

## MOTOR DE SIMULACION PARA SIMULADORES DE VUELO

Javier E. Luiso<sup>a</sup>, María V. Galán<sup>a</sup>, Pedro S. Giraudo<sup>b</sup>, Eduardo Zapico<sup>b</sup>, Horacio A. Abbate<sup>a</sup>

<sup>a</sup>Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEFA), Argentina

<sup>b</sup>Departamento de Aeronáutica, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, Argentina

### RESUMEN:

*Presentamos un motor de simulación destinado a ejecutar los diferentes algoritmos que modelan el comportamiento físico del vuelo en un simulador de vuelo.*

*El motor de simulación fue concebido y diseñado con el objetivo de proveer una herramienta de simulación de propósito general especializada en algoritmos numéricos, la que permite modelar al sistema a simular como una estructura conformada por un conjunto de módulos o componentes independientes, cada uno con su propio modelo computacional.*

*Cada componente tiene un algoritmo de simulación correspondiente. El motor ejecuta ciclos periódicos de simulación y en cada ciclo determina el aporte de cada componente a la simulación, y luego integra los resultados obtenidos para calcular el efecto total sobre el conjunto. De esta forma, la evolución de la simulación se desarrolla en forma discreta en el tiempo.*

*Se optó por una solución que permite que las características propias de la aeronave (sistema a simular) no se refleje en el código del motor sino en el conjunto de datos de configuración que la describen. Esto además garantiza la versatilidad de la solución propuesta y la posibilidad de simular sistemas de naturalezas muy variadas.*

*Se presenta como ejemplo de aplicación del motor de simulación un modelo dinámico de vuelo 6 DOF (grados de libertad) del avión de combate F16.*

**Palabras clave:** motor de simulación, simulador de vuelo, modelo dinámico de vuelo.

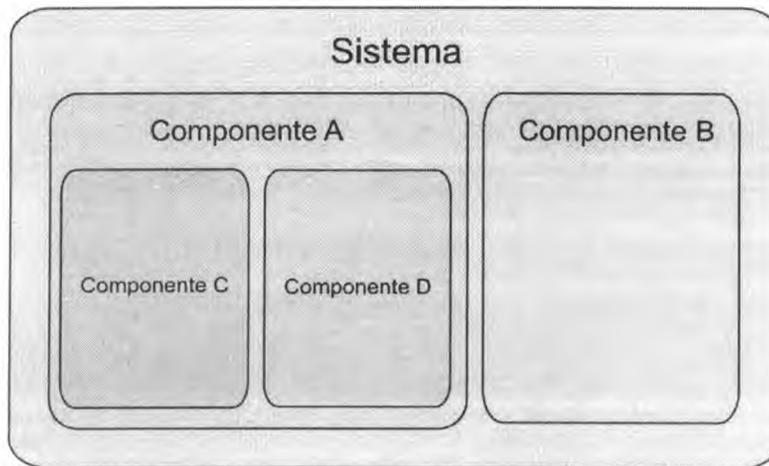
### INTRODUCCIÓN

El Motor de Simulación aquí presentado es un componente de software concebido y diseñado con el objetivo de proveer una herramienta de simulación de propósito general capaz de ejecutar algoritmos numéricos. Dichos algoritmos modelan el comportamiento físico de un determinado sistema de interés. Al ejecutar estos algoritmos el motor de simulación calcula el comportamiento físico de los mecanismos que forman parte del sistema en estudio.

El principal antecedente de nuestro sistema es la plataforma JSBSim desarrollada por Berndt [1], esta plataforma consiste en un motor de simulación, abierto, configurable y escalable dedicado a modelos dinámicos de vuelo; y el proyecto LaSRS de la NASA [2].

Nuestro objetivo es el diseño de una plataforma también configurable, escalable, pero de propósitos generales.

Este trabajo se realiza en el marco del proyecto "Simulador de Vuelo" financiado con el subsidio PAE2004-22614 ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica).



