

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**PLATAFORMA DE MOVIMIENTO DE SEIS GRADOS DE
LIBERTAD APLICABLE A ENTRENADOR DE VUELO**



TESIS
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO

PRESENTADO POR
GUILLERMO OMAR GONZÁLES CUCHO

LIMA – PERÚ

2008

Asesor

Ms. Pedro Huamaní Navarrete

*“El mundo no es sólo algo que está ahí,
exterior a mí; participo de él. Se me ofrece”*

Cyril Collard

*Al personal de la Fuerza Aérea del Perú,
por el sacrificio, entrega e ingenio que
demuestra día a día en beneficio del país.*

INDICE

Introducción	1
Capítulo 1 – Marco teórico	2
1.1 Dinámica del avión en vuelo.....	2
1.1.1 Cómo se genera el desplazamiento del avión en el espacio	3
1.2 Ecuaciones de movimiento y sistemas de referencia.....	6
1.2.1 Derivación de las ecuaciones de movimiento	7
1.2.2 Orientación del avión relativa al sistema de Coordenadas fijo a la Tierra X'Y'Z'	15
Desplazamiento en vuelo del avión relativo a la Tierra	18
1.2.4 Componentes de la fuerza de gravedad	21
1.2.5 Ecuaciones de movimiento	21
1.3 La percepción del movimiento espacial en el ser humano	22
1.4 La simulación del vuelo	24
1.4.1 Objetivos de un Simulador	25
1.4.2 Evolución histórica de la simulación.....	25
1.4.3 Tipos de sistemas	28
1.4.4 Simulación de movimiento	33
1.4.5 Desarrollos de la Fuerza Aérea del Perú	33
1.5 Aspectos metodológicos del estudio a realizar	35
1.5.1 Planteamiento del problema	35
1.5.2 Hipótesis	35

1.5.3	Objetivos de la investigación	36
1.5.4	Metodología del estudio	36
Capítulo 2 – Descripción de la plataforma Stewart		37
2.1	Manipuladores paralelos	37
2.2	Descripción de la plataforma	39
2.3	Análisis de grados de libertad	40
2.4	Cinemática inversa	41
2.5	Ecuaciones dinámicas de la Plataforma Stewart	42
Capítulo 3 – Simulación del vuelo		47
3.1	Herramientas de software empleadas	47
3.1.1	FlightGear	47
3.1.2	Simulink Aerospace Toolbox	51
3.1.3	Simulink Aerosim Toolbox	52
3.2	Diagrama de bloques en simulink para la simulación del Vuelo	52
3.2.1	Bloque pilot	52
3.2.2	Bloque environment.....	54
3.2.3	Bloque Cessna C172	57
3.2.4	Bloque generate run-script	58
3.2.5	Bloque motion conversión for animation	60
3.2.6	Bloque flightgear pre-configured 6 DOF animation.....	61

Capítulo 4 – Simulación de la plataforma Stewart con Matlab/Simulink	63
4.1 Herramientas de software empleadas	63
4.1.1 Simulink SimMechanics Toolbox	63
4.2 Simulación	63
4.2.1 Bloque leg trajectory	64
4.2.2 Bloque controller	65
4.2.3 Bloque body position sensor	67
4.2.4 Bloque plant	67
4.3 Definición del problema de control	71
4.4 Modelamiento de componentes físicos con SimMechanics	71
Capítulo 5 – Implementación de modelo a escala	73
5.1 Modelo Simulink	73
5.2 Resultados obtenidos	75
5.3 Prototipo	78
Conclusiones	81
Agradecimientos	83
Apéndice A: Características del Cessna 172	84
Apéndice B: Código Matlab para prueba de los servomotores	88
Apéndice C: Procedimiento para pruebas usando un solo computador	92

Apéndice D: Procedimiento para pruebas usando	
dos computadores.....	94
Apéndice E: Matrices de estado de la planta usando	
Matlab	96
Referencias bibliográficas.....	98

Introducción

La Fuerza Aérea del Perú cuenta con la capacidad de construir simuladores y entrenadores de vuelo para la formación de sus pilotos, adquirida a través del trabajo en este campo efectuado por el Centro de Desarrollo de Proyectos (CEDEP). Mediante técnicas de realidad virtual se logra recrear los estímulos visuales y sonoros propios del fenómeno de vuelo, con lo cual se logra acostumbrar al piloto alumno a las mismas sensaciones que percibirá en el vuelo real, lo cual permite mejorar el aprendizaje y aumentar la seguridad de vuelos.

Sin embargo, de acuerdo a las regulaciones dadas por la Organización Internacional de Aviación Civil, estos no pueden ser denominados simuladores al carecer de movimiento, lo que obliga a que las tripulaciones de aeronaves con equivalentes civiles (helicópteros y aviones de transporte) a recurrir a simuladores en el extranjero a fin de obtener la certificación respectiva y mantener la habilitación operativa.

En el CEDEP no se cuenta aún con la capacidad de generar estímulos de movimiento espacial, lo cual de acuerdo a los estándares de aviación internacional constituyen el máximo nivel de entrenamiento al que se puede llegar y con lo que se logra la mejor preparación de los pilotos. Consciente de esta carencia, la Dirección del CEDEP ha considerado necesario gestionar el desarrollo de investigaciones en el área de la simulación de movimiento en seis grados de libertad.

En la presente tesis se presenta el diseño de una plataforma que permita dotar de movimiento a los dispositivos de entrenamiento simulado desarrollados por la FAP, a fin de que estos puedan ser homologados ante la OACI y reducir así los costos anuales de entrenamiento.

