

C12

## CONSTRUCCIÓN DE UNA LIBRERÍA DE MECÁNICA DEL VUELO USANDO ECOSIMPRO

Borja García Gutiérrez y Ramón Pérez Vara, Empresarios Agrupados  
Magallanes, 3 28015 Madrid - España  
bgg@ecosimpro.com - rpv@ecosimpro.com

### Resumen

*Se ha desarrollado en EcosimPro una librería de componentes para construir de forma modular modelos de mecánica del vuelo de aviones y otras aeronaves. Dicha librería permite calcular las actuaciones y la respuesta dinámica de la aeronave teniendo en cuenta los posibles autopilotos. A fin de verificar los componentes de la librería se han construido un modelo de F-16 y dos modelos de autopiloto, con los cuales se han calculado las actuaciones de este avión y las respuestas dinámicas del mismo con y sin autopilotos.*

**Palabras Clave:** Simulación, Ecosimpro, modelado de sistemas físicos orientado a objetos, mecánica del vuelo, sistemas de control y autopiloto.

### 1 INTRODUCCIÓN

El propósito de este trabajo es desarrollar una librería de componentes de EcosimPro que permita construir de forma modular modelos para calcular las actuaciones de aviones y otras aeronaves, así como su respuesta dinámica en maniobra.

Pero antes de entrar en los detalles propios de la elaboración de una librería de estas características en EcosimPro parece necesario hacer al menos una leve descripción del modelo que se va a abordar.

En primer lugar se fijarán las hipótesis sobre el modelo físico-matemático de avión que se va a simular y que nos proporcionará las ecuaciones generales del movimiento. Se considera un avión sólido y rígido sobre el cual las únicas fuerzas que actúan son las gravitatorias, las aerodinámicas y las propulsivas.

Con estas hipótesis se está en disposición de plantear las ecuaciones que determinan la evolución del sistema, supuestas conocidas las condiciones iniciales y una expresión de las fuerzas que intervienen en el modelo.

Para ello se debe elegir unos ejes de trabajo, por lo que, en segundo lugar, se van a describir los dos

sistemas que se utilizan para representar el estado de la aeronave. Se utiliza un primer sistema de ejes denominado Ejes Cuerpo con origen ( $O_b$ ) en el centro de masas de la aeronave, eje  $O_bX$  perteneciente al plano de simetría de ésta y dirigido hacia adelante, eje  $O_bZ$  perteneciente al plano de simetría también y dirigido hacia abajo en actitud normal de vuelo y eje  $O_bY$  tal que forme un triedro a derechas. Y un segundo sistema denominado Ejes Tierra con origen ( $O_e$ ) en un punto de la superficie terrestre, eje  $O_eN$  perteneciente al plano horizontal y dirigido hacia el norte, eje  $O_eE$  perteneciente al plano horizontal también y dirigido hacia el este y eje  $O_eD$  perpendicular al plano horizontal y dirigido hacia abajo. Para entender la orientación relativa de estos ejes véase la figura 1.

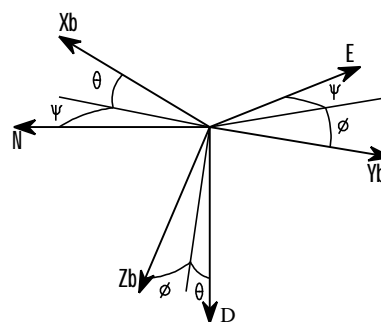


Figura 1: Orientación relativa entre Ejes Cuerpo y Ejes Tierra

Posición relativa que determinará, como se verá más adelante, la actitud de la aeronave respecto a tierra mediante los ángulos  $\psi$  (ángulo de guiñada),  $\theta$  (ángulo de asiento) y  $\phi$  (ángulo de balance).

En tercer lugar debe añadirse una hipótesis sobre el modelo de atmósfera a utilizar para el cálculo de las propiedades del aire en función de la altura de vuelo y que será la atmósfera estándar internacional (ISA).

Y en último lugar se debería aclarar el alcance de los modelos de control y autopiloto. Todos responden a diseños típicos de CAS (Control Augmentation Systems), los cuales comandarán las superficies de control de la aeronave, que se han modelado como actuadores con retardo de primer orden y limitados en deflexión y en velocidad de deflexión.

## 2 MODELO FÍSICO-MATEMÁTICO DE LA AERONAVE.

Como ya se introdujo, nuestra aeronave será considerada un sólido rígido, por tanto, con seis grados de libertad (aunque por una serie de razones que escapan al alcance de este texto éstos quedarán reducidos a cuatro y que se denominarán parámetros de control de la aeronave). Se pueden plantear entonces las seis ecuaciones del movimiento de un sólido rígido en el caso de estar sometido a los tres tipos de fuerzas ya descritas. Como se hará este desarrollo en Ejes Cuerpo, no inerciales, será necesario incluir los términos de inercia en las ecuaciones.

$$m \cdot \left( \dot{\vec{V}} + \vec{\omega}_{be} \wedge \vec{V} \right) = \vec{F}_G + \vec{F}_A + \vec{F}_T \quad (1)$$

$$\dot{\vec{I}} \cdot \dot{\vec{\omega}}_{be} + \vec{\omega}_{be} \wedge \left( \vec{I} \cdot \vec{\omega}_{be} \right) = \vec{M}_G + \vec{M}_? + \vec{M}_? \quad (2)$$

Además se deben incluir las ecuaciones necesarias para obtener, a partir de las velocidades lineal y angular, la posición y la actitud de la aeronave respecto a Tierra. Para ello será necesario dar las relaciones cinemáticas que determinan la actitud en función de la velocidad angular y la matriz de cambio de sistema de referencia que nos permita calcular la velocidad de la aeronave en Ejes Tierra en función de la actitud.

### 2.1 FUERZAS GRAVITATORIAS

Para el cálculo de las fuerzas gravitatorias en nuestro modelo se supondrá que la gravedad es una constante independiente de la posición de la aeronave, lo que implica que se trabaja con la hipótesis de Tierra plana.

Teniendo en cuenta la anterior consideración sólo será necesario proyectar la fuerza de la gravedad sobre Ejes Cuerpo, sabido que es una fuerza aplicada en la dirección del eje  $O_e D$  y de módulo el producto de la masa de la aeronave ( $m$ ) por la aceleración de la gravedad ( $g$ ), obteniéndose

$$\begin{aligned} F_{xG} &= -m \cdot g \cdot \text{sen } ? \\ F_{yG} &= m \cdot g \cdot \text{sen } f \cdot \text{cos } ? \\ F_{zG} &= m \cdot g \cdot \text{cos } f \cdot \text{cos } ? \end{aligned} \quad (3)$$

Si además se fija el centro de masas como origen para el cálculo de los momentos sobre la aeronave no existirá momento alguno sobre ésta debido al campo gravitacional, por lo que

$$\vec{M}_G = \vec{0} \quad (4)$$

### 2.2 FUERZAS AERODINÁMICAS

Para el cálculo de las componentes de la fuerza y el momento aerodinámicos se dispone de un conjunto de tablas que proporcionan cada uno de los coeficientes aerodinámicos como función de las condiciones de vuelo y de los ángulos de deflexión de las superficies de control. Éstas vienen dadas por las expresiones

$$\begin{aligned} F_{xA} &= q \cdot S \cdot C_{xT} \\ F_{yA} &= q \cdot S \cdot C_{yT} \\ F_{zA} &= q \cdot S \cdot C_{zT} \end{aligned} \quad (5)$$

$$\begin{aligned} M_{xA} &= q \cdot S \cdot b \cdot C_{lT} \\ M_{yA} &= q \cdot S \cdot c \cdot C_{mT} \\ M_{zA} &= q \cdot S \cdot b \cdot C_{nT} \end{aligned} \quad (6)$$

donde  $q$  es la presión dinámica,  $S$  la superficie alar,  $b$  la envergadura del ala,  $c$  la cuerda media del ala y los coeficientes de fuerza y de momento totales se obtienen como

$$\begin{aligned} C_{xT} &= C_{xT}(a, q, de) \\ C_{yT} &= C_{yT}(a, \beta, p, r, da, dr) \\ C_{zT} &= C_{zT}(a, \beta, q, de) \end{aligned} \quad (7)$$

$$\begin{aligned} C_{lT} &= C_{lT}(a, \beta, p, r, da, dr) \\ C_{mT} &= C_{mT}(a, p, de, Xr, Xcg) \\ C_{nT} &= C_{nT}(a, \beta, p, r, da, dr, Xr, Xcg) \end{aligned} \quad (8)$$

donde  $a$  es el ángulo de ataque,  $\beta$  el ángulo de deslizamiento,  $p$ ,  $q$  y  $r$  las componentes de la velocidad angular de la aeronave en Ejes Cuerpo,  $d_e$ ,  $d_a$  y  $d_r$  las deflexiones de las superficies de control (elevador, alerones y deriva, respectivamente) y  $X_R$  y  $X_{CG}$  las posiciones nominal y real del centro de masas adimensionalizadas con la cuerda del ala.