**Fig. 1** Sistema y sus componentes internos

## MOTOR DE SIMULACIÓN

Sin importar las características del sistema a simular (ej.: avión volando, un vehículo terrestre desplazándose, barco navegando), éste se modela como una entidad que posee una entrada (parámetros externos que influyen sobre el sistema), una salida (parámetros afectados por el comportamiento del sistema) y una función de transferencia asociada. La función de transferencia del sistema define cómo se calcula en un ciclo de simulación el nuevo valor de la salida y el nuevo estado del sistema en función del estado corriente del sistema y del valor corriente de la entrada. Dependiendo de la complejidad del sistema, este a su vez podrá estar conformado por módulos independientes interconectados entre sí, llamados componentes (Fig. 1). Para cada uno de estos componentes, que también son representados como entidades que poseen entradas, salidas y funciones de transferencia, existirá un modelo que simule su comportamiento.

Para poder describir al sistema modelado, sus entradas, salidas, componentes internos y sus interconexiones, se diseñó y especificó un formato de archivo basado en XML [3][4]. El Archivo de Especificación del Modelo es el que contiene esta descripción. Nuestro Motor de Simulación tiene la capacidad de interpretar este formato de archivo y de este modo es posible cargar el sistema para llevar adelante la simulación.

Durante la ejecución el Motor no sólo resuelve la simulación de cada uno de los modelos sino que, como conoce la forma en que los componentes se relacionan entre sí, determina el comportamiento del sistema completo. La evolución de la simulación se desarrolla en forma discreta en el tiempo, es decir se ejecutan ciclos periódicos de simulación y en cada uno se determina el aporte de cada componente a la simulación, y luego se integran los resultados obtenidos y obteniéndose el efecto total sobre el conjunto.

El Motor de Simulación ha sido estructurado en tres componentes principales: Biblioteca de componentes propios, el Interpretador XML y el módulo de Configuración y Control (Fig. 2).

Tanto el Archivo de Especificación del Modelo como los Archivos de Inicialización de Componentes utilizan el formato XML. El Interpretador XML es el encargado de interpretar dichos archivos. En el primero de ellos se describe el sistema a simular declarando los componentes que lo constituyen y especificando las conexiones entre entradas y salidas de los componentes [4], y en los Archivos de Inicialización de Componentes se describen los conjuntos de datos requeridos por ciertos componentes para la implementación computacional de sus respectivas funciones de transferencia.

El módulo de Configuración y Control ofrece una interfaz que permite configurar parámetros propios de la simulación: establecer el paso de discretización, definir un tiempo de simulación y llevar adelante la simulación hasta que ese tiempo se cumpla, realizar simulaciones paso a paso, etc. Además brinda acceso al Estado de Sistema en cualquier momento, permitiendo incluso modificarlo sin tener que detener la simulación. Entendiéndose como Estado del Sistema al conjunto de magnitudes continuas y discretas, que caracterizan en cada momento la situación del mismo.

La Biblioteca de Componentes Básicos es un conjunto de componentes genéricos disponibles para ser aprovechados por el usuario en el diseño de modelos. En el proceso de describir el sistema como un conjunto de componentes interconectados, el usuario puede hacer uso de esta biblioteca simplificando el trabajo. También se dispone de un mecanismo simple y poderoso que permite el diseño e incorporación de nuevos componentes en función de las necesidades.