Capítulo 1

Marco teórico

1.1 Dinámica del avión en vuelo

Cuatro fuerzas (sustentación, peso, empuje y resistencia) están presentes en todo momento cuando un avión está en el aire. En la figura 1 se muestra la acción de estas fuerzas en un diagrama de cuerpo libre del avión.

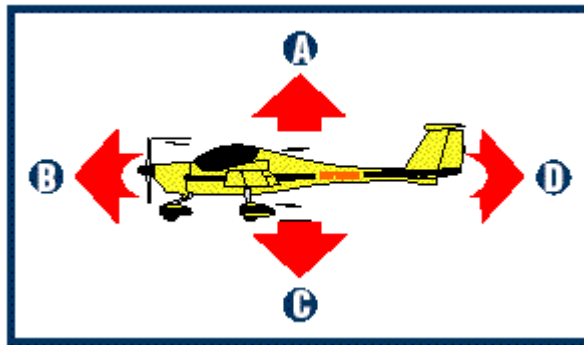


Figura 1 Las cuatro fuerzas que actúan sobre un avión en vuelo. A-Sustentación, B-Empuje, C-Peso y D-Resistencia

La sustentación es la fuerza que actúa en sentido ascendente cuando las alas de un avión se desplazan por el aire. El movimiento hacia adelante produce una ligera diferencia de presión entre la superficie superior e inferior de las alas. Esta diferencia se convierte en sustentación, y es lo que mantiene un avión en el aire venciendo a su propio peso.

El peso es la fuerza que actúa en sentido descendente. Es la única fuerza que los pilotos controlan en cierta medida, pues eligen la carga que lleva el avión. Con excepción del consumo de combustible, el peso real del avión es difícil de modificar durante el vuelo. En vuelo sin aceleración (cuando la velocidad y dirección del avión son constantes), las fuerzas opuestas de sustentación y peso están en equilibrio.

El empuje es una fuerza que actúa hacia adelante producida por una hélice impulsada por un motor. En la mayoría de los casos, cuanto más grande sea el motor (más potencia tenga), mayor será el empuje producido y más rápido volará el avión. El movimiento hacia adelante siempre genera una penalización aerodinámica, denominada resistencia. La resistencia empuja el avión hacia atrás y se trata simplemente de la resistencia molecular de la atmósfera al movimiento que la atraviesa.

El empuje hace que el avión acelere, pero la resistencia determina su velocidad final. Al aumentar la velocidad del avión, también aumenta la resistencia.

1.1.1 Cómo se genera el desplazamiento del avión en el espacio

En la figura 2 se muestran los tres ejes espaciales del avión. Mediante los controles de vuelo [1], se puede hacer que el avión gire alrededor de uno o varios de estos ejes.

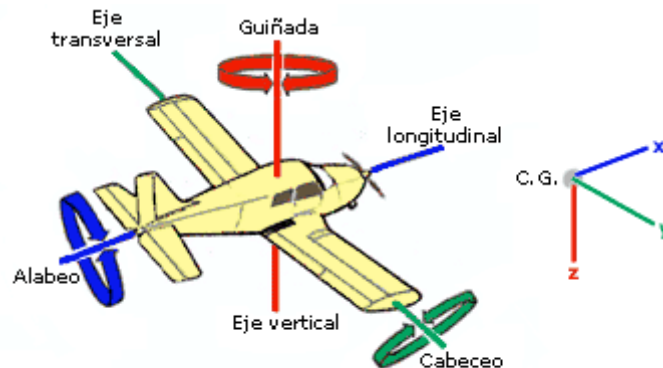


Fig.1.5.1 - Ejes del avión y movimientos sobre ellos.

Figura 2. Los tres ejes de un avión, A-Eje vertical (Guiñada o yaw), B-Eje longitudinal (Alabeo o roll), C-Eje lateral (Cabeceo o pitch)

El eje longitudinal recorre la línea central del avión desde el morro hasta la cola. Los aviones giran, o se ladean, sobre su eje longitudinal. El eje lateral discurre de un lado al otro del avión, entre los extremos de las alas. Los aviones cabecean sobre su eje lateral. El eje vertical del avión discurre de arriba abajo. Los aviones guiñan sobre su eje vertical.

Ahora se examinarán los tres controles de vuelo principales que hacen que un avión se mueva sobre sus ejes.

Alerones

Los alerones son las superficies móviles situadas en el borde exterior de salida de las alas. Su objetivo es ladear el avión en la dirección que desea virar. Cuando el control se gira a la derecha, como se muestra en la figura 3, los alerones se mueven simultáneamente en direcciones opuestas. El alerón del ala izquierda baja, lo que aumenta la sustentación del ala. El alerón del ala derecha sube, lo que reduce la sustentación del ala. Esto hace que el avión se ladee a la derecha.



Figura 3. Ladeo lateral hacia la derecha

Si el control se gira a la izquierda, como se muestra en la figura 4, el alerón del ala izquierda sube, lo que reduce la sustentación del ala izquierda. El alerón del ala derecha baja, lo que reduce la sustentación del ala derecha. Con esto, el avión se ladea (alabea) hacia la izquierda.

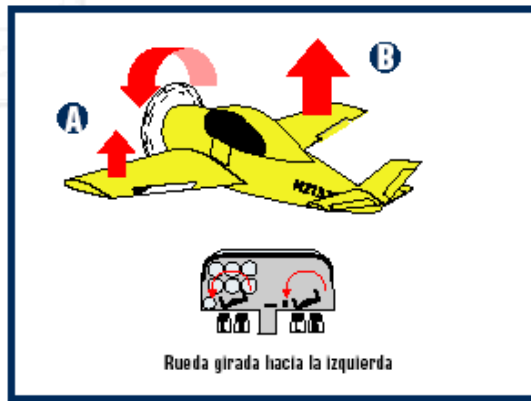


Figura 4 Ladeo hacia la izquierda

Los alerones permiten que un ala desarrolle más sustentación que la otra. La diferencia de sustentación ladea el avión, que inclina la fuerza de sustentación total en la dirección que desea virar.

