

Asociación Argentina
de Mecánica Computacional



Mecánica Computacional Vol XXX, págs. 2121-2131 (artículo completo)
Oscar Möller, Javier W. Signorelli, Mario A. Storti (Eds.)
Rosario, Argentina, 1-4 Noviembre 2011

CONTROL PREDICTIVO APLICADO A MODELOS SIMPLES DE AVIONES

Marina H. Murillo^a, Alejandro C. Limache^a y Leonardo L. Giovanini^b

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

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

Palabras Clave: control predictivo, control óptimo, optimización, aviones.

Resumen. Este trabajo presenta los primeros pasos en el desarrollo de sistemas de control para aviones. Como primera aplicación nuestro sistema a controlar es un modelo longitudinal de un avión. El mismo presenta tres grados de libertad: movimiento de traslación, movimiento de pitch y movimiento en el plano vertical. En primera instancia trabajamos con el modelo linealizado del mismo pudiendo luego comparar los resultados obtenidos al utilizar el sistema no lineal. La técnica de control elegida es Model Predictive Control (MPC). Esta técnica nos permite realizar maniobras con nuestro avión al aplicarle al mismo el control óptimo resultante de la minimización de una función costo sujeta a restricciones provenientes de la mecánica y de la física propias del sistema. Este trabajo nos servirá como base para el desarrollo de sistemas de control para modelos de aviones más sofisticados.

1. INTRODUCCION

La técnica de control por excelencia es el *feedback control* (Stevens y Lewis, 2003), (Roskam, 2001) pero desde hace algunas décadas, éstas técnicas han sido reemplazadas por técnicas de *control óptimo* (Bryson, 1985), (Bryson y Ho, 1975), (Borggaard et al., 1994). En estos días éstas son las técnicas más utilizadas en aeronáutica.

Sin embargo, recientemente se ha centrado el interés en el uso de técnicas de control basadas en horizontes móviles, las cuales permiten trabajar con sistemas con restricciones (Hilscher y Zeidan, 2000), (Lewis y Syrmos, 1995), (Lewis, 1992). Éstas técnicas forman lo que conocemos como *model predictive control* (MPC) (Bemporad y Morari, 1999), (Mayne et al., 2000), (Kouvaritakis y Cannon, 2001), (Maciejowski, 2002), (Rossiter, 2003) y (Giovanini, 2003).

Como la mayoría de los sistemas físicos presentan ciertas restricciones que se encuentran relacionadas con las características propias del sistema, es decir, las entradas de control y, en general, los estados deben permanecer acotados dentro de ciertos límites, resulta entonces de suma importancia el manejo de este conjunto de restricciones en el diseño de los sistemas de control. MPC permite un fácil manejo de estas restricciones, ésta es una de sus principales ventajas. Además puede utilizarse con modelos en espacio de estado multivariables y con los adelantos tecnológicos hoy en día, la aplicación del MPC a un sistema no supone un gran costo computacional puesto que las velocidades de los procesadores son lo suficientemente elevadas y además la aparición de los procesadores de varios núcleos permiten paralelizar los cálculos y así reducir aún más el costo computacional.

El objetivo principal del MPC es *predecir* una secuencia de control que nos permita optimizar el comportamiento futuro de un sistema. Dicha secuencia de control está confinada a un intervalo de tiempo determinado, es decir, la predicción se realiza dentro de una ventana variante en el tiempo y durante un intervalo temporal constante. Este intervalo es el horizonte de predicción h_p ; es el parámetro que nos indica cuanto queremos «ver» a futuro. Otro parámetro muy importante es el horizonte de control h_u , el cual nos indica durante cuanto tiempo vamos a aplicar la señal de control al sistema.