Este modelo tiene unos intervalos de validez para las variables limitados por los valores disponibles en las tablas y es por esto que en algún momento se deberán hacer notar las siguientes restricciones

$$\begin{aligned} a &\in [-15^\circ, 45^\circ] \\ \beta &\in [-30^\circ, 30^\circ] \\ de &\in [-25^\circ, 25^\circ] \\ da &\in [-21'5^\circ, 21'5^\circ] \\ dr &\in [-21^\circ, 21^\circ] \end{aligned} \quad (9)$$

### 2.3 FUERZAS PROPULSIVAS

Al igual que para el caso de la fuerza aerodinámica, para el caso de la fuerza propulsiva se tienen unas tablas que determinan el empuje en función de las condiciones de vuelo y el parámetro de control del motor. A efecto de todos los cálculos se supone que la dirección de aplicación del empuje coincide con el eje  $O_bX$  por lo que

$$\begin{aligned} F_{xT} &= F_{xT}(M_\infty, h, p) \\ F_{yT} &= 0 \\ F_{zT} &= 0 \end{aligned} \quad (10)$$

donde  $M_\infty$  es el Mach de vuelo de la aeronave,  $h$  es la altura de vuelo y  $p$  el parámetro de control del motor.

Según las hipótesis hechas, el empuje no genera momento alguno respecto del centro de masas de la aeronave, aunque si se considerará un cierto momento angular (H) constante debido al giro del motor.

$$\begin{aligned} M_{xT} &= 0 \\ M_{yT} &= -r \cdot H \\ M_{zT} &= q \cdot H \end{aligned} \quad (11)$$

### 3 MODELO FÍSICO-MATEMÁTICO DE LA ATMÓSFERA

El modelo de atmósfera a utilizar es el denominado modelo ISA (International Standar Atmosphere) para el cual se suponen las siguientes hipótesis: el aire es un gas ideal, las condiciones a nivel del mar son ( $T_0=288K$  y  $p_0=101.325Pa$ ) y es conocido el perfil de temperaturas como función continua de la altura y, por tanto, el de presiones y el de densidades. De estas consideraciones se desprenden los valores de la temperatura (T), la presión (p) y la densidad ( $\rho$ ) del aire y la velocidad del sonido (a) a cualquier altura

$$\rho = \frac{p}{R \cdot T} \quad (12)$$

$$T = \begin{cases} T_0 + a_0 \cdot h \\ T_{11} \\ T_{11} + a_{11} \cdot h \end{cases} \quad (13)$$

$$p = \begin{cases} p_0 \cdot \left[ 1 + \left( \frac{a_0}{T_0} \cdot h \right) \frac{-g}{R \cdot a_0} \right] \\ p_{11} \cdot \exp \left[ \frac{-g}{R \cdot T_0} \cdot (h - 11.000) \right] \\ p_{11} \cdot \left\{ 1 + \left[ \frac{a_{11}}{T_{11}} \cdot (h - 20.000) \right] \frac{-g}{R \cdot a_{11}} \right\} \end{cases} \quad (14)$$

$$a = \sqrt{\gamma \cdot R \cdot T} \quad (15)$$

donde los subíndices 0 y 11 indican los valores de la variable a nivel del mar y 11.000 m,  $a$  es el gradiente de temperatura con la altura,  $R$  es la constante de los gases para el aire y las expresiones de la temperatura y la presión están dadas para los tres intervalos siguientes: desde el nivel del mar hasta los 11.000m, desde los 11.000m hasta los 20.000m y para más de 20.000m.

### 4 MODELO FÍSICO-MATEMÁTICO DE LOS SISTEMAS DE CONTROL Y AUTOPILOTO

Con objeto de simular algunas maniobras que puedan resultar de interés se van a modelar dos autopilotos complementarios. Un autopiloto longitudinal que actúe sobre los movimientos en el plano de simetría de la aeronave (control de la velocidad angular de cabeceo) y otro lateral-direccional que actúe sobre los movimientos fuera del plano de simetría (control de la velocidad angular de balance y de guiñada).

El esquema detallado de estos sistemas se verá cuando se realice el modelado para EcosimPro. Por el momento se limitará a una representación que permita entender cuales son las variables de entrada de los sistemas y cuales las de salida. Véase las figuras 2 y 3.

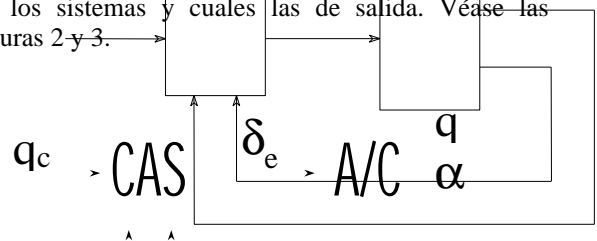


Figura 2: Autopiloto longitudinal

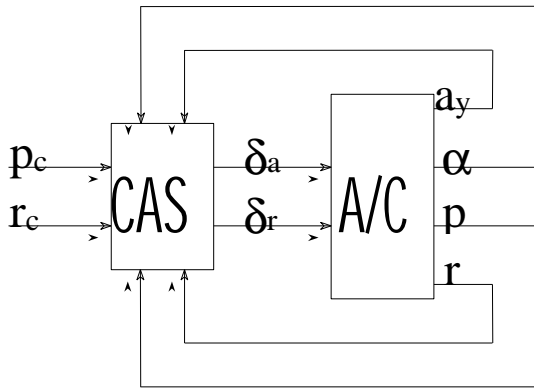


Figura 3: Autopiloto lateral-direccional

donde *A/C* representa la aeronave, *CAS* el sistema de control o autopiloto y  $p_c$ ,  $q_c$  y  $r_c$  las variables de control o comandadas.

### 5 MODELO FÍSICO-MATEMÁTICO DE LOS ACTUADORES

Para los actuadores se ha elegido un modelo sencillo de respuesta (retardo de primer orden) y además se consideran unos ciertos límites para el ángulo de deflexión y para la velocidad de deflexión, de tal forma que

$$d' = \frac{1}{t} \cdot (dc - d)$$

$$d \in [dmin, dmax] \quad (16)$$

$$d' \in [d' min, d' max]$$

donde  $dc$  y  $d$  son los ángulos de deflexión comandado y real de la superficie de control y  $t$  el tiempo característico de respuesta del actuador.

### 6 GENERALIDADES DE LA LIBRERÍA FLIGHT\_SIM

En este apartado se van a tratar aquellos elementos previos que son necesarios implementar para abordar el desarrollo de una librería de estas características. Esto es, definición de las coordenadas de los distintos ejes de referencia, constantes, puertos y funciones no ligadas directamente al modelo.

#### 6.1 ELEMENTOS COMUNES

Todos estos elementos están incluidos en el fichero *FLIGHT\_common.el*, como es usual en toda librería de EcosimPro.