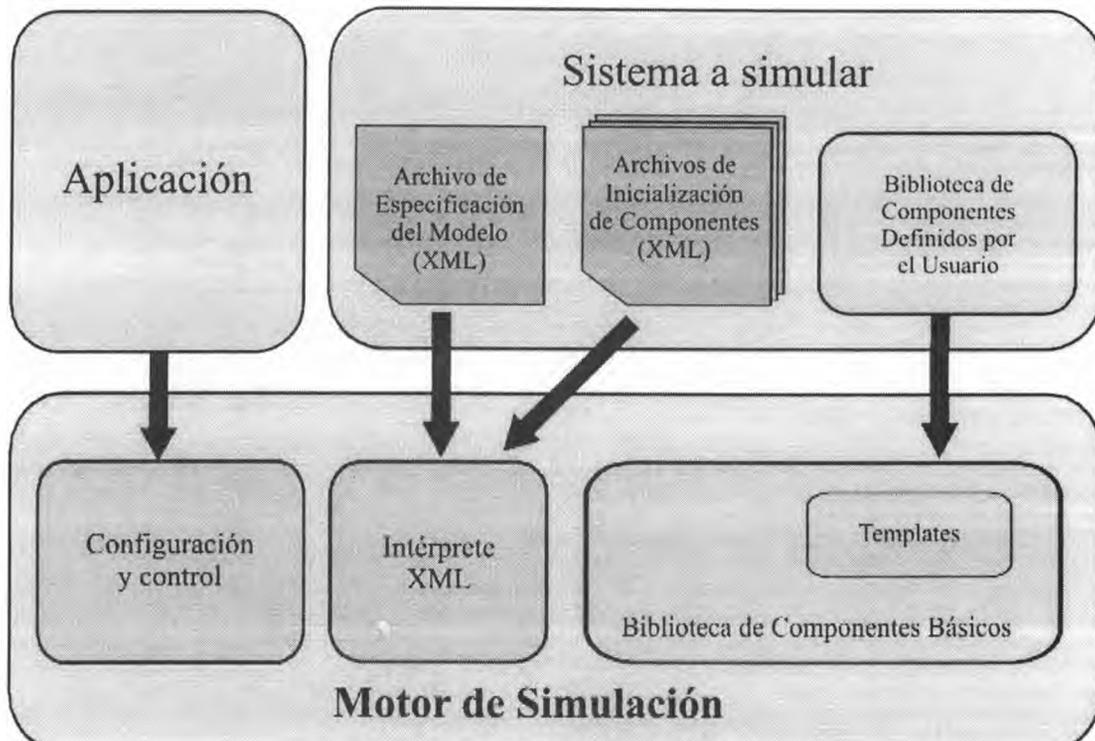


Fig. 2 Estructura del Motor de Simulación

## COMPONENTES BÁSICOS

En el proceso de modelar un sistema para ser luego simulado en nuestro motor, el usuario debe dividirlo en subsistemas interconectados de menor complejidad. Cada subsistema estará caracterizado por su propia función de transferencia.

Cada uno de estos subsistemas se implementan como componentes dentro del motor de simulación y el usuario podrá para ello emplear los componentes básicos del motor. La complejidad o especificidad de un subsistema puede implicar que no sea posible modelar su función de transferencia con los componentes básicos, o que el usuario desee emplear algún algoritmo numérico particular, en estos casos el usuario diseñará un componente propio.

La vida de un componente durante la ejecución del modelo en el motor abarca las siguientes etapas:

**Creación:** a medida que el Intérprete XML procesa el archivo de especificación las distintas instancias de los componentes presentes en el sistema son creadas.

**Interconexión:** el archivo de especificación también contiene la información que determina el modo en que los distintos componentes se conectan entre sí. Una vez que todas las instancias han sido creadas la interconexión entre todas ellas es realizada.

**Inicialización:** antes de realizar el primer ciclo de simulación, el sistema en general y cada componente en particular es inicializado.

**Ciclo de Simulación** en cada ciclo se ejecuta la función de transferencia del componente para determinar su aporte a la simulación.

## COMPONENTES DEFINIDOS POR EL USUARIO

El Motor de Simulación provee un mecanismo simple y poderoso para que el usuario pueda diseñar e implementar componentes externos en función de las necesidades que surjan al modelar un sistema particular, cuando por ejemplo una función de transferencia no pueda ser modelada con los componentes básicos del motor.