Nuestro sistema está representado por el modelo longitudinal de un avión (Stevens y Lewis, 2003). Para la implementación del MPC, utilizaremos una representación en espacio de estado con múltiples entradas y múltiples salidas (SS-MIMO). Este modelo presenta tres grados libertad: traslación, pitch y movimiento en el plano vertical y es representativo de un avión de tamaño mediano en condición de vuelo de baja velocidad. La fuerza de sustentación es calculada mediante una función lineal y la condición de pérdida no se ha tenido en cuenta. El arrastre se obtiene utilizando una curva no lineal. El empuje proporcionado por el motor decrece linealmente con la velocidad del aire. Como la fuerza de empuje no es aplicada en el centro de gravedad del avión, cambios de aceleración tienden a causar el movimiento de cabeceo o pitch. El modelo del avión es un sistema no lineal. Para poder utilizar MPC, realizamos una linealización de dicho sistema y exploramos tres casos particulares. En primer lugar vemos que sucede si consideramos al sistema no lineal como un sistema lineal; es decir, predecimos la secuencia de control óptima y luego la aplicamos al modelo linealizado. En segundo lugar, utilizamos los estados reales provenientes del sistema no lineal para realizar la optimización. Y por último, linealizamos al sistema en varios puntos, predecimos las variables de control y las aplicamos al sistema real. Debe quedar claro que en las tres variantes mencionadas los controles son determinados matemáticamente por resolución continua de las ecuaciones del MPC. En ningún caso los puntos de linealización están asociados a una reparametrización de reguladores típicamente usados en enfoques tipo *gain scheduling*.

2. MODELADO DEL SISTEMA

La representación en espacio de estados de nuestro sistema (Stevens y Lewis, 2003) es la siguiente:

$$\begin{aligned}
 \dot{v}_t &= \frac{(t_{stat} + v_t \cdot dt dv) \cdot thtl \cdot \cos \alpha - Q_s \cdot (Dcdg + C_{d0} + C_{dcls} \cdot (C_{l0} + C_{la} \cdot \alpha)^2)}{am \cdot v_t + Q_s \cdot C_{ladot}} - g \cdot \sin \gamma \\
 \dot{\alpha} &= \frac{-(t_{stat} + v_t \cdot dt dv) \cdot thtl \cdot \sin \alpha - Q_s \cdot (C_{l0} + C_{la} \cdot \alpha) + am \cdot (v_t \cdot Q + g \cdot \cos \gamma)}{am \cdot v_t + Q_s \cdot C_{ladot}} \\
 \dot{\theta} &= Q \\
 \ddot{\theta} &= \frac{(Q_s \cdot C_{bar} \cdot (Dcmg + C_{m0} + C_{ma} \cdot \alpha + C_{mde} \cdot elev) + (C_{l0} + C_{la} \cdot \alpha) \cdot (x_{cg} - 0,25) + D) + (t_{stat} + v_t \cdot dt dv) \cdot thtl \cdot z_e}{A_{iyy}} \\
 \dot{h} &= v_t (\cos \alpha \cdot \sin \theta - \sin \alpha \cdot \cos \theta) \\
 \dot{x} &= v_t (\cos \alpha \cdot \cos \theta + \sin \alpha \cdot \sin \theta)
 \end{aligned} \tag{1}$$

donde:

- v_t es la velocidad en $[\frac{rad}{seg}]$
- α es el ángulo de ataque en $[rad]$
- θ es el ángulo de pitch en $[rad]$
- $\dot{\theta}$ es la velocidad de pitch en $[\frac{rad}{seg}]$
- h es la altitud en $[ft]$
- x es el desplazamiento horizontal $[ft]$
- $D = \frac{1}{2} \cdot \frac{C_{bar} \cdot (C_{mq} \cdot Q + C_{madot} \cdot \dot{\alpha})}{v_t}$, es la fuerza de arrastre
- $Dcdg$, C_{d0} y C_{dcls} son constantes utilizadas para determinar el coeficiente de arrastre C_D
- g es la aceleración de la gravedad en $[\frac{ft}{seg^2}]$
- γ es el ángulo de trepada, $\gamma = (\theta - \alpha)$
- Q_s es el producto de la presión dinámica y la superficie alar
- $thtl$ es la deflexión de la columna de aceleración, la misma se encuentra normalizada. El valor físico es el empuje $T = (t_{stat} + v_t \cdot dt dv) \cdot thtl$
- am es el número de Mach
- C_{ladot} es la constante que describe el aumento de la sustentación debido a $\dot{\alpha}$
- Q es la velocidad angular del avión en el eje y_{body}
- C_{bar} , C_{madot} , C_{mq} son las constantes que describen la fuerza de arrastre
- $Dcmg$, C_{m0} , C_{ma} , C_{mde} son constantes utilizadas para obtener el coeficiente de cabeceo C_m
- x_{cg} es la posición del centro de masa en el eje x_{body}
- $elev$ es la deflexión del elevador del avión expresada en grados
- z_e es la distancia del punto de aplicación del trust al centro de gravedad