Para poder manejar variables vectoriales se definen dos conjuntos de coordenadas, unas para Ejes Cuerpo (x,y,z) y otras para Ejes Tierra (N,E,D).

```
ENUM MovilAxis = {x, y, z}
ENUM EarthAxis = {N, E, D}
```

Entre las constantes definidas cabe destacar la aceleración de la gravedad ( $g$ ), la constante de los gases para el aire ( $R$ ) y el cociente de calores específicos ( $\gamma$ ).

```
CONST REAL g = 9.806 "gravity acceleration (m/s**2)"
CONST REAL R = 287. "air gas constant (J/kg*K)"
CONST REAL gamma = 1.4 "ratio of air specific heat ()"
```

Y, por último, para pasar los valores de las fuerzas en el modelo y para tratar las variables de estado del sistema se han definido dos puertos: *Forces* y *State* respectivamente. El primero pasa dos variables *SUM* vectoriales correspondientes a la fuerza y al momento en Ejes Cuerpo, y el segundo trece variables *EQUAL* que describen toda la evolución del sistema: la posición respecto a Tierra, la velocidad y la aceleración en Ejes Cuerpo y en Ejes Tierra y la actitud respecto a tierra y respecto a la corriente incidente.

```
PORT Forces
SUM REAL F[MovilAxis] "external forces (N)"
SUM REAL M[MovilAxis] "external moments (N*m)"
END PORT

PORT State
EQUAL REAL alpha "angle of attack (rad)"
EQUAL REAL beta "sideslip angle (rad)"
EQUAL REAL VT "aircraft velocity (m/s)"
EQUAL REAL V[MovilAxis] "body axis velocity (m/s)"
EQUAL REAL Vdot[MovilAxis] "body axis accel. (m/s**2)"
EQUAL REAL W[MovilAxis] "body axis angular velocity (rad/s)"
EQUAL REAL Wdot[MovilAxis] "body axis angular accel. (rad/s**2)"
EQUAL REAL A[MovilAxis] "body axis absolute acceleration (m/s**2)"
EQUAL REAL psi "yaw angle (rad)"
EQUAL REAL theta "pitch angle (rad)"
EQUAL REAL phi "roll angle (rad)"
EQUAL REAL R[EarthAxis] "earth axis position (m)"
EQUAL REAL Rdot[EarthAxis] "earth axis velocity (m/s)"
END PORT
```

Todos estos puertos están definidos y son usados de tal manera que pasan las variables con unidades pertenecientes al sistema internacional como se indica en los comentarios del código.

#### 6.2 MODELO ATMOSFÉRICO

Todos los elementos necesarios están contenidos en el fichero *FLIGHT\_f\_Atm.el*.

El modelo esta compuesto de todas las funciones necesarias para evaluar la temperatura, la presión y la densidad del aire y la velocidad del sonido en función de la altura.

En definitiva, son funciones con un único argumento de entrada: la altura de vuelo, y que devuelven el valor de la propiedad que se desea calcular. Los nombres que se han dado a estas funciones son *TAtm*, *pAtm*, *rhoAtm* y *VsoundAtm*. Un ejemplo de estas funciones es

```

FUNCTION REAL rhoAtm (IN REAL altitude)
DECLS
  REAL T          "temperature (K)"
  REAL p          "pressure (Pa)"
BODY
  T = TAtm(altitude)
  p = pAtm(altitude)
RETURN p / ( R * T)
END FUNCTION

```

## 7 COMPONENTES PRINCIPALES DE LA LIBRERÍA FLIGHT\_SIM

Estos componentes son los que incluyen todas las características que definen nuestra aeronave y que darán lugar al modelo final de F-16.

Se debe aclarar antes de proseguir que parte de estos componentes son genéricos y permiten, por tanto, su reutilización en cualquier modelo que se pretenda simular pero otros son propios de la configuración del F-16 y deberán ser implementados para cada caso, aun así éstos sirven como ejemplo de aplicación.

### 7.1 COMPONENTE “Frame”

Es el componente básico de la librería, contiene todas las ecuaciones del movimiento de un sólido rígido y las necesarias para calcular las distintas variables de estado características del movimiento de la aeronave y está definido en el fichero FLIGHT\_Frame.el.

Es un componente definido con un puerto de entrada tipo *Forces* y otro de salida tipo *State*. Tales que por el puerto de entrada se le pasan los valores de las distintas fuerzas que actúan sobre el sólido y por el de salida devuelve el estado del mismo.

```

COMPONENT Frame
PORTS
  IN Forces forces_in  "external forces input"
  OUT State state_out  "aircraft state output"

```

Otra característica reseñable de este componente son sus datos, a través de los cuales se pueden cambiar las propiedades de inercia de nuestra aeronave (como son la masa y los momentos de inercia) lo que lo hace totalmente reutilizable.

```

DATA
  REAL mass = 10000.  "mass (kg)"
  REAL Ixx  = 10000.  "moment of inertia Ix (kg*m**2)"
  REAL Iyy  = 100000. "moment of inertia Iy (kg*m**2)"
  REAL Izz  = 100000. "moment of inertia Iz (kg*m**2)"
  REAL Ixy  = 0.      "product of inertia Pxy (kg*m**2)"
  REAL Ixz  = 1000.   "product of inertia Pxz (kg*m**2)"
  REAL Iyz  = 0.      "product of inertia Pyz (kg*m**2)"

```

### 7.2 COMPONENTE “F16Aerodynamics”

Es el componente encargado de obtener las fuerzas aerodinámicas que actúan sobre el modelo de F-16, por lo que es un componente específico, y está definido en el fichero FLIGHT\_F16Aerodynamics.el.

Contiene todas las ecuaciones necesarias para obtener las fuerzas en función de los coeficientes de fuerzas y momentos, tal y como se definieron en el apartado relativo al modelo físico-matemático.

Es un componente definido con cuatro puertos de entrada, uno tipo *State* y tres tipo *analog\_signal*, y dos puertos de salida, uno tipo *Forces* y otro tipo *analog\_signal*. Tales que por los puertos de entrada se le pasan las variables de estado de la aeronave y las deflexiones de las tres superficies de control y por los de salida devuelve las fuerzas y el Mach de vuelo.

Dado que todos los parámetros necesarios para la obtención del valor de los coeficientes vienen dados por el estado de la aeronave, excepto las deflexiones de las superficies de control, este componente tendrá tres grados de libertad: *de*, *da* y *dr*, los que fijarán tres de los cuatro parámetros de control de la aeronave.

Al igual que en el caso del componente *Frame* resulta interesante hacer alguna reseña sobre los datos de éste, que, además de todas las tablas para el cálculo de los coeficientes, incluye los valores de la cuerda, la envergadura y la superficie alar, y dos magnitudes adimensionales que nos determinan la posición nominal del centro de masas ( $X_R$ ) y la posición real ( $X_{CG}$ ), cuyos valores van a condicionar notablemente la estabilidad de la aeronave.

### 7.3 COMPONENTE “F16Engine”

Es el componente encargado de obtener las fuerzas propulsivas que actúan sobre el modelo de F-16, por lo que es también un componente específico, y está definido en el fichero FLIGHT\_F16Engine.el.

Contiene todas las ecuaciones necesarias para obtener las fuerzas en función de los niveles de potencia demandados (incluyendo las ecuaciones que dan los tiempos característicos de retardo) y las condiciones de vuelo, tal y como se definieron en el apartado relativo al modelo físico-matemático.

Es un componente definido con tres puertos de entrada, uno tipo *State* y dos tipo *analog\_signal*, y un puerto de salida tipo *Forces*. Tales que por los puertos de entrada se le pasan las variables de estado de la aeronave, el parámetro de control del motor y el Mach de vuelo y por el de salida devuelve las fuerzas.

Dado que, de igual manera, todos los parámetros necesarios para la obtención de las fuerzas vienen dados por el estado de la aeronave, excepto el parámetro de control del motor, este componente tendrá un sólo grado de libertad: *p*, el que fijará el cuarto de los parámetros de control de la aeronave.

Se hará a continuación alguna reseña sobre los datos de este componente, como se ha hecho para los

anteriores, y que en este caso incluye tres tablas de empuje para tres condiciones de empuje (ralentí, militar y máxima) en función de altitud y Mach de vuelo y el valor del momento angular del motor.

#### 7.4 COMPONENTE “Trimmer”

Es un componente que permite obtener el estado de la aeronave para unas determinadas condiciones de vuelo estacionarias. Como el sistema completo tiene cuatro grados de libertad bastará imponer cuatro condiciones independientes, como son la de vuelo coordinado (aceleración lateral nula) y las de velocidad de vuelo, ángulo de trepada y velocidad de viraje horizontal constantes. El componente *Trimmer* es un componente genérico válido, por tanto, para cualquier modelo y está definido en el fichero FLIGHT\_Trimmer.el.