Para crear un componente el usuario utiliza la clase base o plantilla *UserComponent* provista por el motor y debe definir dos funciones callback las cuales son invocadas durante las etapas de *Inicialización* y en cada *Ciclo de Simulación* respectivamente.

### EJEMPLO DE APLICACIÓN

En el marco del proyecto fue desarrollado entre otros un modelo de 6 grados de libertad para el avión F16.

El modelo numérico de la aeronave está basado en datos obtenidos de ensayos en túnel de viento en las instalaciones NASA-Langley. Los datos representan el comportamiento de la aeronave en forma de coeficientes aerodinámicos que luego son utilizados para calcular fuerzas y momentos. Los coeficientes aerodinámicos cubren un amplio rango de ángulo de ataque ( $-15^\circ$  a  $+45^\circ$ ) y de deslizamiento ( $-30^\circ$  a  $+30^\circ$ ) y también reproducen los efectos de la deflexión de comandos (timón de profundidad, timón de dirección y alerones). Los datos son válidos hasta Mach 0,6. Los valores están organizados en tablas (en su mayoría de dos dimensiones) y se utilizan funciones de interpolación lineal para el cálculo de los coeficientes.

A partir de este modelo numérico se desarrolló el modelo de componentes apropiado para el ensayo en el motor de simulación (Fig. 3). Se definieron los componentes del sistema y se determinó la función de transferencia que los mismos debían implementar. En algunos casos, dada la complejidad de estas, los componentes fueron divididos internamente en sub-componentes. Los principales componentes del modelo son:

*Windaxis*: calcula los ángulos de ataque y deslizamiento a partir de la velocidad del viento.

*TGear* y *PDot*: determina la potencia del motor en función de la posición del control del acelerador.

*ADC*: calcula la presión barométrica a partir de la velocidad y la altura de la aeronave.

*THRUST*: determina el empuje del avión.

*Navigation*: calcula las variaciones de la posición de la aeronave en vuelo.

*Quaternions*: este módulo concentra todos los cálculos relacionados con el álgebra de cuaterniones

*Dynamics Coefficients*: calcula los coeficientes de aceleración y velocidad.

*Derivatives*: determina las derivadas de los coeficientes de aceleración y velocidad.

*Moments*: calcula los momentos de inercia.

Paralelamente se codificó el modelo numérico en MATLAB con el objetivo de realizar los mismos ensayos y poder contrastar los resultados (Fig. 5 y 6).

Uno de los componentes que forma parte del modelo desarrollado para el avión F16 es el que denominamos THRUST, el cual es responsable de calcular el empuje en función de tres variables: potencia del motor, altura y la velocidad.

El modelo matemático desarrollado que sirve de base para este componente, calcula a partir de las variables de entrada (potencia, altura y velocidad) tres coeficientes (TMIL, TIDL, TMAX) mediante el uso de tablas de interpolación, según el algoritmo 1.

if (potencia ≤ 50)

$$\text{Empuje} = \text{TIDL} + (\text{TMIL} - \text{TIDL}) \frac{\text{potencia}}{50}$$

else

$$\text{Empuje} = \text{TMIL} + (\text{TMAX} - \text{TMIL}) \frac{\text{potencia} - 50}{50}$$

#### Algoritmo 1 Transferencia del componente THRUST

Dentro de la Biblioteca de Componentes existe un componente que implementa un mecanismo de interpolación lineal sobre conjuntos de datos vectoriales y matriciales. El conjunto de datos conocidos sobre el que se realiza la interpolación se especifica en el Archivo XML de Inicialización del Componente. Dependiendo de si la interpolación se realiza sobre un conjunto de datos vectorial (Ecuación 1) o matricial (Ecuación 2).