Para facilitar la comprensión y la lectura de las ecuaciones (1), en la figura siguiente puede observarse un esquema del sistema modelado. El sistema de ejes *cuerpo* (body) es una terna que se mueve solidariamente con el avión, el eje x_{body} se alinea apuntando a la nariz del avión, el eje el eje y_{body} con el ala derecha del mismo y el eje el eje z_{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_{wind} se alinea con la dirección del viento relativo.

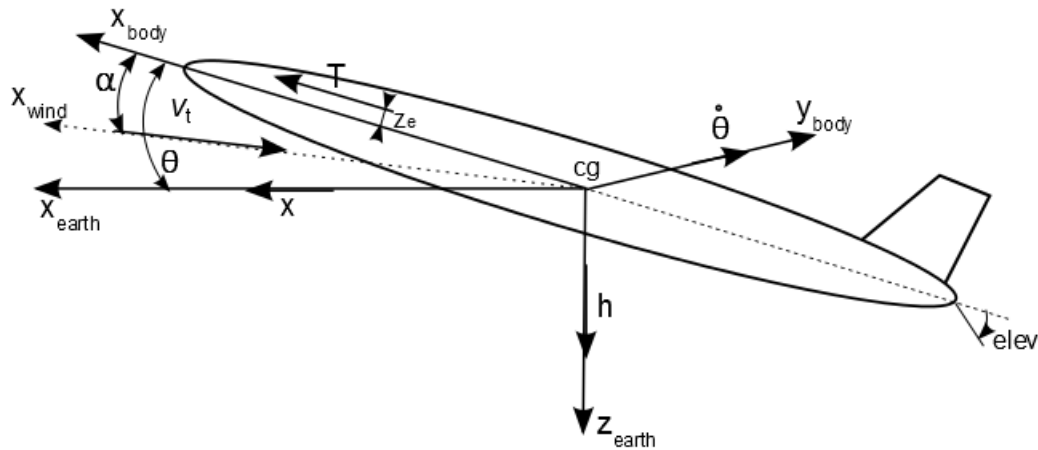


Figura 1: Esquema del avión

Definiendo

$$\bar{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \quad \bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} v_t \\ \alpha \\ \theta \\ \dot{\theta} \\ h \\ x \end{bmatrix} \quad \bar{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} thtl \\ elev \end{bmatrix}$$

donde f_1, f_2, \dots, f_6 representan las seis funciones del lado derecho de la ecuación (1).

Desarrollando (1) en serie de Taylor de primer orden alrededor de (\bar{x}_e, \bar{u}_e) (Maciejowski, 2002)

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

donde

$$\frac{\partial \bar{f}}{\partial \bar{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_6} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_6}{\partial x_1} & \frac{\partial f_6}{\partial x_2} & \dots & \frac{\partial f_6}{\partial x_6} \end{bmatrix} \quad \text{y} \quad \frac{\partial \bar{f}}{\partial \bar{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \vdots & \vdots \\ \frac{\partial f_6}{\partial u_1} & \frac{\partial f_6}{\partial u_2} \end{bmatrix}$$

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

$B_u = \frac{\partial \bar{f}}{\partial \bar{u}}$, $\bar{d} = \bar{f}(\bar{x}_e, \bar{u}_e)$, $\delta \bar{x} = (\bar{x} - \bar{x}_e)$ y $\delta \bar{u} = (\bar{u} - \bar{u}_e)$
Podemos reescribir (2) como

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

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

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

Donde B_{dd} es la matriz identidad discreta.