Timón de profundidad

El timón de profundidad es la superficie horizontal móvil situada en la parte posterior del avión (figura 5). Su objetivo es cabecear el morro del avión arriba o abajo.



Figura 5 . Cómo el control del timón de profundidad cambia el cabeceo del avión

El control del timón de profundidad funciona con el mismo principio aerodinámico que el alerón. Al aplicar presión hacia atrás en el mando de control del avión, como se muestra en la figura 5, la superficie del timón de profundidad se

eleva. En la parte inferior de la cola se crea menor presión, lo que hace que descienda y el morro del avión cabecea arriba.

El avión de la figura 6 muestra lo que sucede cuando el mando de control se mueve hacia adelante. La superficie del timón de profundidad desciende, con lo que se crea menor presión en la parte superior de la cola.



Figura 6 . Cómo el control del timón de profundidad cambia el cabeceo del avión

1.2 Ecuaciones de movimiento y sistemas de referencia

Sistemas de coordenadas y fuerzas externas

La figura 7 muestra dos sistemas de ejes: el sistema $X'Y'Z'$ fijo a la Tierra, y el sistema XYZ fijado al cuerpo del avión. El sistema fijado a la Tierra será llamado marco de referencia inercial: uno en el cual las leyes de movimiento de Newton son válidas. Esto significa que la velocidad rotacional de la Tierra es despreciada. La experiencia indica que esto es aceptable aun para aviones supersónicos pero no para vehículos hipersónicos.

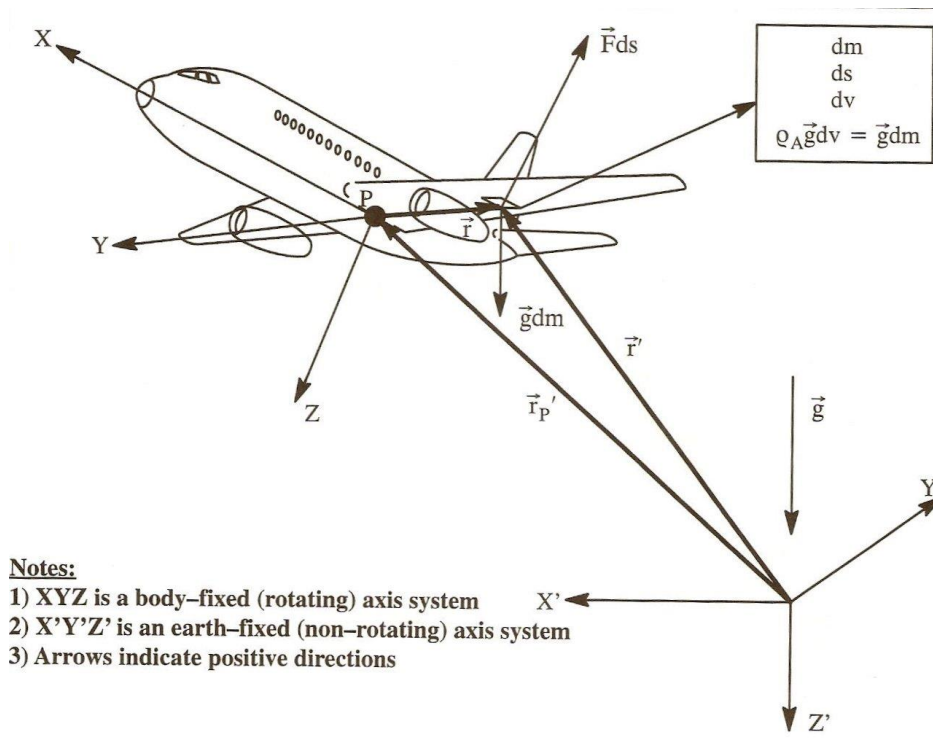


Figura 7. Sistemas de coordenadas relativos a la Tierra y al avión

Se asume que el avión de la figura consiste de una masa cuyos elementos son continuos. Estos elementos de masa son fijados a los vectores \vec{r}' que conectan el origen de $X'Y'Z'$ con cada elemento de masa. En el caso de aviones rígidos estos elementos de masa mantienen su distancia relativa al resto excepto aquellos que están en las partes rotacionales (compresores, turbinas y hélices).

Cada elemento de masa está sujeto a la aceleración de la gravedad, \vec{g} . Como se ve en la figura, se asume que el vector \vec{g} está orientado a lo largo del eje Z' positivo. Esto se conoce como asumir la **“tierra plana”**, como consecuencia de lo cual, una fuerza $\zeta_{AS} \vec{g} dv = \vec{g} dm$ actúa sobre cada elemento de masa. La magnitud ζ_A representa la densidad de masa local del avión. Estos elementos de masa localizados en la superficie del avión también están sujetos a la acción combinada de las fuerzas aerodinámicas y de empuje: F . Las fuerzas $\zeta_{AS} \vec{g} dv$ y $F ds$ se asumen como las únicas fuerzas externas actuando en el avión.

