$ . Como puede observarse, hemos pesado solo los estados que nos interesan: velocidad  $u$ , los cuaterniones  $q_1$  y  $q_3$  y la altura  $z_e$ . Para evitar pesar los estados con valores de escala muy diferentes, realizamos una normalización de los mismos y minimizamos la función objetivo utilizando estos estados normalizados.

La matriz de pesos de las entradas es una matriz identidad

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los valores de ambas matrices se obtuvieron empíricamente observando la respuesta del sistema, pesando más los estados que debían alcanzar su valor de *setpoint* más rápidamente. Para la matriz de peso de entradas, los valores unitarios fueron suficientes para lograr una respuesta adecuada.

### 3.2.1. Maniobra de ascenso con vuelo recto nivelado (ángulo de *roll* nulo)

El avión se encuentra volando en equilibrio y se aplica control predictivo para realizar una maniobra de ascenso manteniendo la condición de vuelo recto y nivelado desde  $h = 1200m$  a  $h = 1525m$  a velocidad constante. En las Figs. 4(a) a 4(c) se puede observar la evolución resultante de los estados del sistema frente a la maniobra de ascenso indicada por el piloto automático, y en las Figs. 5(a) a 5(d) se puede ver la correspondiente evolución de los cuatro controles del avión.

En las Figs. 4(a) y 4(b) puede observarse que el sistema de control automático permitió alcanzar la altitud deseada manteniendo constante la velocidad del avión. En la Fig. 4(c) se puede apreciar que el avión mantuvo la condición de vuelo recto y nivelado (ángulo de *roll* aproximadamente cero).

Como en este ejemplo no se encuentran activos modos laterales del avión, en las Figs. 5(c) y 5(d) puede observarse que las variaciones de las superficies de control correspondientes a los alerones y *rudder*, toman valores aproximadamente nulos.

### 3.2.2. Maniobra de ascenso con vuelo recto no nivelado (ángulo de *roll* no nulo)

El avión se encuentra volando en equilibrio con una inclinación lateral definida por un ángulo de *roll*  $\phi = 5,7$  grados. Deseamos realizar la misma maniobra de ascenso que en el caso anterior pero además queremos que el sistema de control sea capaz de corregir el ángulo de *roll* a cero. Aquí se activan tanto modos longitudinales como laterales. Como queremos que el avión llegue a la altitud deseada en la condición de vuelo recto nivelado, los valores de los ángulos de *roll*