Este componente es el más singular de esta librería y su justificación se encuentra en la necesidad de calcular unas determinadas condiciones iniciales estacionarias a partir de las cuales comenzar a simular cualquier maniobra. El hecho es que, para un sistema tan complejo e inestable, la determinación de estas condiciones iniciales no es nada trivial y por ello se requiere de este componente para que las imponga.

Contiene las cuatro ecuaciones necesarias para que se impongan las condiciones de vuelo estacionario o se mantengan los parámetros de control de la aeronave constantes, e iguales a los obtenidos al calcular las condiciones iniciales, según el valor que tome la variable global *Trim*. Variable a la cual se le asigna el valor 1 cuando se quiere calcular un estacionario y el valor 0 cuando se quiere propagar el sistema. Véase en el código la influencia de esta variable en las ecuaciones del bloque “*Stationary Conditions*”.

Es un componente definido con un puerto de entrada tipo *State* y cuatro de salida tipo *analog\_signal*. Tales que por el puerto de entrada se le pasan las variables de estado de la aeronave y por el de salida devuelve los valores de los cuatro parámetros de control para las condiciones estacionarias impuestas.

En este componente es a través de los tres datos que se definen: *VT\_req* (velocidad de vuelo requerida), *ClimbAngle\_req* (ángulo de trepada requerido) y *dpsi\_req* (ángulo de viraje requerido), como se controlan las condiciones iniciales de vuelo. Esto hace que se puedan modificar estos valores al usarlo para definir otros componentes o cada vez que lo se requiera en un experimento.

```
COMPONENT Trimmer
PORTS
  IN State state_in
  OUT analog_signal s_elevator
  OUT analog_signal s_aileron
  OUT analog_signal s_rudder
  OUT analog_signal s_throttle
```

```
DATA
  REAL VT_req = 100.      "required aircraft velocity (m/s)"
  REAL ClimbAngle_req = 0. "required climb angle (rad)"
  REAL dpsi_req = 0.     "required turn rate (rad/s)"
DECLS
  REAL alpha      "angle of attack (rad)"
  REAL beta      "sideslip angle (rad)"
  REAL VT        "aircraft velocity (m/s)"
  REAL V[MovilAxis] "body axis aircraft velocity (m/s)"
  REAL W[MovilAxis] "body axis angular velocity (m/s)"
  REAL psi      "yaw angle (rad)"
  REAL theta   "pitch angle (rad)"
  REAL phi     "roll angle (rad)"
  REAL ClimbAngle "climb angle (rad)"
  REAL dpsi    "derivative of the yaw angle (rad/s)"
  REAL elevator "deflection of the elevator (rad)"
  REAL aileron "deflection of the ailerons (rad)"
  REAL rudder  "deflection of the rudder (rad)"
  REAL throttle "throttle position [0., 1.] ()"
CONTINUOUS
  alpha = state_in.alpha
  beta = state_in.beta
  VT = state_in.VT
  V = state_in.V
  W = state_in.W
  psi = state_in.psi
  theta = state_in.theta
  phi = state_in.phi
  -- Climb angle for coordinated flight
  sin(ClimbAngle) = cos(alpha) * cos(beta) * sin(theta) - \
    ( sin(beta) * sin(phi) + sin(alpha) * cos(beta) * cos(phi) ) * \
    cos(theta)
  -- Derivative of the yaw angle
  dpsi = ( W[y] * sin(phi) + W[z] * cos(phi) ) / cos(theta)

  -- Stationary Conditions
  elevator' + Trim * ( ClimbAngle_req - ClimbAngle ) = 0.
  aileron' + Trim * ( dpsi_req - dpsi ) = 0.
  rudder' + Trim * ( W[x] * V[z] - W[z] * V[x] + \
    g * sin(phi) * cos(theta) ) = 0.
  throttle' + Trim * ( VT_req - VT ) = 0.

  -- Outputs Asignation
  s_elevator.signal = elevator
  s_aileron.signal = aileron
  s_rudder.signal = rudder
  s_throttle.signal = throttle
END COMPONENT
```

#### 7.5 COMPONENTE “Actuator”

Es el componente implementado para simular el comportamiento de las superficies de control y que contiene, por tanto, todas las ecuaciones necesarias para modelar el retardo de primer orden y establecer los límites de deflexión y velocidad de deflexión. Está definido en el fichero FLIGHT\_actuators.el.

Es un componente definido con un puerto de entrada y otro de salida, ambos tipo *analog\_signal*. Tales que por el puerto de entrada se le pasan los valores de las deflexiones exigidas y por el de salida devuelve el valor real de la deflexión, incluidas las limitaciones del actuador.

De nuevo, los datos son una de las características reseñables del componente ya que a través de ellos se establece la inversa del tiempo característico de respuesta del actuador (*Rtau*), el límite de deflexión (*DL*) y el límite de velocidad de deflexión (*RL*).

## 8 CONSTRUCCIÓN DEL COMPONENTE “F16” DE LA LIBRERÍA FLIGHT\_SIM

Ya se está en disposición de conectar todos los componentes definidos hasta el momento (*Frame*, *F16Aerodynamics*, *F16Engine*, *Trimmer* y *Actuator*) y que darán lugar al modelo final de F-16 y que está definido en el fichero FLIGHT\_F16.el.

Es un componente definido con cuatro puertos de entrada tipo *analog\_signal* y un puerto de salida tipo *State*. Tales que por los puertos de entrada se le pasan los valores de las variables de control y por el de salida devuelve el estado de la aeronave.

Además se debe indicar en el código todos aquellos componentes de ésta u otras librerías que van a ser utilizados.

Es en el momento de implementar este componente cuando se debe asignar los valores del modelo a los datos de los componentes genéricos de la librería con el objeto de particularizar las ecuaciones al modelo de F-16, es decir, los de los componentes *Frame*, *Trimmer* y *Actuator*. Se puede ver como se hace esto en el bloque “*TOPOLOGY*” en el código.

```
COMPONENT F16
PORTS
  IN analog_signal s_elevator
  IN analog_signal s_aileron
  IN analog_signal s_rudder
  IN analog_signal s_throttle
  OUT State state_out

TOPOLOGY
  Frame Frame(
    mass = 9071.8,
    Ixx = 28384.,
    Iyy = 166832.,
    Izz = 188610.,
    Ixy = 0.,
    Ixz = 2935.,
    Iyz = 0.)

  F16Aerodynamics Aero

  F16Engine Engine

  Actuator El_Actuator(
    Rtau = 20.2,
    RL = 60.,
    DL = 25.)

  Actuator Ail_Actuator(
    Rtau = 20.2,
    RL = 80.,
    DL = 21.5)

  Actuator Rdr_Actuator(
    Rtau = 20.2,
    RL = 120.,
    DL = 30.)

  Trimmer Trimmer(
    VT_req = 100.,
    ClimbAngle_req = 0.,
    dps_i_req = 0.)
```

```
S_summing Sum1
S_summing Sum2
S_summing Sum3
S_summing Sum4

-- Components connections
CONNECT s_elevator TO Sum1.s_in_1
CONNECT s_aileron TO Sum2.s_in_1
CONNECT s_rudder TO Sum3.s_in_1
CONNECT s_throttle TO Sum4.s_in_1

CONNECT Trimmer.s_elevator TO Sum1.s_in_2
CONNECT Trimmer.s_aileron TO Sum2.s_in_2
CONNECT Trimmer.s_rudder TO Sum3.s_in_2
CONNECT Trimmer.s_throttle TO Sum4.s_in_2

CONNECT Sum1.s_out TO El_Actuator.s_in
CONNECT Sum2.s_out TO Ail_Actuator.s_in
CONNECT Sum3.s_out TO Rdr_Actuator.s_in
CONNECT Sum4.s_out TO Engine.s_throttle

CONNECT El_Actuator.s_out TO Aero.s_elevator
CONNECT Ail_Actuator.s_out TO Aero.s_aileron
CONNECT Rdr_Actuator.s_out TO Aero.s_rudder

CONNECT Aero.s_Mach TO Engine.s_Mach

CONNECT Aero.forces_out, Engine.forces_out TO \
  Frame.forces_in

CONNECT Frame.state_out TO Trimmer.state_in, \
  Aero.state_in, Engine.state_in, state_out

END COMPONENT
```

Ha podido verse en el código con que sencillez se conectan entre si los distintos componentes de la librería para dar lugar al componente *F16* y, por tanto, que fácilmente podría modelarse cualquier otra aeronave con una altísima reutilización de código.