1.2.1 Derivación de las ecuaciones de movimiento

Se hace uso de la Segunda Ley de Newton, que establece que las derivadas con respecto al tiempo de los momentos angular y lineal son iguales respectivamente a las fuerzas y momentos aplicados externamente. De esto se deducen las formas integrales de las ecuaciones:

$$\frac{d}{dt} \int_V \zeta_A \frac{dr'}{dt} dv = \int_V \zeta_{AS} g dv + \int_S F ds \quad (1.1)$$

momento lineal *fuerzas aplicadas*

$$\frac{d}{dt} \int_V r' \times \zeta_A \frac{dr'}{dt} dv = \int_V r' \times \zeta_{AS} g dv + \int_S r' \times F ds \quad (1.2)$$

momento angular *momentos aplicados*

Las integrales \int_V y \int_S representan la integral de volumen y área para todo el avión. Estas integrales pueden ser evaluadas solo si se conoce la geometría externa del avión. Este es el caso si el avión es rígido. Si la aeronave es elástica se debe establecer un equilibrio aeroelástico para el cual se pueda determinar la forma externa de la aeronave.

La masa total del avión se encuentra de:

$$m = \int_V \zeta_A dv \quad (1.3)$$

En este punto se asume que el total de la masa del avión permanece constante con respecto al tiempo:

$$\frac{dm}{dt} = 0 \quad (1.4)$$

Esto se asume en tanto que el cambio de masa es suficientemente pequeño sobre un periodo de tiempo de 30-60 segundos. Este período de tiempo es el típico empleado para evaluar la respuesta del avión. Si un cambio de masa está alrededor

del 5% con respecto a la inicial después de un período de 60 segundos, se acepta el asumir la masa como constante.

Además, también se asume que la distribución de masa es constante con respecto al tiempo, de lo que se infiere que el centro de gravedad del avión permanece en el mismo lugar durante un intervalo de 60 segundos.

Hasta ahora y de acuerdo a la figura 7, los elementos de masa del avión fueron relacionados con el vector \mathbf{r}' . En adelante se emplearán los vectores \mathbf{r} y \mathbf{r}'_P , para este fin se introduce el sistema de coordenadas XYZ relativo al cuerpo. La selección y orientación del sistema XYZ relativo al avión se hace de manera absolutamente arbitraria. En la figura se asume el punto P como origen del sistema de coordenadas, ahora situado en el centro de masa del avión. Los tres vectores de posición se relacionan por:

$$\mathbf{r}' = \mathbf{r}'_P + \mathbf{r} \quad (1.5)$$

Si el punto P es el centro de masa se debe cumplir que:

$$\int_V \mathbf{r} \zeta_A dv = 0 \quad (1.6)$$

Como consecuencia la siguiente relación se mantiene para \mathbf{r}'_P :

$$\mathbf{r}'_P = \frac{1}{m} \int_V \zeta_A \mathbf{r}' dv \quad (1.7)$$

Se puede reescribir el lado izquierdo de la ecuación de momento lineal:

$$\frac{d}{dt} \frac{d}{dt} \int_V \zeta_A (\mathbf{r}'_P + \mathbf{r}) dv = \frac{d}{dt} \frac{d}{dt} m \mathbf{r}'_P = m \frac{d\mathbf{V}_P}{dt} \quad (1.8)$$

donde se define como velocidad del centro de masa del avión:

$$\mathbf{V}_P = \frac{d\mathbf{r}'_P}{dt} \quad (1.9)$$

El lado derecho de la ecuación 1.1 puede reescribirse como:

$$\int_V \zeta_A \mathbf{g} dv + \int_S \mathbf{F} ds = m\mathbf{g} + \mathbf{F}_A + \mathbf{F}_T \quad (1.10)$$

donde F_A representa el vector total de fuerza aerodinámica y F_T representa el vector de empuje total. La ecuación 1.1 puede expresarse ahora como:

$$m \frac{dV_P}{dt} = mg + F_A + F_T \quad (1.11)$$

donde se establece que la variación de momento lineal con respecto al tiempo, mV_P es igual a la suma de las fuerzas aplicadas externamente.

Luego, la ecuación de momento angular 1.2 necesita ser desarrollada, sustituyendo 1.5 en 1.2:

$$\frac{d}{dt} \int_V r \times \frac{dr'}{dt} \rho_A dv = \int_S r \times F ds = M_A + M_T \quad (1.12)$$

donde M_A representa el vector de momento aerodinámico total y M_T representa el vector de momento de empuje total.

La ecuación 1.12 establece que la variación con respecto al tiempo del momento angular $\int_V r \times \frac{dr'}{dt} \rho_A dv$ es igual a la suma de los momentos aplicados externamente al avión.

La ecuación 1.12 implica que el volumen integral es una función dependiente del tiempo, tales integrales de tiempo son inadecuadas para trabajar. Para eliminar la dependencia del tiempo, se debe hacer un cambio del sistema de coordenadas, si se coloca como ejes de referencia el sistema XYZ en vez de X'Y'Z', la integral de volumen deja de ser dependiente del tiempo. Se presenta el problema de que XYZ es un sistema de coordenadas rotante (no inercial), en cuyo caso las leyes de Newton no se aplican como fueron usadas previamente, sin embargo esto puede evitarse aplicando la siguiente relación de transformación de vectores:

$$\frac{dA}{dt} = \frac{\partial A}{\partial t} + \omega \times A = \vec{A} + \omega \times A \quad (1.13)$$

X'Y'Z' XYZ
fijo rotante

El vector \vec{A} representa a cualquier vector que va a ser transformado. El vector $\vec{\omega}$ es el vector de rotación angular del sistema XYZ relativo al sistema X'Y'Z'. Este vector también está referido a la velocidad angular del avión respecto a la Tierra.