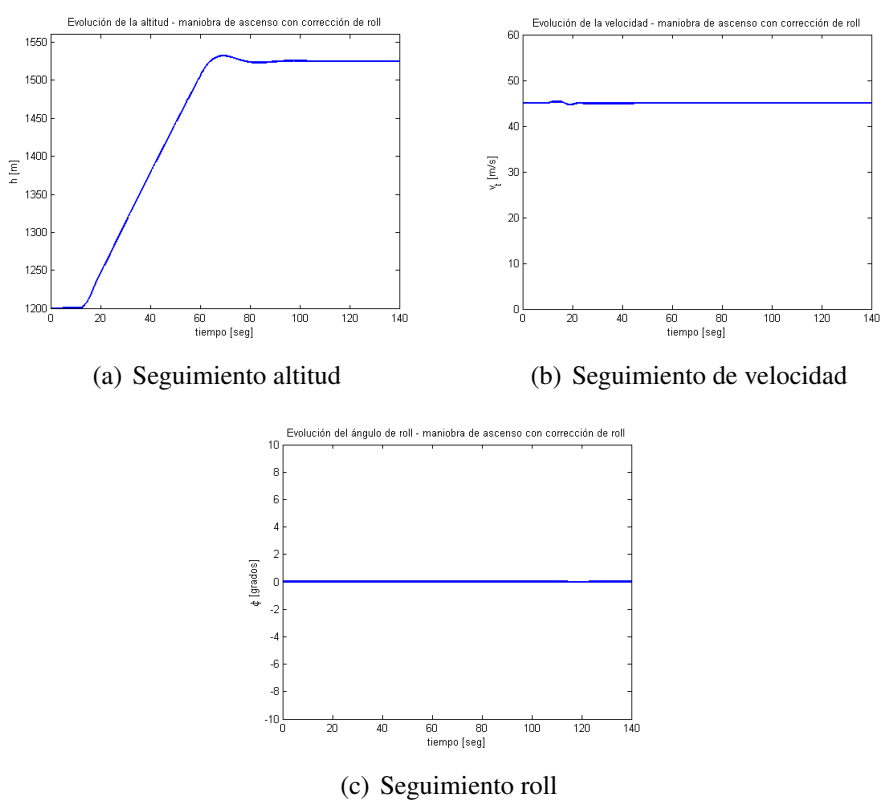


Figura 4: Evolución de los estados - maniobra de ascenso

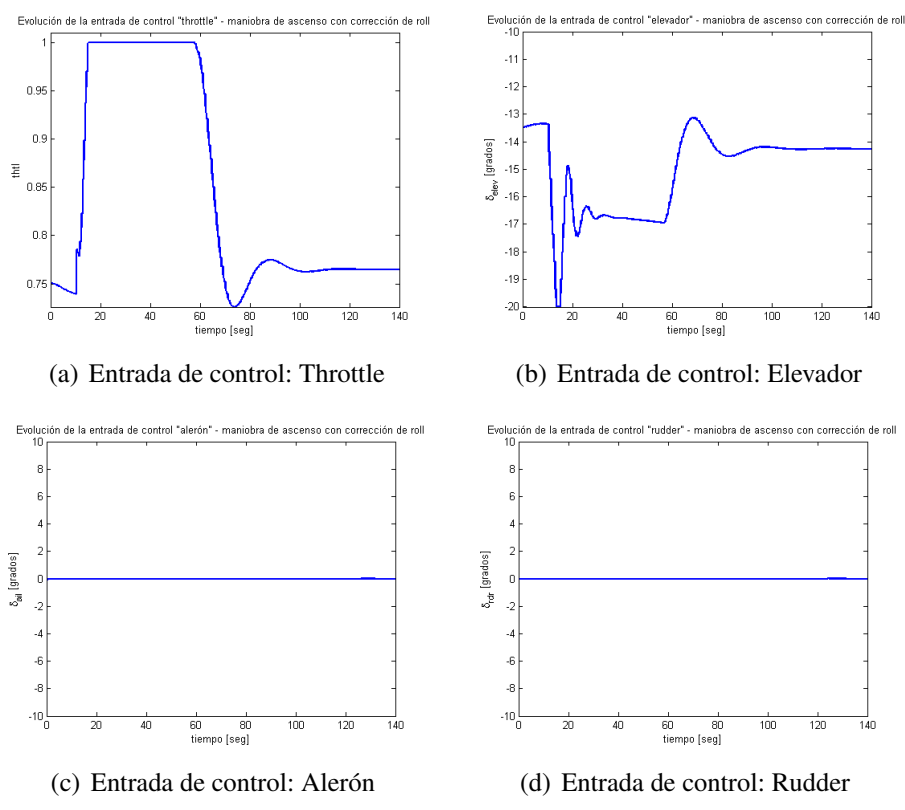


Figura 5: Evolución de las entradas de control - maniobra de ascenso

y de *yaw* deben ser nulos. Esta condición es equivalente a decir que los cuaterniones  $q_1$  y  $q_3$  deben ser nulos.

La evolución de los estados frente a la maniobra mencionada pueden observarse en las Figs. 6(a) a 6(d).