En general, nos interesa minimizar las variaciones en la entrada de control (Maciejowski, 2002), (Rossiter, 2003). Si definimos

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

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

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

Finalmente, podemos expresar (6) como

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

donde

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

\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 ecuación de salida es

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

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

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

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

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

donde

$$\delta\vec{X} = \begin{bmatrix} \delta\vec{x}(k+1) \\ \delta\vec{x}(k+2) \\ \delta\vec{x}(k+3) \\ \vdots \\ \delta\vec{x}(k+h_p) \end{bmatrix} \quad \Delta\delta\vec{U} = \begin{bmatrix} \Delta\delta\vec{u}(k) \\ \Delta\delta\vec{u}(k+1) \\ \Delta\delta\vec{u}(k+2) \\ \vdots \\ \Delta\delta\vec{u}(k+h_c-1) \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \tilde{A}^3 \\ \vdots \\ \tilde{A}^{h_p} \end{bmatrix} \quad H_u = \begin{bmatrix} \tilde{B}_u & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{A}_t\tilde{B}_u & \tilde{B}_u & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{A}_t^2\tilde{B}_u & \tilde{A}_t\tilde{B}_u & \tilde{B}_u & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_t^{h_p-1}\tilde{B}_u & \tilde{A}_t^{h_p-2}\tilde{B}_u & \tilde{A}_t^{h_p-3}\tilde{B}_u & \cdots & \tilde{A}_t^{h_p-h_c}\tilde{B}_u \end{bmatrix}$$

$$\vec{D}(k) = \begin{bmatrix} \vec{d}(k) \\ \vec{d}(k+1) \\ \vec{d}(k+2) \\ \vdots \\ \vec{d}(k+h_p-1) \end{bmatrix} \quad H_d = \begin{bmatrix} \tilde{B}_d & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{A}_t\tilde{B}_d & \tilde{B}_d & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{A}_t^2\tilde{B}_d & \tilde{A}_t\tilde{B}_d & \tilde{B}_d & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_t^{h_p-1}\tilde{B}_d & \tilde{A}_t^{h_p-2}\tilde{B}_d & \tilde{A}_t^{h_p-3}\tilde{B}_d & \cdots & \tilde{B}_d \end{bmatrix}$$

3. FUNCIÓN OBJETIVO Y SIMULACIONES MPC

3.1. Función objetivo cuadrática

Para poder predecir la secuencia de control óptima necesitamos de un criterio que refleje nuestro objetivo. En este caso utilizamos una función costo (o función objetivo) Obj que relaciona la diferencia entre la respuesta deseada y la respuesta predicha. La secuencia de control óptima la obtenemos minimizando dicha función objetivo (Wang, 2009).

Sea el vector ampliado de setpoints \vec{X}_{sp} . Para poder comparar con nuestro vector de estados ampliado predicho $\delta\vec{X}$, debemos escribir al setpoint como $\delta\vec{X}_{sp} = \vec{X}_{sp} - \vec{X}_e$ con $\vec{X}_e = [X_e \ U_e]^T$

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

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

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

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

De (12), podemos ver que el problema a resolver se reduce a minimizar una función objetivo cuadrática, donde la hessiana está definida por $H_{ess} = (R + H_u^T \cdot \tilde{Q} \cdot H_u)$ y el jacobiano por $J_{ac} = 2(\delta\vec{x}(k)^T \cdot \Gamma^T - \delta\vec{X}_{sp}^T + \vec{D}^T \cdot H_d^T) \cdot \tilde{Q} \cdot H_u$.