Estas interconexiones se muestran esquemáticamente en la figura 4 donde los cuatro puertos de entrada están situados a la izquierda de la figura y el de salida a la derecha. Se observa que las señales de control de la entrada se suman a las obtenidas en el componente *Trimmer*, es decir, a las del vuelo estacionario. A continuación los tres parámetros de control *de*, *da* y *dr*, tras pasar por cada uno de los componentes *Actuator*, entran al componente *F16Aerodynamics* para calcular las fuerzas aerodinámicas mientras que el parámetro de control *p* entra directamente al componente *F16Engine*, pues este componente ya incluye el modelo de actuador, para calcular las fuerzas propulsivas. Así las salidas de ambos componentes proporcionan los valores de las fuerzas al componente *Frame* que integra el estado del sistema. Como varios componentes necesitan conocer el estado de la aeronave en cada instante las salidas del componente *Frame* van al resto, al igual que sucede para el componente *F16Engine* que requiere conocer el Mach de vuelo en cada instante, el cual se lo proporciona el componente *F16Aerodynamics*.

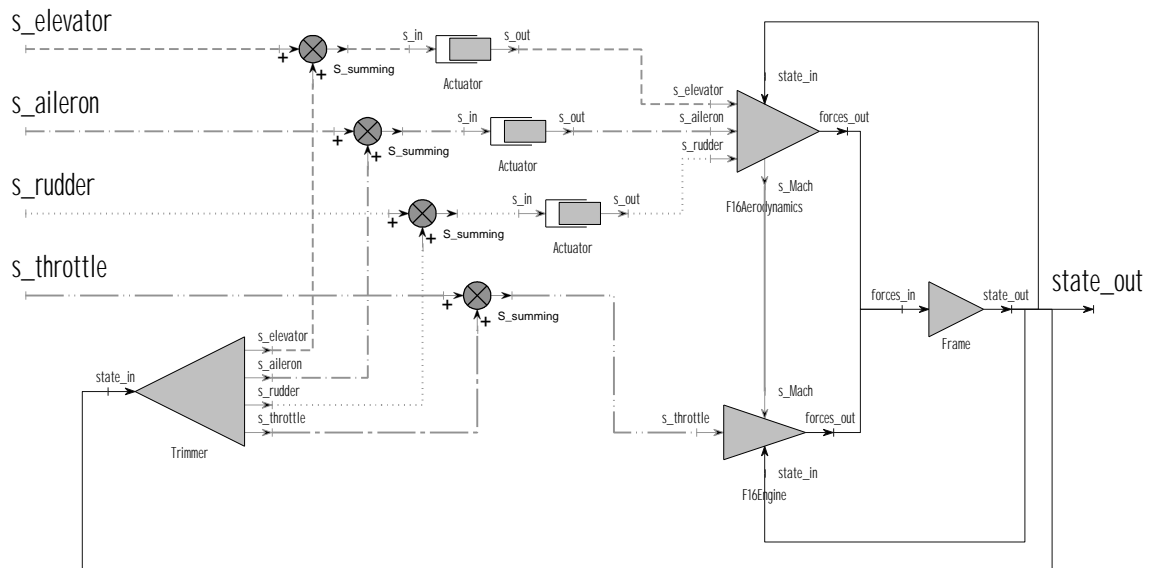


Figura 4: Conexión de los componentes para la construcción del modelo de F-16

## 9 MODELADO DE LOS AUTOPILOTOS

El modo de implementar un modelo de autopiloto en EcosimPro sigue unos pasos muy semejantes a los seguidos a la hora de construir el componente *F16*. Es por ésta razón que se necesita conocer un esquema con los componentes necesarios y como están interconectados.

Hay que recordar antes que los autopilotos, como todo sistema de control, son específicos de cada sistema y deberán ser implementados para cada aeronave en particular. Si bien los que se detallarán a continuación sirven de ejemplo para posteriores desarrollos de modelos para otras aeronaves.

Serán necesarios algunos componentes comunes a cualquier sistema de control y que se encuentran definidos en la librería de control (multiplicadores, integradores, sumadores y restadores, ...) y otros propios de la librería *FLIGHT\_SIM*. Algunos de éstos componentes responden a diferentes modelos de actuadores como es el caso del componente *ARI* (Aileron-Rudder Interconnect) que está definido en el fichero *FLIGHT\_actuators.el* y otros a diferentes modelos de sensores que nos permiten obtener algunas variables de estado de la aeronave como son el sensor de ángulo de ataque (*AOASensor*), el sensor de ángulo de deslizamiento (*SDLSensor*), los giróscopos (*GyroSensor*) y los acelerómetros (*AccelSensor*) y que están definidos en el fichero *FLIGHT\_sensors.el*. Será sobre estas variables sobre las que se ejercerá el control.

A continuación se detallarán, siguiendo el esquema establecido, dos autopilotos característicos. Un primer autopiloto longitudinal que comandará la velocidad angular de cabeceo ( $q$ ) y que se

denominará *F16qAutopilot*. Y un segundo autopiloto lateral-direccional que comandará la velocidad angular de balance ( $p$ ) manteniendo la condición de velocidad angular de guiñada ( $r$ ) nula y que se denominará *F16prAutopilot*. Ambos componentes están definidos en el fichero *FLIGHT\_autopilots.el*.

### 9.1 COMPONENTE “F16qAutopilot”

Es un componente definido con dos puertos de entrada, uno tipo *State* y otro tipo *analog\_signal*, y otro de salida tipo *analog\_signal*. Tales que por los puertos de entrada se le pasan los valores de estado de la aeronave y la velocidad angular de cabeceo comandada y por el de salida devuelve el ángulo de deflexión comandado del elevador.

Se puede ver en la figura 5 el esquema detallado del sistema de control donde los dos puertos de entrada están situados a la izquierda y el de salida a la derecha.

Se observa como se pasan las variables de estado a los distintos sensores para obtener las señales de realimentación del autopiloto y las comandadas al sistema de control.

Aparecen en este modelo algunos componentes que pertenecen a la librería *CONTROL* y otros a la librería *FLIGHT\_SIM*. Los de la librería de control son componentes típicos como filtros, integradores, multiplicadores, sumadores, ... Y los de la librería *FLIGHT\_SIM* son en este caso un sensor que mide el ángulo de ataque de la aeronave (*AOASensor*) y un giróscopo que mide la velocidad angular de cabeceo (*GyroSensor*).



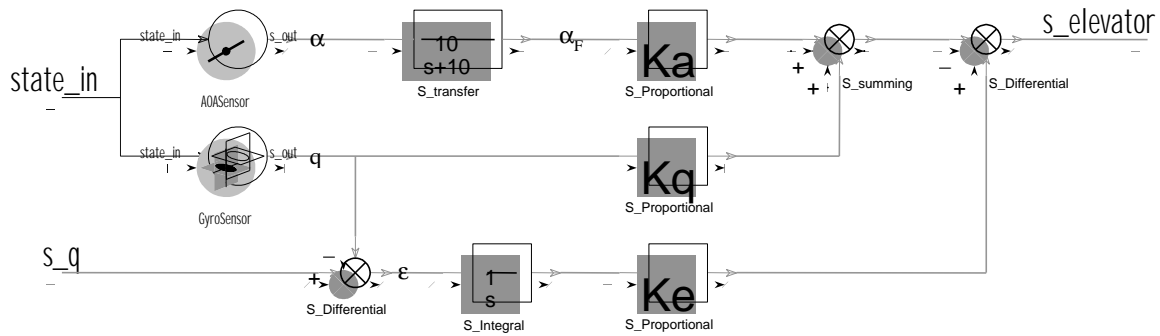


Figura 5: Conexión de los componentes para la construcción del modelo de autopiloto longitudinal

## 9.2 COMPONENTE “F16prAutopilot”

Es un componente definido con tres puertos de entrada, uno tipo *State* y dos tipo *analog\_signal*, y otros dos de salida tipo *analog\_signal*. Tales que por los puertos de entrada se le pasan los valores de estado de la aeronave y las velocidades angulares de balance y guiñada comandadas y por los de salida devuelve los ángulos de deflexión comandados de los alerones y la deriva.