La transformación de la fórmula 1.13 ahora será aplicada al lado izquierdo de 1.10 y 1.12. Primero en 1.10:

$$m \frac{d\vec{V}_P}{dt} = m \left(\frac{\partial \vec{V}_P}{\partial t} + \vec{\omega} \times \vec{V}_P \right) \quad (1.14) \text{ lo que lleva a:}$$

$$m(\vec{V}_P + \vec{\omega} \times \vec{V}_P) = m\vec{g} + \vec{F}_A + \vec{F}_T \quad (1.15)$$

Luego, en el lado izquierdo de 1.12:

$$\begin{aligned} \frac{d}{dt} \int_V \vec{r} \times \frac{d\vec{r}}{dt} \zeta_A dv &= \int_V \vec{r} \times \frac{d}{dt} \frac{d\vec{r}}{dt} \zeta_A dv = \int_V \vec{r} \times \frac{d}{dt} (\vec{r} + \vec{\omega} \times \vec{r}) \zeta_A dv = \\ & \int_V \vec{r} \times \left(\vec{\dot{r}} + \vec{\omega} \times \vec{r} + 2\vec{\omega} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) \right) \zeta_A dv \end{aligned} \quad (1.16)$$

Asumiendo que los elementos de masa permanecen juntos y que no se cuenta con motores giratorios en el avión, se reconoce que $\vec{r} = \vec{r} = 0$ y entonces, la ecuación 1.12 se puede escribir como:

$$\int_V \vec{r} \times \left(\vec{\omega} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r}) \right) \zeta_A dv = \vec{M}_A + \vec{M}_T \quad (1.17)$$

Nótese que, desde $\vec{\omega}$ la aceleración angular en el sistema XYZ respecto al sistema X'Y'Z' es igual a la aceleración angular del avión respecto a la Tierra, propiedad que puede extraerse del signo integral de volumen. Esto hace la integral de volumen independiente del tiempo que es lo que se deseaba.

Las ecuaciones 1.15 y 1.17 representan las ecuaciones de movimiento del avión en la forma vectorial. Esta forma es útil para llegar a resultados de validez general e interpretaciones físicas. Sin embargo, estas formas no pueden emplearse para estudiar el estado de equilibrio estacionario y la respuesta en el tiempo del avión. Para cumplir esto último es necesario escribir las formas vectoriales en sus

equivalentes escalares. Antes de poder hacerlo, es necesario definir las componentes de todos los vectores que aparecen en las ecuaciones, los cuales se ven en la tabla 1. Las cantidades i, j, k son definidas como vectores unitarios en los ejes X, Y, Z respectivamente.