Para resolver (12) utilizamos la función *quadprog* de Matlab, la cual resuelve problemas del tipo

$$\begin{aligned} \text{Min} \quad & \frac{1}{2} \cdot \vec{Y}^T \cdot H \cdot \vec{Y} + \vec{f}^T \cdot \vec{Y} \\ \text{s.a} \quad & A \cdot \vec{Y} \leq \vec{b} \\ & \hat{A} \cdot \vec{Y} = \vec{\hat{b}} \\ & \underline{Y} \leq Y \leq \bar{Y} \end{aligned} \quad (13)$$

3.2. Ejemplo: uso del MPC para control automático de velocidad

Una de las maniobras más comunes que pueden llevarse a cabo durante el vuelo de un avión es el aumento de la velocidad manteniendo la altitud constante. Supongamos que el avión se encuentra volando en equilibrio a $h = 30000 \text{ ft}$ con una velocidad $v_t = 500 \frac{\text{ft}}{\text{seg}}$ y queremos que el avión realice la siguiente maniobra: aumentar su velocidad a $v_t = 550 \frac{\text{ft}}{\text{seg}}$ manteniendo la altitud constante en 30000 ft . Para las simulaciones utilizamos $h_p = 18$ (9 seg) y $h_u = 15$ (7,5 seg). Estos valores se obtuvieron empíricamente, siendo la limitación el tamaño de las matrices de predicción (Γ y H_u) para evitar el mal condicionamiento de las mismas. Además, más allá de los valores utilizados, el aumento de dichos horizontes no produjo cambios significativos en la respuesta del sistema. Las matrices de peso utilizadas son

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{y} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Puede observarse que la matriz Q pesa exclusivamente los desvíos de velocidad y altitud, no siendo necesario agregar otro tipo de penalización a los estados. La matriz R tiene por objetivo penalizar los desvíos de ambas entradas de control: *thtl* y *elev*. Empíricamente observamos que los factores de escala unitarios son suficientes para controlar el sistema con variaciones físicas razonables tanto de los estados \vec{x} como de los controles \vec{u} .

En lo que sigue centraremos nuestro interés en obtener la secuencia de control óptima que nos permita realizar la maniobra ya descrita.

3.2.1. Control del sistema linealizado

En la condición de vuelo recto y nivelado, los estados toman el siguiente valor

$$\vec{x}_e = [500 \quad 5,43 \cdot \pi/180 \quad 5,43 \cdot \pi/180 \quad 0 \quad 30000 \quad 0]^T \quad (14)$$

mientras que la entrada de control es

$$\vec{u}_e = [0,204 \quad -4,1]^T \quad (15)$$

Considerando el modelo (1), lo linealizamos alrededor de (14) y (15) y luego lo discretizamos para poder utilizar el control predictivo.

Las restricciones que consideramos en las entradas de control son

- $0 \leq \text{thtl} \leq 1$

- $-30^\circ \leq elev \leq 30^\circ$
- $-0,2 \leq \Delta thtl \leq 0,2$
- $-1^\circ \leq \Delta elev \leq 1^\circ$

y las restricciones en las variables de estado son

- $0 \leq v_t \leq 600$
- $-\frac{12 \cdot \pi}{180} \leq \alpha \leq \frac{12 \cdot \pi}{180}$
- $-\frac{12 \cdot \pi}{180} \leq \theta \leq \frac{12 \cdot \pi}{180}$
- $-\frac{2 \cdot \pi}{180} \leq \dot{\theta} \leq \frac{2 \cdot \pi}{180}$
- $0 \leq h \leq 35000$
- $0 \leq x \leq \infty$

La respuesta del sistema (7) a la entrada de control hallada para el sistema linealizado puede verse en las figuras 2(a) - 2(d). La respuesta del sistema es correcta pudiendo el mismo alcanzar