Se puede ver en la figura 6 el esquema detallado del sistema de control donde los tres puertos de entrada están situados a la izquierda y los dos de salida a la derecha.

Se observa nuevamente como se pasan las variables de estado a los distintos sensores para obtener las

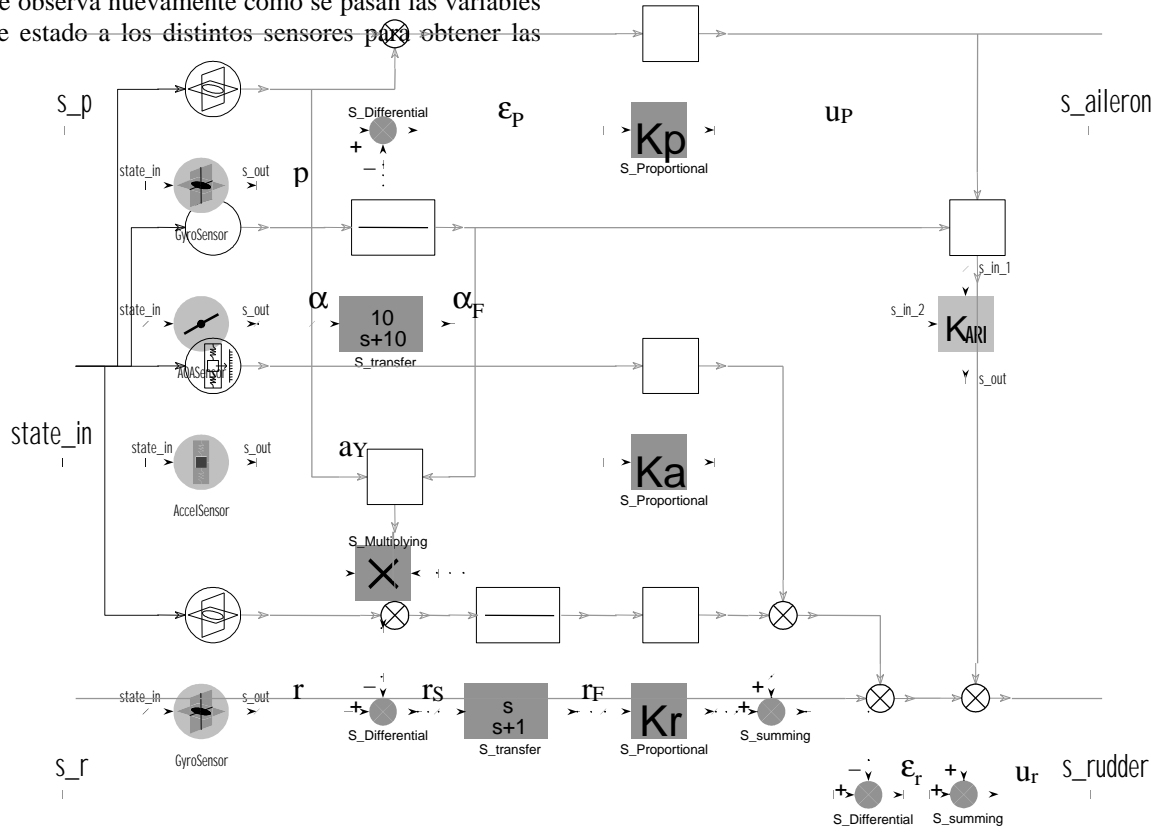


Figura 6: Conexión de los componentes para la construcción del modelo de autopiloto lateral-direccional

señales de realimentación del autopiloto y las comandadas al sistema de control.

En este modelo se destacará únicamente aquellos componentes diferentes. Los pertenecientes a la librería de control son del mismo tipo y no requieren mayor atención. Por el contrario aparecen algunos componentes nuevos de la librería FLIGHT\_SIM como son un acelerómetro que mide la aceleración lateral de la aeronave (AccelSensor) y un actuador denominado ARI (Aileron-Rudder Interconnection) que suele corresponder con un sistema típicamente hidráulico o eléctrico ( $K_{ARI}$ ). Además de sendos giróscopos que en este caso miden las velocidades angulares de cabeceo y de guiñada de la aeronave (GyroSensor).

## 10 CONSTRUCCIÓN DEL COMPONENTE "F16\_Autopiloted" DE LA LIBRERÍA FLIGHT\_SIM

Ya se está en disposición de conectar ambos autopilotos al modelo de F-16 (componente *F16*) y que darán lugar al modelo final de F-16 con control longitudinal y lateral-direccional, que está definido en el fichero FLIGHT\_F16\_Autopiloted.el.

Es un componente definido con un único puerto de entrada tipo *analog\_signal*, ya que sólo se deja un parámetro de control de la aeronave libre, el del motor. Tal que por este puerto de entrada se le pasan los valores del mando de potencia del motor.

A continuación se muestra en la figura 7 como se conectan los distintos componentes.

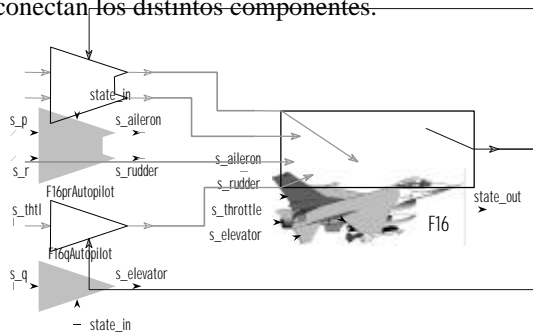


Figura 7: Conexión de los componentes para la construcción del modelo de F-16 con control longitudinal y lateral-direccional

## 11 SIMULACIÓN CON ECOSIMPRO DE LOS MODELOS DE LA LIBRERÍA FLIGHT\_SIM

Todo este desarrollo ha llevado a la consecución de dos modelos (dos componentes) que permitirán realizar una serie de experiencias de interés y mostrar así las capacidades de simulación y las posibles aplicaciones en el ámbito de la mecánica del vuelo que ofrece la herramienta EcosimPro. Por una parte se podrá simular el comportamiento de la aeronave con el componente *F16* y por otra el comportamiento de la misma al control de ambos autopilotos (longitudinal y lateral-direccional) con el componente *F16\_Autopiloted*.

Teniendo en cuenta el modelo de F-16 elegido y las capacidades implementadas esta librería permite simular tanto maniobras estacionarias como no estacionarias. Se verán a continuación algunos ejemplos. En concreto se obtendrán las curvas de actuaciones mas características de la aeronave (para diferentes condiciones y regímenes de vuelo) y curvas de respuesta de la aeronave a los parámetros de control (elevador, alerones, deriva y potencia), así como distintas maniobras gobernadas por ambos autopilotos.

### 11.1 CURVAS DE ACTUACIONES DEL F-16

Se pretende obtener el valor de una serie de variables de estado o de control para distintas condiciones de vuelo simétrico y estacionario para distintos valores de los parámetros libres, por ejemplo, curva de potencia o de deflexión de elevador requeridos para condición de vuelo horizontal, simétrico y estacionario en un plano vertical en función de la velocidad de vuelo, o para condición de viraje coordinado en un plano horizontal en función de la velocidad de viraje para una velocidad de vuelo dada.

Todos estos resultados se obtienen calculando los estacionarios del componente *F16* para distintos valores de los datos del *Trimmer*, tales que impongan las condiciones de vuelo estacionario deseadas. Se puede ver en el primer ejemplo con que facilidad se hace esto en el caso de un experimento implementado con EcosimPro

--- Steady state conditions:

- VT = [40 m/s, 240 m/s]
- ClimbAngle = 0 rad
- psi' = 0 rad/s

```
Trim = 1
Trimmer.ClimbAngle_req = 0.
Trimmer.dpsi_req = 0.
FOR(Trimmer.VT_req = 40.; Trimmer.VT_req < 240.; \
    Trimmer.VT_req = Trimmer.VT_req + 5.)
    STEADY()
END FOR
```

En primer lugar se fija el valor de la variable *Trim* para calcular condiciones de vuelo estacionario, luego se fijan las condiciones de vuelo horizontal en un plano vertical y por último se obtiene los resultados para distintas velocidades de vuelo comprendidas entre 40 y 240 m/s.