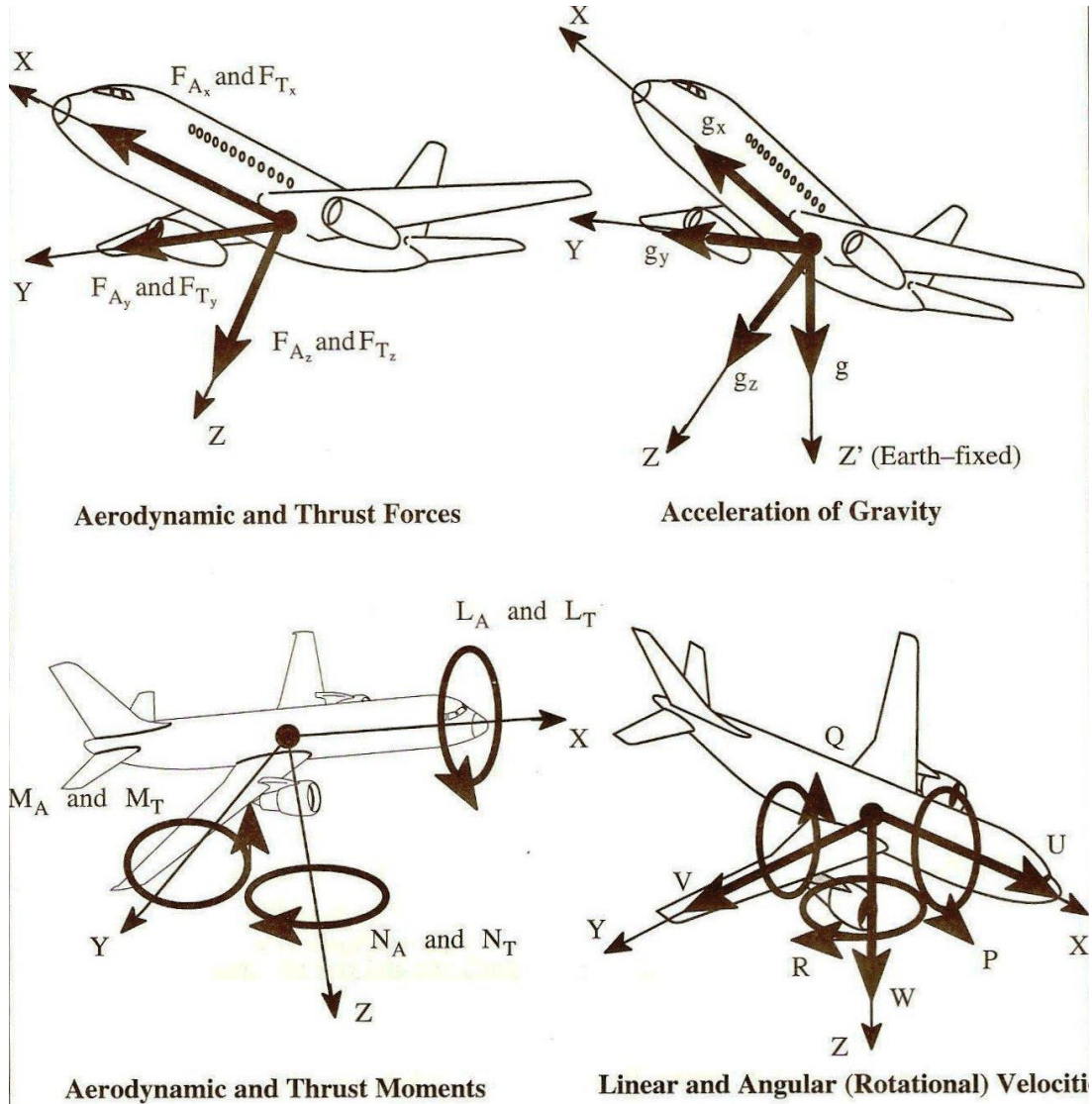


Figura 8. Definición de componentes vectoriales en las ecuaciones de movimiento

Colocando en formato escalar las ecuaciones de momento lineal

$$m(\dot{U} - VR + WQ) = mg + F_{Ax} + F_{Tx} \quad (1.19a)$$

$$m(\dot{V} + UR - WB) = mg + F_{Ay} + F_{Ty} \quad (1.19b)$$

$$m(\dot{W} - UQ - VP) = mg + F_{Az} + F_{Tz} \quad (1.19c)$$

Definición de las componentes de los vectores para las ecuaciones 1.15 y 1.17

Fuerzas

$$F_A = iF_{Ax} + jF_{Ay} + kF_{Az} \quad \text{Para las componentes de la fuerza aerodinámica. (1.18a)}$$

$$F_T = iF_{Tx} + jF_{Ty} + kF_{Tz} \quad \text{Para las componentes de la fuerza de empuje (1.18b)}$$

$$g = ig_x + jg_y + kg_z \quad \text{Para las componentes de la aceleración gravitacional (1.18c)}$$

Momentos

$$M_A = iL_A + jM_A + kN_A \quad \text{Para las componentes de momento aerodinámico: momento de alabeo, de cabeceo y guiñada respectivamente (1.18d)}$$

$$M_T = iL_T + jM_T + kN_T \quad \text{Para las componentes de momento debido al empuje: momento de alabeo, de cabeceo y guiñada respectivamente (1.18e)}$$

Velocidades

$$\omega = iP + jQ + kR \quad \text{Para las componentes de velocidad angular: tasa de alabeo, cabeceo y guiñada respectivamente (1.18f)}$$

$$V_p = iU + jV + kW \quad \text{Para las componentes de velocidad lineal: velocidad longitudinal, lateral y de trepada/descenso (1.18g)}$$

Distancias

$$r = ix + jy + kz \quad \text{Para las componentes de distancia que definen los elementos de masa en el avión (1.18h)}$$

Tabla 1. Definición de componentes vectoriales

Debido a la integral de volumen, la ecuación 1.17 es más difícil de expandir, por lo cual la expansión se hace en tres pasos:

Primero, el lado izquierdo de la ecuación 1.17 es re-escrito:

$$\int_V r \times (\bar{\omega} \times r + \omega \times r) \zeta_A dv = \int_V \bar{\omega} (r \cdot r) \zeta_A dv - \int_V r (r \cdot \bar{\omega}) \zeta_A dv + \int_V r \times \omega (\omega \cdot r) \zeta_A dv - \left[\int_V r \times r (\omega \cdot \omega) \zeta_A dv = 0 \right] \quad (1.20)$$

Luego los primeros dos términos de 1.20 se expanden:

$$\int_V \vec{\omega}(\vec{r} \cdot \vec{r}) \zeta_A dv = (i\dot{P} + j\dot{Q} + k\dot{R}) \int_V (x^2 + y^2 + z^2) \zeta_A dv \quad y$$

$$-\int_V \vec{r}(\vec{r} \cdot \vec{\omega}) \zeta_A dv = -\int_V (ix + jy + kz)(x\dot{P} + y\dot{Q} + z\dot{R}) \zeta_A dv$$

Combinando estas expresiones:

$$\begin{aligned} & i \left[\dot{P} \int_V (y^2 + z^2) \zeta_A dv - \dot{Q} \int_V xy \zeta_A dv - \dot{R} \int_V xz \zeta_A dv \right] + \\ & j \left[\dot{Q} \int_V (x^2 + z^2) \zeta_A dv - \dot{P} \int_V yx \zeta_A dv - \dot{R} \int_V yz \zeta_A dv \right] + \\ & k \left[\dot{R} \int_V (x^2 + y^2) \zeta_A dv - \dot{P} \int_V zx \zeta_A dv - \dot{Q} \int_V zy \zeta_A dv \right] \end{aligned} \quad (1.21)$$

Las integrales de volumen de la expresión 1.21 se refieren a los momentos y productos de inercia del avión. Los símbolos comúnmente usados para estas cantidades (inercias) son:

$$\int_V (y^2 + z^2) \zeta_A dv = I_{xx} \quad \int_V xy \zeta_A dv = I_{xy} \quad \int_V xz \zeta_A dv = I_{xz} \quad (1.22a)$$

$$\int_V (x^2 + z^2) \zeta_A dv = I_{yy} \quad \int_V yx \zeta_A dv = I_{yx} = I_{xy} \quad \int_V yz \zeta_A dv = I_{yz} = I_{zy} \quad (1.22b)$$

$$\int_V (x^2 + y^2) \zeta_A dv = I_{zz} \quad \int_V zx \zeta_A dv = I_{zx} = I_{xz} \quad \int_V zy \zeta_A dv = I_{zy} = I_{yz} \quad (1.22c)$$

Con ayuda de las ecuaciones 1.22, la expresión 1.21 se puede escribir:

$$i(\dot{P}I_{xx} - \dot{Q}I_{xy} - \dot{R}I_{xz}) + j(\dot{Q}I_{yy} - \dot{P}I_{xy} - \dot{R}I_{yz}) + k(\dot{R}I_{zz} - \dot{P}I_{xz} - \dot{Q}I_{yz}) \quad (1.23)$$