**Ecuación 1.** Función de transferencia para la interpolación sobre un conjunto de datos vectorial.

$$Y, X \in R^n, V = (v_0, v_1, \dots, v_{N-1}) / v_0 \leq v_j < X < v_{j+1} \leq v_{N-1}$$

$$Y = \frac{X - v_j}{|v_{j+1} - v_j|} v_{j+1} + 1 - \frac{X - v_j}{|v_{j+1} - v_j|} v_j$$

**Ecuación 2.** Función de transferencia para la interpolación sobre un conjunto de datos matricial.

$$Y \in R, X = (x_0, x_1) \in R^2,$$

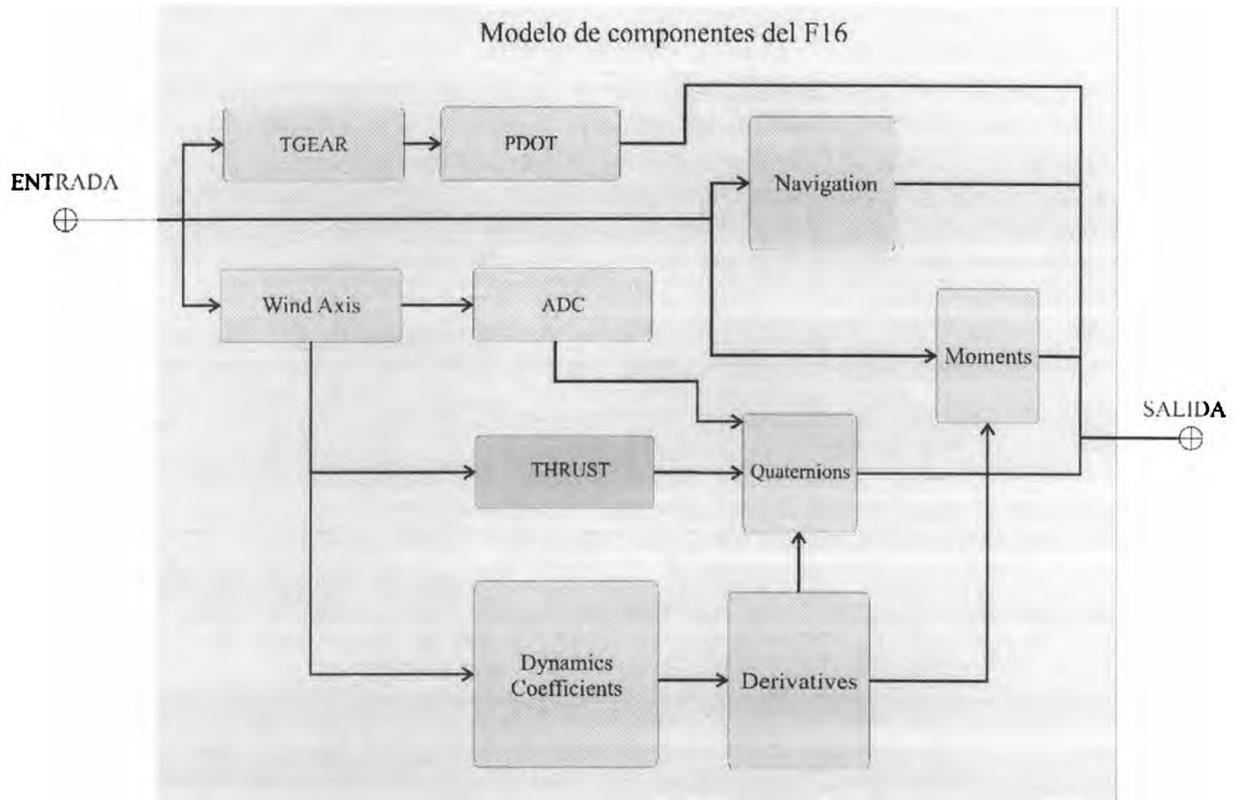
$$M \in R^{N \times M}, A = (a_0, \dots, a_{N-1}) \in R^N, B = (b_0, \dots, b_{M-1}) \in R^M,$$

$$a_0 \leq a_i < x_0 < a_{i+1} \leq a_{N-1} \quad \wedge \quad b_0 \leq b_j < x_1 < b_{j+1} \leq b_{M-1}$$