Así se obtienen las siguientes curvas de actuaciones del F-16, véase figuras 8 y 9, para vuelo horizontal, simétrico y estacionario en un plano vertical

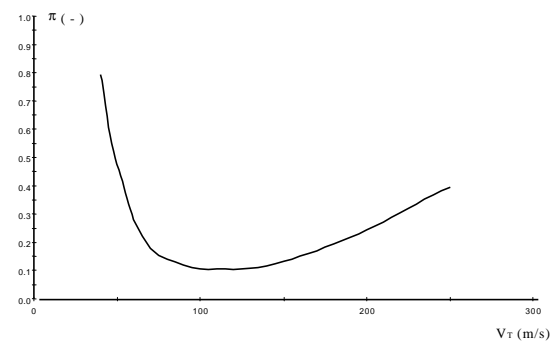


Figura 8: Curva de potencia para vuelo horizontal, simétrico y estacionario en un plano vertical

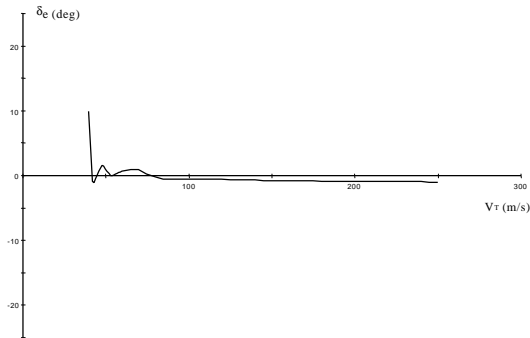


Figura 9: Curva de deflexión del elevador para vuelo horizontal, simétrico y estacionario en un plano vertical

Haciendo de forma análoga para las condiciones de viraje coordinado en un plano horizontal se obtiene, véase figuras 10 y 11, las curvas de carga y de ángulo de balance.

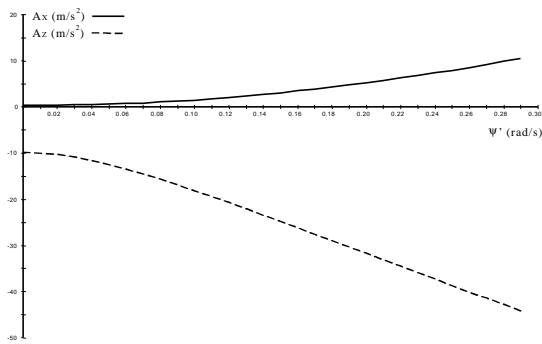


Figura 10: Curva de carga para viraje coordinado en un plano vertical

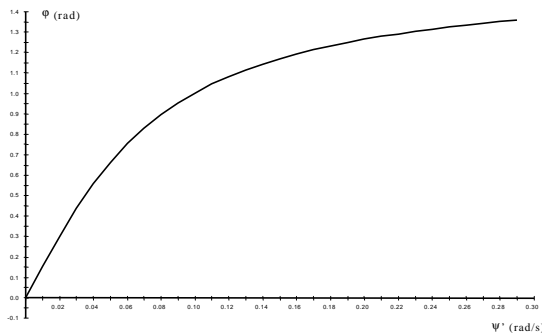


Figura 11: Curva de ángulo de balance para viraje coordinado en un plano vertical

### 11.2 CURVAS DE RESPUESTA DEL F-16

Se pretende obtener con estos experimentos la respuesta del F-16 a diferentes perturbaciones en los parámetros de control de la aeronave, tiempos característicos de respuesta, picos máximos de respuesta, ...

Para ello se debe someter al componente *F16*, a partir de una cierta condición de vuelo estacionaria, a estas perturbaciones de los parámetros de control (condiciones de contorno) y propagar el modelo.

La obtención de unas condiciones estacionarias iniciales se hace como se ha visto en los experimentos anteriores, así que se mostrará únicamente como se implementan las condiciones de contorno para un ejemplo de perturbación en forma de *doublet* de la superficie de control del timón de profundidad (elevador).

```

BOUNDS -- set expressions for boundary variables
s_elevator.signal = 2.*step(TIME, 1.0, newInt()) - \
4.*step(TIME, 1.5, newInt()) + 2.*step(TIME, 2.0, newInt())
s_aileron.signal = 0
s_rudder.signal = 0
s_throttle.signal = 0
    
```

Se puede ver en las figuras 12 y 13 la deflexión real de la superficie (incluido el comportamiento del actuador) y la respuesta de los ángulos de cabeceo ( $\theta$ ) y ataque ( $\alpha$ ).

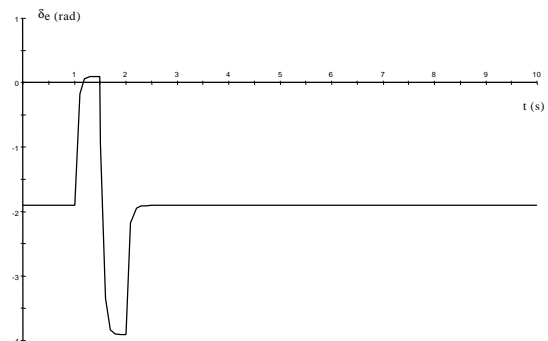


Figura 12: Deflexión de la superficie de control (elevador)

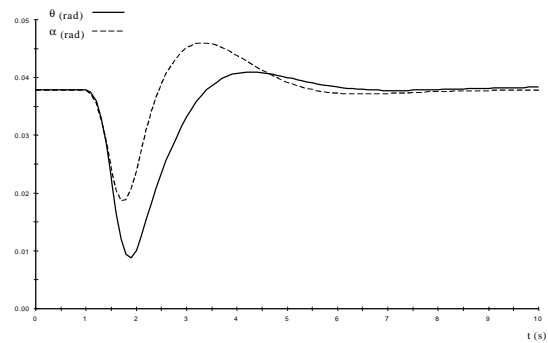


Figura 13: Curva de respuesta de los ángulos de cabeceo y de ataque

De igual forma se puede hacer para las restantes variables de estado y para los restantes parámetros de control de la aeronave. Se obtienen entonces curvas de respuesta equivalentes, sin necesidad de realizar un esfuerzo suplementario gracias a la reutilización de código.

### 11.3 MANIOBRAS CONTROLADAS POR LOS AUTOPILOTOS DEL F-16

Se mostrará en último lugar dos ejemplos diferentes de maniobras con autopiloto. La primera involucra sólo control longitudinal y por tanto tiene lugar en

un plano vertical. La segunda involucra a ambos, y la trayectoria final resulta tridimensional.

En la figura 14 se muestra una gráfica con la trayectoria de la primera maniobra. El eje vertical representa la altura de vuelo y el eje transversal el desplazamiento horizontal de la aeronave.

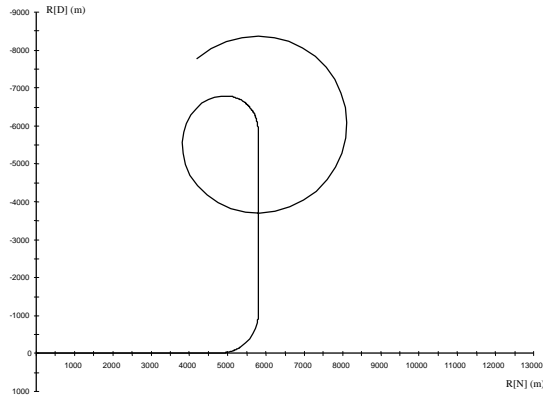


Figura 14: Trayectoria de la maniobra realizada mediante control longitudinal de la aeronave

Para la obtención de esta maniobra se exige al autopiloto la siguiente secuencia de control: 5s de vuelo horizontal estacionario ( $q_c = 0$ ), 10s a  $8.7^\circ/s$  de velocidad de cabeceo ( $q_c = 8.7$ ), 30s de vuelo vertical estacionario ( $q_c = 0$ ) y 50s a  $10^\circ/s$  de velocidad de cabeceo ( $q_c = 10$ ). Además se establece que en el instante en el que comienza la maniobra de cabeceo se coloca el parámetro de control del motor al 100%.