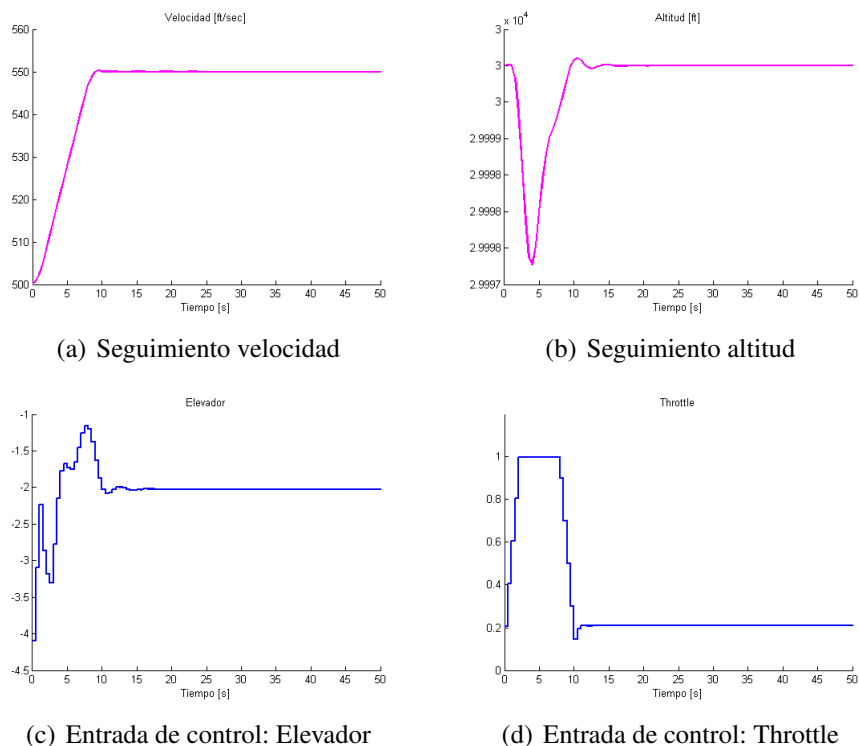
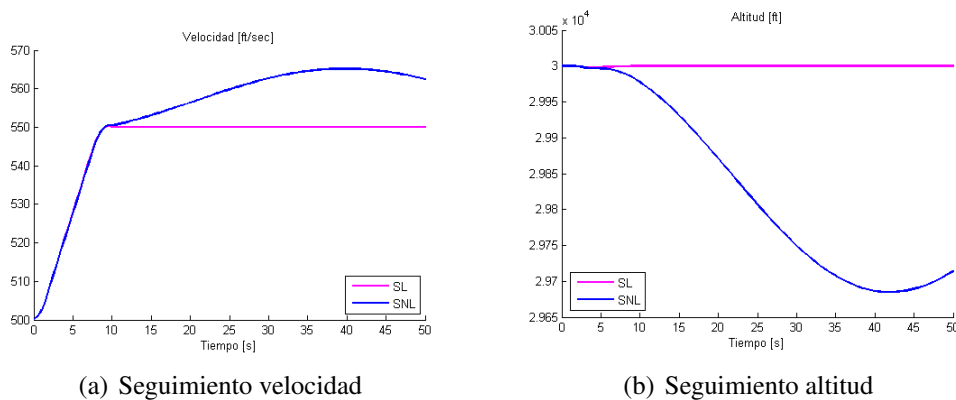


Figura 2: Respuesta del sistema linealizado

el valor de velocidad deseado como así también mantener su altitud constante. Sin embargo, el sistema linealizado no es una buena estimación del sistema real lejos del punto de linealización. Podemos ver como responde este último a la secuencia de controles hallada para el sistema linealizado. Para ello integramos numéricamente (1) y simulamos su respuesta. En las figuras 3(a) y 3(b), vemos que la respuesta del sistema no lineal a los controles obtenidos para el sistema linealizado no es buena, sólo coinciden prácticamente en el entorno del punto de linealización.