Al igual que en el caso anterior, vemos que el sistema de control automático permitió realizar

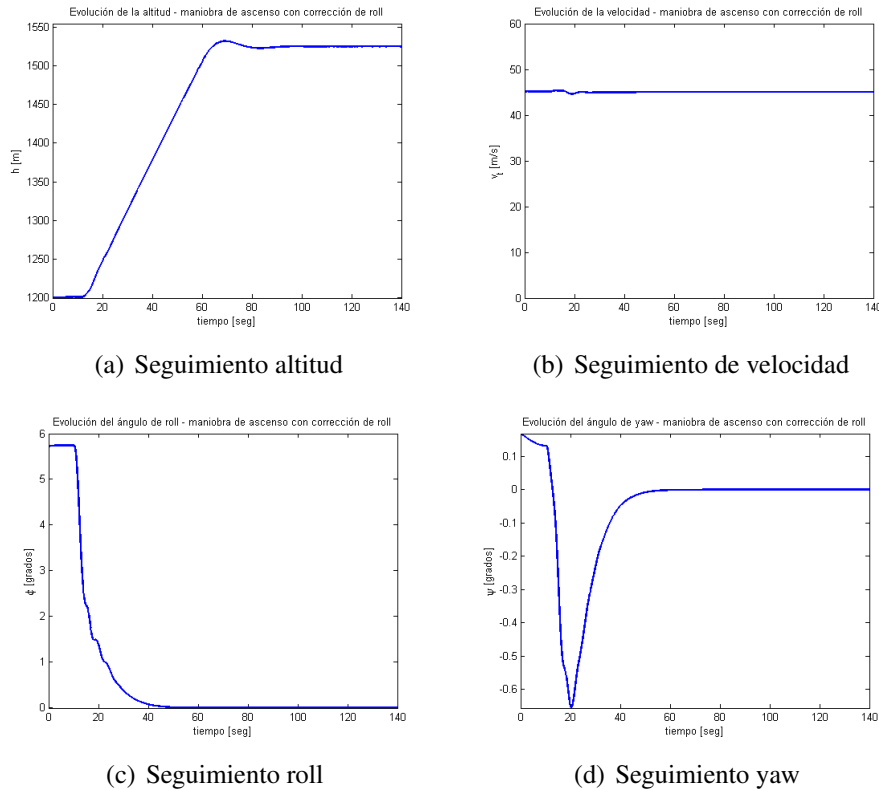


Figura 6: Evolución de los estados - maniobra de ascenso con  $\phi \neq 0$

el ascenso a la altura deseada manteniendo la velocidad constante. En la Figs. 6(c) y 6(d) se puede observar cómo el control es capaz de corregir los movimientos de *rollido* y de *guiñada*, llevando los ángulos correspondientes a cero. En las Figs. 7(a) a 7(d) se muestran la evolución de las entradas de control.

La evolución del throttle y del elevador son similares al caso del ejemplo anterior, pero ahora entran en juego los controles que modifican la dinámica lateral, es decir, los alerones y el *rudder*, como puede verse en las Figs. 7(c) y 7(d). Estos controles se mueven para nivelar el avión, cuando esto ocurre las deflexiones de ambas superficies de control tienden a cero.

#### 4. CONCLUSIONES

Mediante la realización de este trabajo pudimos desarrollar un sistema de control automático para aviones basado en la técnica MPC. El modelo de avión utilizado presenta seis grados de libertad y es representativo de un avión Cessna 172. El sistema de control desarrollado fue implementado en el simulador de vuelo *Excalibur*. Para verificar el comportamiento del avión en piloto automático, hemos explorado dos maniobras de ascenso, a velocidad constante, hasta una altitud preestablecida: una en la cual el avión se encuentra en vuelo recto nivelado y otra en la cual el avión vuela en equilibrio con una inclinación lateral dada (*i.e.* con activación de