Esto se implementa con la misma sencillez que en los experimentos vistos hasta el momento

```
BOUNDS -- set expressions for boundary variables
prAutopilot.s_p.signal = 0.
qAutopilot.s_q.signal = 8.7*( step(TIME, 30., newInt()) - \
step(TIME, 40., newInt()) ) + 10.*step(TIME, 70., newInt())
prAutopilot.s_r.signal = 0.
s_thtl.signal = ( 1. - F16.Trimmer.throttle ) * \
step (TIME, 30., newInt())
```

Algunos resultados interesantes que se desprenden de este experimento, y que permiten ver algunas capacidades más de la librería, se observan en las figuras 15 y 16. En la primera se muestra la respuesta de la aeronave al control ejercido por el autopiloto, y como éste obtiene las velocidades de cabeceo comandadas. En la segunda se muestran los valores que toman los parámetros de control para obtener la maniobra exigida (en particular la deflexión del elevador y el parámetro de control del motor).

Igual que se ha hecho para estas magnitudes es inmediato representar cualquier otra variable de estado.

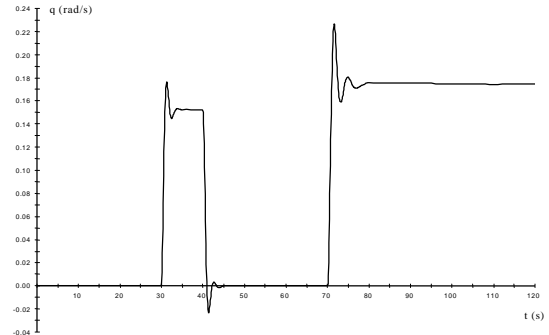


Figura 15: Velocidad angular de cabeceo obtenida por el autopiloto

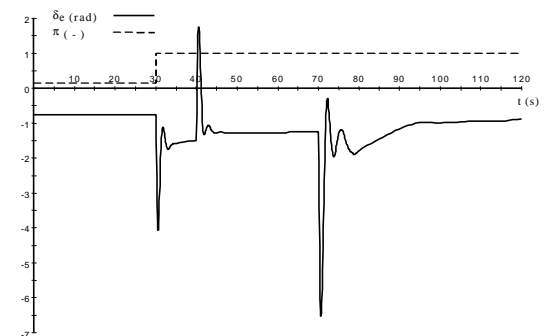


Figura 16: Deflexión del elevador y parámetro de control del motor necesarios para la maniobra

En la figura 17 se muestra la trayectoria resultante de la segunda maniobra. El eje vertical representa la altura de vuelo y el desplazamiento lateral y en el eje transversal el desplazamiento horizontal de la aeronave.

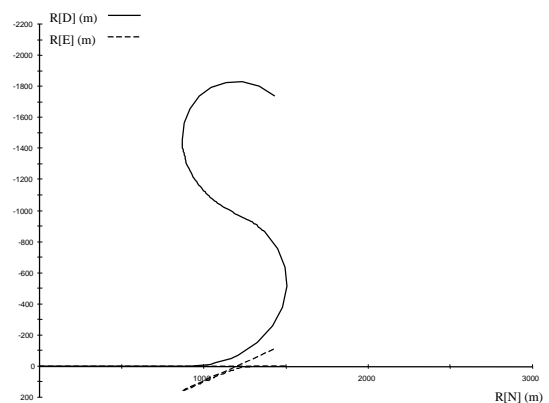


Figura 17: Trayectoria de la maniobra realizada mediante control lateral-direccional de la aeronave

Para la obtención de esta maniobra se exige al autopiloto la siguiente secuencia de control: 5s de vuelo horizontal estacionario ( $q_c = 0$ ,  $p_c = 0$ ) y 30s a  $15^\circ/s$  de velocidad de cabeceo ( $q_c = 15$ ). Además se exige que cuando la aeronave esté casi en el punto más alto del *looping* realice medio tonel durante 2s a  $150^\circ/s$  de velocidad de balance ( $p_c = 150$ ). Por último, y como se hizo en el caso anterior, se establece que en el instante en el que comienza la

maniobra de cabeceo se coloca el parámetro de control del motor al 100%.

En este caso se usará el control del autopiloto lateral sobre la velocidad angular de guiñada para evitar el resbalamiento durante la maniobra ( $r_c = 0$ ).

Todo esto se implementa con la misma sencillez que en los experimentos vistos hasta el momento

```
BOUNDS -- set expressions for boundary variables
prAutopilot.s_p.signal = 150.*( step(TIME, 15., newInt()) - \
step(TIME, 17., newInt()) )
qAutopilot.s_q.signal = DtoR * 15. * step(TIME, 5., newInt())
prAutopilot.s_r.signal = 0.
s_thrl.signal = ( 1. - F16.Trimmer.throttle ) * \
step (TIME, 5., newInt())
```

Algunos resultados interesantes que se desprenden de este experimento se observan en las figuras 18 y 19. En la primera se muestra la respuesta de la aeronave al control ejercido por los autopilotos, y como tratan de obtener las velocidades angulares comandadas. En la segunda se muestran los valores que toman los parámetros de control para obtener la maniobra exigida.

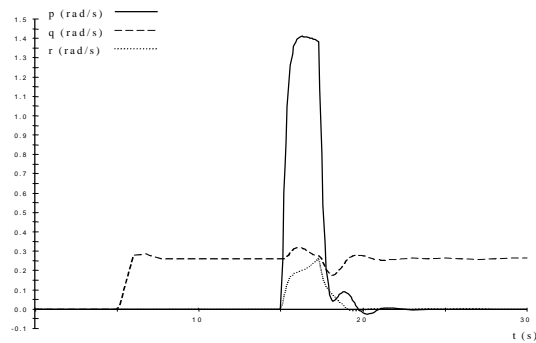


Figura 18: Velocidades angulares obtenidas por el autopiloto

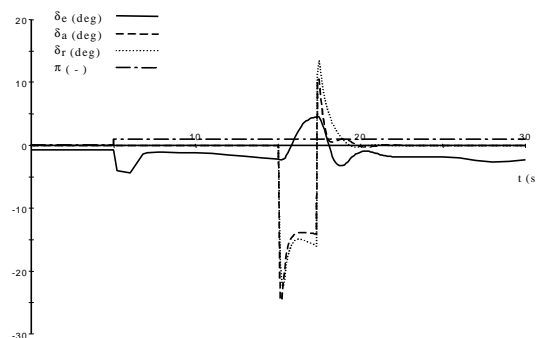


Figura 19: Parámetro de control de la aeronave necesarios para la maniobra

De esta manera se ha hecho un recorrido por todos los componentes desarrollados para modelos de mecánica del vuelo, que han sido implementados en la librería FLIGHT\_SIM, y de las posibilidades de aplicación que ofrece EcosimPro en este campo. Se enumerarán a continuación las capacidades que se

podrían añadir, muchas de las cuales están siendo incluidas ya.

### Nuevas Capacidades para la Librería

- Mejorar el componente *Frame* de manera que la integración de la actitud de la aeronave se realice utilizando indistintamente cuaterniones o ángulos de Euler.
- Desarrollo de un modelo que incluya los efectos de atmósfera en movimiento (vientos, ráfagas, turbulencia, ...)
- Mejoras de detalle en los modelos de los actuadores y de los sensores.
- Desarrollo de otros modelos de aeronave, como helicópteros y misiles.
- Desarrollo de componentes para representar los autopilotos más típicos.

### Conclusiones

La librería FLIGHT\_SIM permite construir de forma rápida y sencilla modelos de aeronave, los cuales pueden ser utilizados para calcular las actuaciones estacionarias o la respuesta transitoria de la misma. Esta capacidad que ofrece EcosimPro permite el uso de un modelo único, reduciendo el esfuerzo de desarrollo de modelos y la labor de mantenimiento de los mismos.

Los autopilotos se simulan de forma similar a como se representa para los modelos de aeronave utilizando conjuntamente los componentes de la librería CONTROL (preexistente de EcosimPro) y la librería FLIGHT\_SIM.

### Agradecimientos

A Pedro Cobas por todos sus consejos.

### Referencias

[1] Stevens, B. L. & Lewis, F. L., (1992) "Aircraft control and simulation", John Wiley & Sons, Inc, USA.

[2] Martínez García, J. J. y Gómez Tierno, M. A., (2000) "Apuntes de mecánica del vuelo", Publicaciones de la E. T. S. I. Aeronáuticos, España.