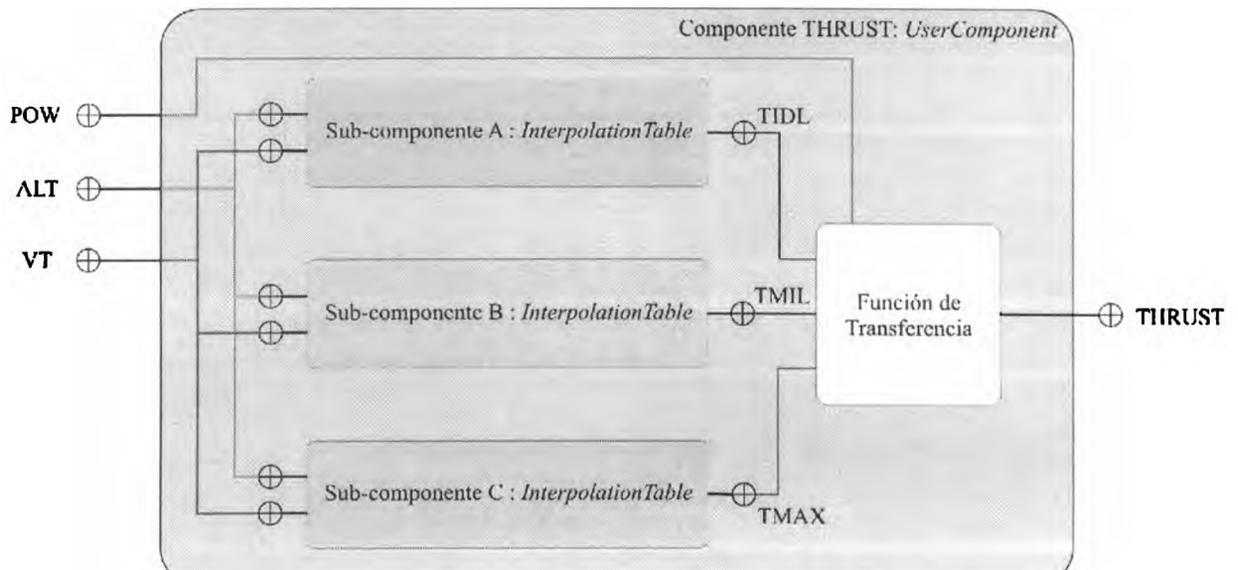
$$u = \frac{x_0 - a_i}{|a_{i+1} - a_i|}, \quad v = \frac{x_1 - b_j}{|b_{j+1} - b_j|}$$

$$Y = (1 - u)(1 - v)m_{i,j} + u(1 - v)m_{i+1,j} + v(1 - u)m_{i,j+1} + u v m_{i+1,j+1}$$

La estructura elegida para implementar el componente THRUST fue la de crear un nuevo componente con una función de transferencia descrita en Algoritmo 1. El mismo fue creado a partir de la plantilla *UserComponent* provista por el motor para estos casos. Internamente para el cálculo de los coeficientes TMIL, TIDL y TMAX se utilizó el componente *InterpolationTable* que forma parte de la Biblioteca de Componentes Básicos. La estructura resultante es la que se observa en la Fig. 4



**Fig 3 Modelo de componentes del avión F16**



**Fig. 4 Estructura interna del componente THRUST**

## ENSAYO Y COMPARACIÓN

Se realizó una simulación para determinar la respuesta del modelo a un pulso aplicado al timón de dirección. La posición del timón cambia desde  $0^\circ$  a  $-5^\circ$  un segundo después de iniciada la simulación y se mantiene en esta posición durante un segundo. Luego cambian a  $5^\circ$  sosteniéndose en esa posición por otro segundo para volver al estado inicial hasta el final de la simulación.