(a) Seguimiento velocidad

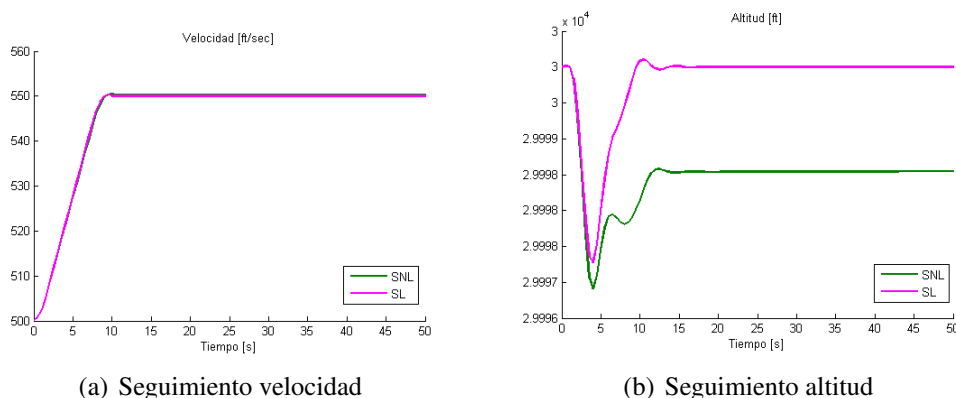
(b) Seguimiento altitud

Figura 3: Comparación de respuestas sistema linealizado y sistema no lineal

3.2.2. Control del sistema linealizado con corrección de función objetivo

Como J_{ac} depende de $\delta\vec{x}(k)$ y conocemos el sistema real no lineal, podemos integrar (1) y actualizar el valor de J_{ac} con el estado real, es decir, manteniendo las matrices de predicción constantes e iguales que para el caso linealizado, obtenemos la secuencia de control óptima y la aplicamos a (1). Integrando este resultado, obtenemos el valor de los estados en el siguiente instante de tiempo. Es éste el valor que utilizamos para actualizar la matriz jacobiana y seguir optimizando la función objetivo.

Los resultados obtenidos se muestran en las figuras 4(a) y 4(b). Comparando con el sistema linealizado, si bien existe un pequeño offset en estado estacionario, el seguimiento tanto de velocidad como de altitud es mucho mejor que para el caso anterior. Aquí también la respuesta del sistema linealizado y la del sistema no lineal coinciden en la proximidad del punto de linealización, pero en ambos casos los estados alcanzan un equilibrio.



(a) Seguimiento velocidad

(b) Seguimiento altitud

Figura 4: Respuesta del sistema no lineal actualizando estados

3.2.3. Control del sistema linealizado en diferentes puntos con corrección de función objetivo

Otra posibilidad para mejorar los resultados obtenidos en las figuras 4(a) - 4(b) es linealizar el sistema en determinados puntos a lo largo de la trayectoria. En nuestro caso, realizamos las

sucesivas linealizaciones en cada instante de muestreo. Hay que destacar que al realizar esto es necesario actualizar todas las matrices del sistema, las matrices de predicción, la matriz hessiana y la matriz jacobiana, lo cual implica un aumento en la carga computacional.

En las figuras 5(a) - 5(d) puede verse que la respuesta del sistema no lineal (con corrección de la función objetivo) a la secuencia de controles obtenida para el sistema linealizado en diferentes puntos es buena. En este caso particular no notamos diferencias entre las respuestas del sistema linealizado y del sistema no lineal. Sin embargo, para otros sistemas el hecho de contar con la información del sistema real puede ser de mucha utilidad pudiendo mejorar los resultados al no tratar con sistemas linealizados.

Analizando las figuras 5(a) - 5(d) vemos que existe un aumento tanto de la potencia (*thtl*) como del ángulo del elevador (*elev*). Al aumentar la potencia se crea un momento de cabeceo en el avión que hace que la nariz del mismo se eleve. En nuestro caso como queremos mantener altitud constante, para contrarrestar este efecto el elevador se mueve hacia abajo con lo cual existe una fuerza de sustentación mayor en la cola del avión produciendo un efecto nariz abajo disminuyendo el ángulo de ataque. A medida que la velocidad va aumentando, ambas entradas de control se mueven en forma conjunta para permitir el incremento de velocidad a altitud constante llegando finalmente a un punto de equilibrio.