Finalmente, en el tercer término de 1.20:

$$\begin{aligned} & \int_V \vec{r} \times \vec{\omega}(\vec{\omega} \cdot \vec{r}) \zeta_A dv = \int_V \{ (ix + jy + kz) \times (iP + jQ + kR)(Px + Qy + Rz) \} \zeta_A dv = \\ & i \left[I_{xy} PR + I_{yz} (R^2 - Q^2) - I_{xz} PQ + (I_{zz} - I_{yy}) RQ \right] + \\ & j \left[(I_{xx} - I_{zz}) PR + I_{xz} (P^2 - R^2) - I_{xy} QR + I_{yz} PQ \right] + \\ & k \left[(I_{yy} - I_{xx}) PQ + I_{xy} (Q^2 - P^2) + I_{xz} QR - I_{yz} PR \right] \end{aligned} \quad (1.24)$$

En este punto se reconoce que muchos aviones son simétricos respecto al plano XZ. Si ese es (o se aproxima) el caso automáticamente se deduce que: $I_{xy} = I_{yz} = 0$.

Ahora se dispone de todos los elementos para convertir la ecuación de momento angular (1.17) en su forma de componentes escalares. Usando la expresión 1.23 y 1.24 se tiene:

$$I_{XX}\dot{P} - I_{XZ}\dot{R} - I_{XZ}PQ + (I_{ZZ} - I_{YY})RQ = L_A + L_T \quad (1.25a)$$

$$I_{YY}\dot{Q} + (I_{XX} - I_{ZZ})PR + I_{XZ}(P^2 - R^2) = M_A + M_T \quad (1.25b)$$

$$I_{ZZ}\dot{R} - I_{XZ}\dot{P} + (I_{YY} - I_{XX})PQ + I_{XZ}QR = N_A + N_T \quad (1.25c)$$

Las ecuaciones 1.19 y 1.25 forman seis ecuaciones diferenciales de movimiento con U, V, W, P, Q y R como variables dependientes, y el tiempo como variable independiente.

1.2.2 Orientación del avión relativa al sistema de coordenadas fijo a la Tierra X'Y'Z'

Para definir la orientación de un avión relativa a las coordenadas fijadas a la Tierra es suficiente definir la orientación de su sistema XYZ fijado al cuerpo. La figura 7 muestra los dos ejes sin definir su interrelación. En la figura 9 se muestra el sistema X'Y'Z' trasladado paralelo a sí mismo hasta que su origen coincide con el centro de masas, punto P de la nave. Este sistema trasladado ha sido rebautizado $X_1Y_1Z_1$. La orientación relativa del sistema de ejes XYZ al $X_1Y_1Z_1$ es definida con ayuda de tres rotaciones secuenciales sobre los denominados ángulos de Euler: Ψ , Θ y Φ .

Rotación 1: El sistema de coordenadas $X_1Y_1Z_1$ es rotado con respecto al eje Z_1 sobre un ángulo Ψ que es el llamado ángulo de guiñada. El ángulo es positivo como se muestra en la figura 9. Después de rotar sobre el ángulo Ψ el sistema de coordenadas es denominado $X_2Y_2Z_2$.

Rotación 2: El sistema de coordenadas $X_2Y_2Z_2$ es rotado con respecto al eje Y_2 sobre un ángulo Θ que es el llamado ángulo de actitud de cabeceo. El ángulo es positivo como se muestra en la figura 9. Después de rotar sobre el ángulo Θ el sistema de coordenadas es denominado $X_3Y_3Z_3$.

Rotación 3: El sistema de coordenadas $X_3Y_3Z_3$ es rotado con respecto al eje X_3 sobre un ángulo Φ que es el llamado ángulo de alabeo. El ángulo es positivo como se muestra en la figura 9. Después de rotar sobre el ángulo Φ el sistema de coordenadas es denominado XYZ .