Los resultados obtenidos fueron contrastados con la versión del modelo implementado en MATLAB y muestran una evolución similar de acuerdo a lo esperado (Fig. 5 y 6).

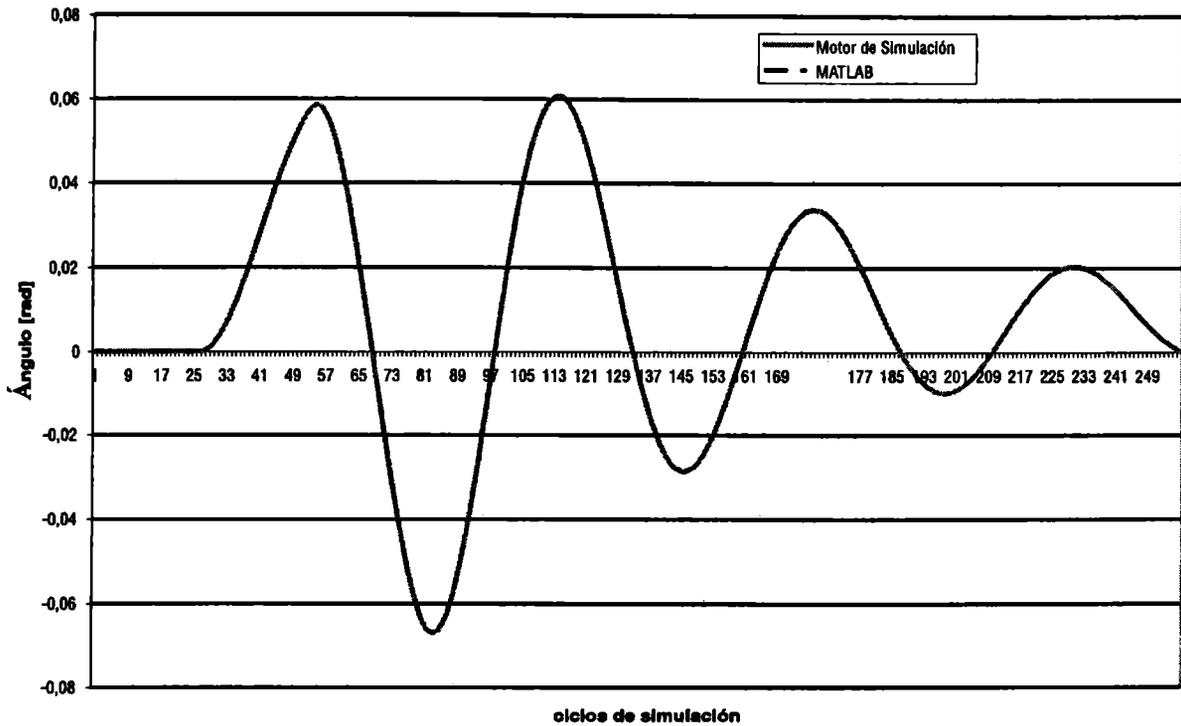


Fig 5. Ángulo alrededor del eje x del avión

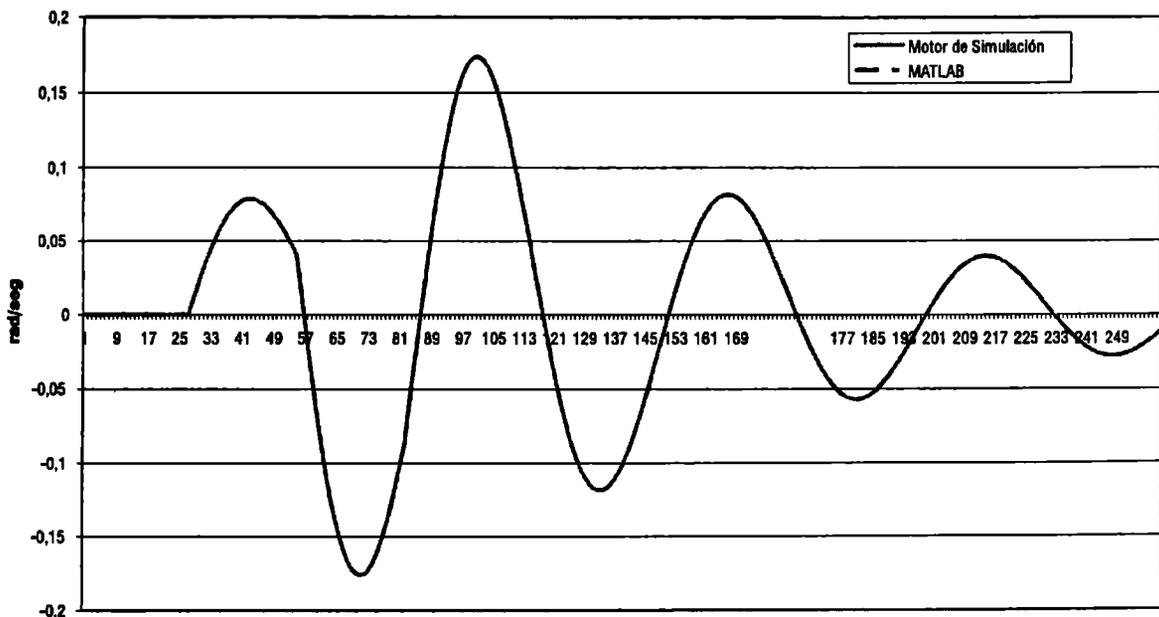


Fig 6. Velocidad de guiñada

## CONCLUSIONES Y TRABAJOS FUTUROS

Se presentó un motor de simulación de propósitos generales, que modela los sistemas a simular como un conjunto de componentes genéricos independientes. Cada componente se caracteriza con un conjunto de parámetros, de esta manera el sistema a simular no se refleja en el código del motor sino en el conjunto de datos de configuración que lo describen. Se ha implementado ya, usando este motor, un modelo de dinámica de vuelo de 6 DOF (grados de libertad) de la aeronave F16, del cual se mostró un componente como ejemplo. Se está trabajando actualmente en un modelo de vuelo de la aeronave Mirage2000 [5].