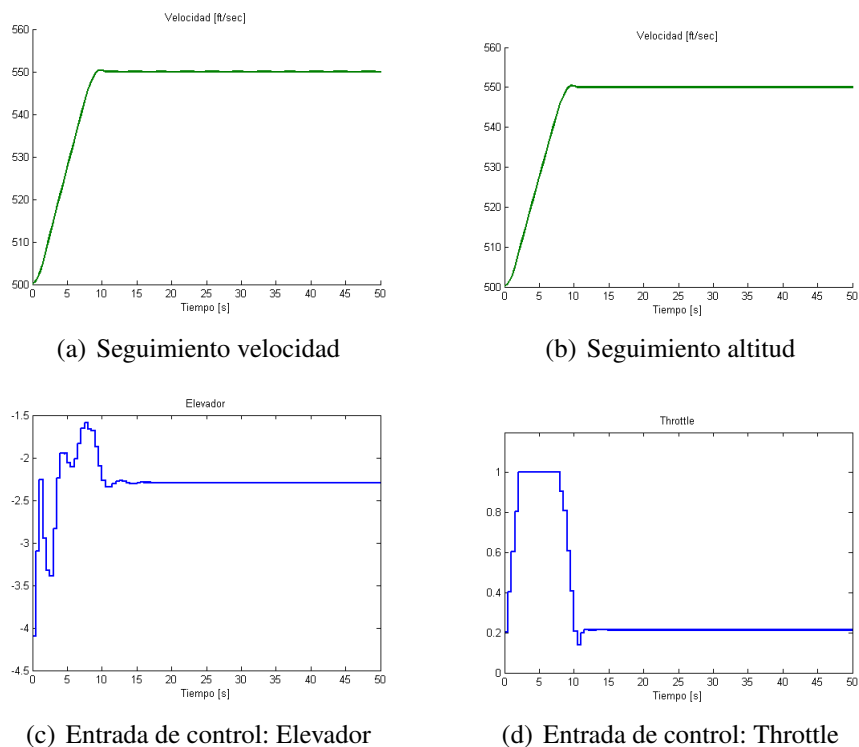


Figura 5: Respuesta del sistema no lineal actualizando estados

4. CONCLUSIONES

Mediante la realización de este trabajo pudimos desarrollar un sistema de control simple que permite realizar maniobras a un modelo longitudinal de un avión. Partiendo de un modelo linealizado fuimos explorando distintas configuraciones para la obtención de la secuencia de variables manipuladas. Como los sistemas reales son no lineales, es necesario contar con

herramientas que nos permitan «manejar» estos sistemas. La utilización del MPC nos permitió explorar los resultados obtenidos al aplicarle al sistema real no lineal el control óptimo predicho para el sistema linealizado, cuando utilizamos el sistema no lineal dentro de la optimización para la obtención de los controles los resultados fueron bastante buenos, y así evitamos linealizar el modelo en distintos puntos de la trayectoria reduciendo así la carga computacional. Simulando distintos cambios de setpoints, pudimos verificar que si bien los autovalores del sistema continuo se desplazan hasta un 20 % del valor inicial, aún en estas condiciones la aproximación del sistema no lineal por el sistema linealizado alrededor de un único punto y actualizando los estados con el sistema real, el seguimiento de las referencias también es bastante bueno. Nuestro objetivo es el desarrollo de técnicas de control más complejas para aplicar a modelos más completos, con lo cual este trabajo nos sirve como paso inicial.

REFERENCIAS

- 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.
- Borggaard J., Burkardt J., Gunzburger M., y Peterson J. *Optimal Design and Control*. Proceedings of the Workshop on Optimal Design and Control, 1994.
- Bryson A.E.J. New concepts in control theory. *J. Guid.*, 8:417–425, 1985.
- Bryson A.E.J. y Ho Y.C. *Applied Optimal Control*. Taylor & Francis, 1975.
- Giovanini L. Predictive feedback control. *Instruments Society of America Transaction Journal*, 42(2):207–226, 2003.
- Hilscher R. y Zeidan V. Discrete optimal control: The accessory problem and necessary optimality conditions. *Journal of Mathematical Analysis and Applications*, 243:429–452, 2000.
- Kouvaritakis B. y Cannon M. *Nonlinear predictive control: theory and practice*. 2001.
- Lewis F. *Applied optimal control and estimation*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1992.
- Lewis F. y Syrmos V. *Optimal control*. Wiley-Interscience, 1995.
- Maciejowski J. *Predictive control: with constraints*. Prentice Hall, 2002.
- Mayne D., Rawlings J., Rao C., y Sokaert P. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.
- 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.
- Wang L. *Model Predictive Control System Design and Implementation Using Matlab*. Springer, 2009.