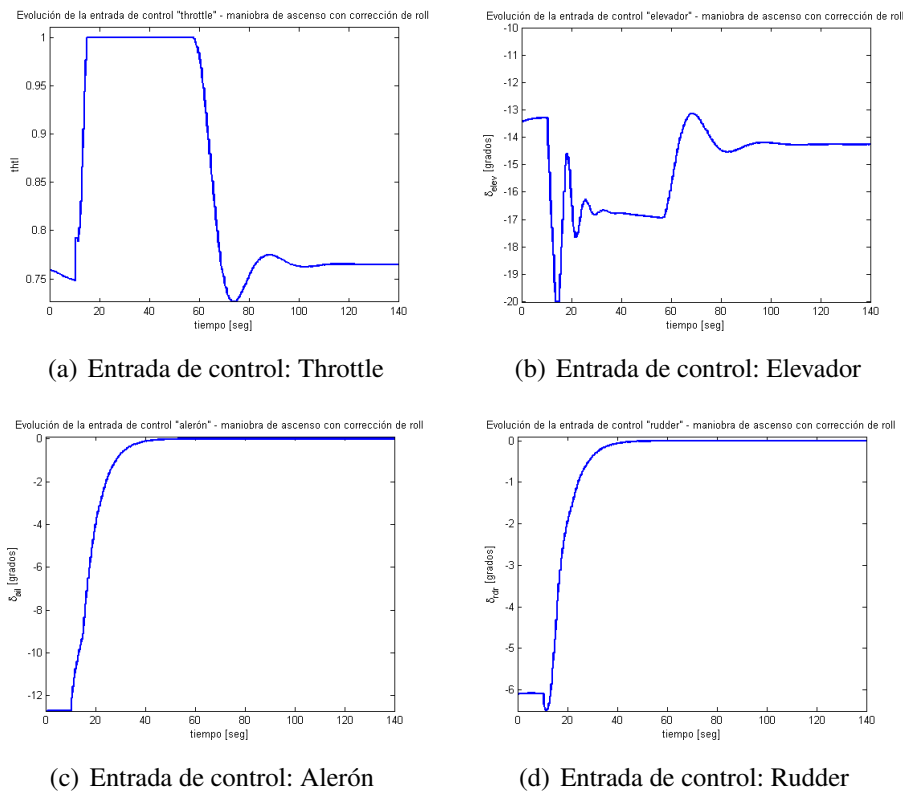


Figura 7: Evolución de las entradas de control - maniobra de ascenso con  $\phi \neq 0$

modos longitudinales y laterales). En ambos casos el control de altitud se pudo realizar satisfactoriamente, alcanzando los objetivos deseados.

## REFERENCIAS

- Allerton D. *Principles of Flight Simulation*. John Wiley & Sons, Ltd., 2009.
- Bemporad A. y Morari M. Robust model predictive control: A survey, in robustness in identification and control. *Lecture Notes in Control and Information Sciences*, 245:207–226, 1999.
- Duke, Antoniewicz, y Krambeer. Derivation and definition of a linear aircraft model. 1988.
- Falcone P., Tufo M., Borrelli F., Asgari J., y Tseng H.E. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. *Proceedings of the 46th IEEE Conference on Decision and Control*, páginas 2980–2985, 2007.
- jun Yang Z., hui Qi X., y li Shan G. Simulation of flight control laws design using model predictive controllers. *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, páginas 4213–4218, 2009.
- Kouvaritakis B. y Cannon M. *Nonlinear predictive control: theory and practice*. 2001.
- Limache, Murillo, Rojas, y Giovanini. Aspectos de diseño de un simulador de vuelo. 2011.
- Limache, Rojas, y Murillo. Diseño de un moderno simulador de vuelo en tiempo real. 2010.
- Maciejowski J. *Predictive control: with constraints*. Prentice Hall, 2002.
- Roskam J. *Airplane flight dynamics and automatic flight controls*. DARcorporation, 2001.
- Rossiter J. *Model-based predictive control: a practical approach*. CRC, 2003.
- Stevens B.L. y Lewis F.L. *Aircraft Control and Simulation*. John Wiley & Sons Inc., 2003.

USAF. Datcom. <http://www.holy cows.net/Datcom/>, 2012.