## REFERENCIAS

1. Berndt, Jon S., JSBSim: An Open Source Flight Dynamics Model in C++. AIAA Modeling and Simulation Technologies Conference and Exhibit , August 2004, Providence, Rhode Island
2. LaSRS++ AN OBJECT-ORIENTED FRAMEWORK FOR REAL-TIME SIMULATION OF AIRCRAFT Richard A. Leslie, David W. Geyer, Kevin Cunningham Patricia C. Glaab, P. Sean Kenney, Michael M. Madden AIAA-98-4529
3. E. Bruce Jackson, Bruce L. Hildreth, Flight Dynamic Model Exchange Using XML, Modeling & Simulation Technologies Conference, Paper Number AIAA-2002-4482, August, 2002.
4. Horacio Abbate - Maria Victoria GalEn – Javier E. Luiso, Motor de Simulación de Propósitos Generales, 4º Jornadas de Informática Industrial JII 2007 – Jornadas Argentinas de Informática e Investigación Operativa JAIIO 2007, Agosto 2007, Mar del Plata, Argentina
5. Eduardo N. Zapico - Pedro S. Giraudó – Horacio A. Abbate, A tensor 6-DOF Flight Dynamics Model for a Low-Cost Real-Time Flight Simulator (AIAA 2008-6852), AIAA Modeling and Simulation Technologies Conference and Exhibit, August 2008, Honolulu, Hawaii