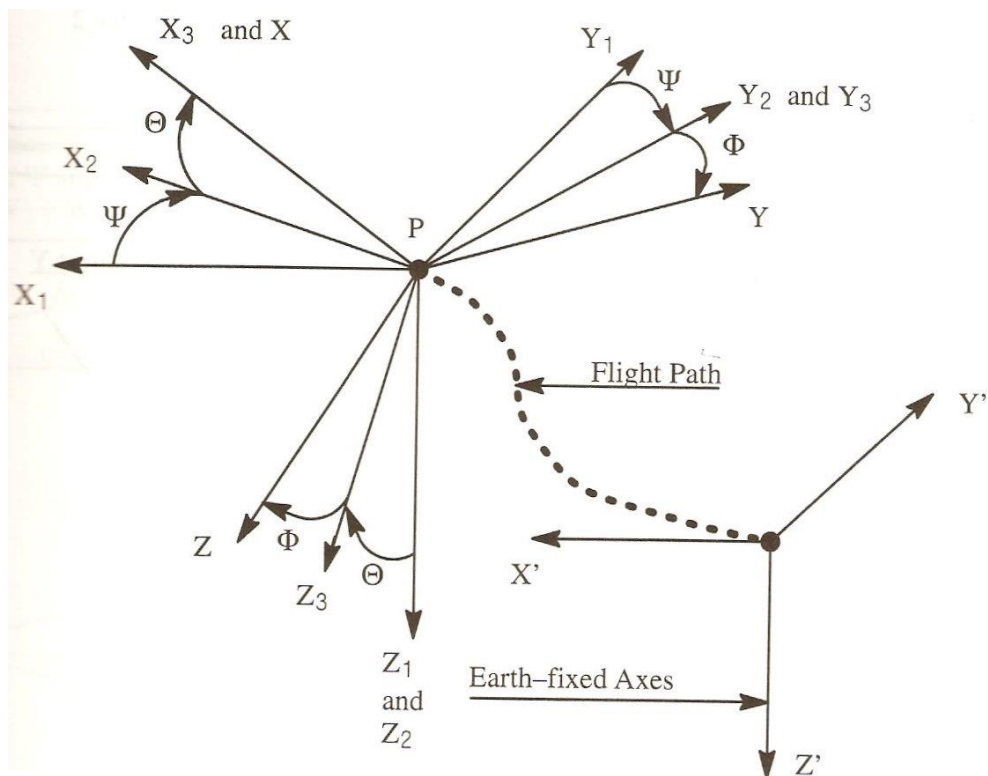


Figura 9. Orientación del avión con ángulos de Euler

En la figura 10 se muestra porqué la definición de los ejes sobre los que se hacen las rotaciones angulares es importante: las rotaciones sobre ángulos finitos no se comportan como vectores que tienen propiedad conmutativa. Un problema con el uso de ángulos de Euler es que para $\Theta = 90^\circ$, el ángulo Φ pierde importancia. En simulaciones donde maniobras del tipo rizo deben realizarse, esto no es aceptable, para lo cual se emplean métodos más sofisticados.

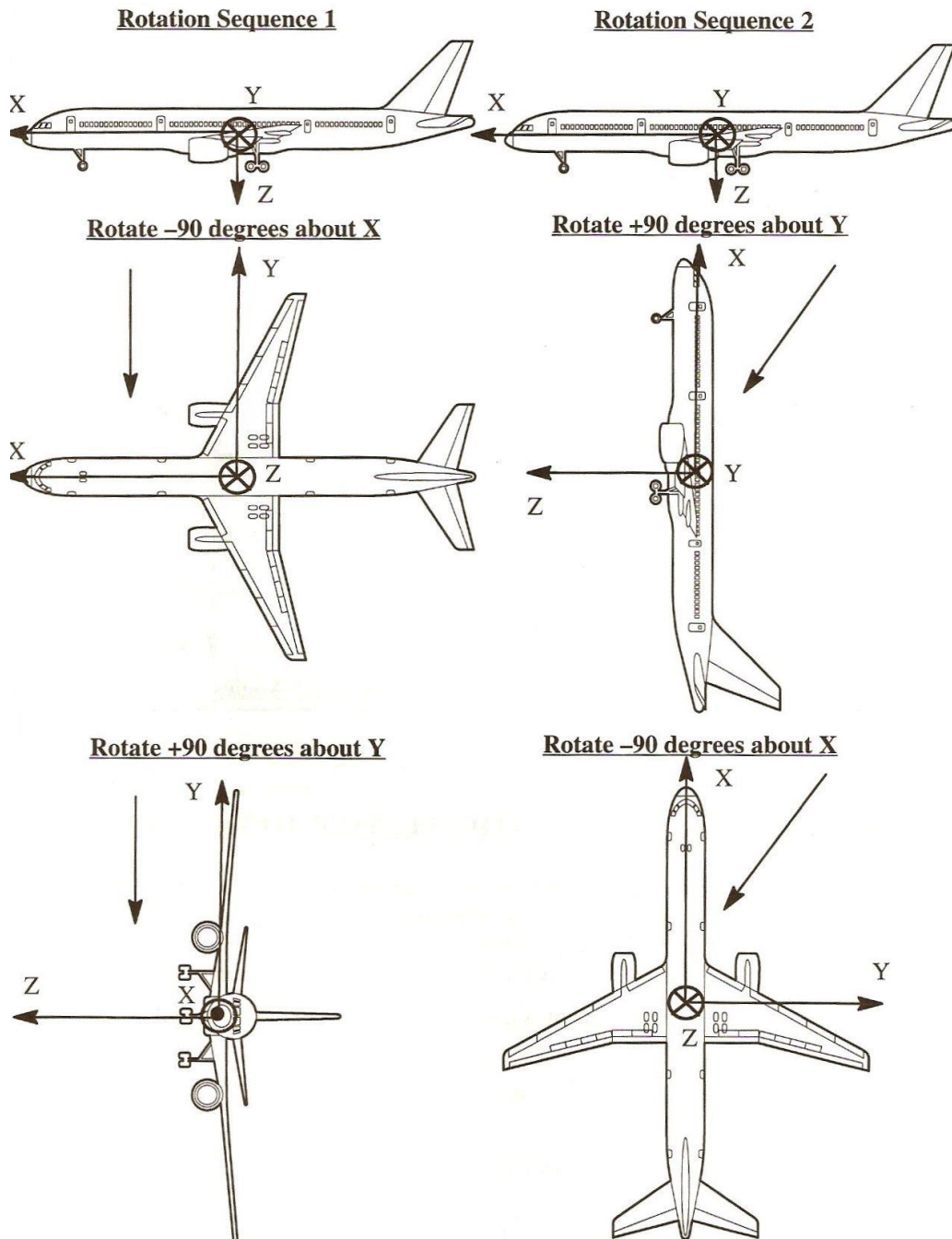


Figura 10. Demostración del comportamiento de las rotaciones de ángulos finitos

1.2.3 Desplazamiento en vuelo del avión relativo a la Tierra

Es posible determinar la trayectoria de un avión relativa a la Tierra conociendo los componentes de los vectores U , V y W en el sistema de coordenadas XYZ centrado en el cuerpo de la nave y los ángulos de Euler previamente vistos. Esto se hace estableciendo la relación entre las componentes de velocidad en XYZ con las

correspondientes en $X'Y'Z'$. Como $X_1Y_1Z_1$ y $X'Y'Z'$ son paralelas entre sí se tiene que:

$$U_1 = \dot{x}' \quad V_1 = \dot{y}' \quad W_1 = \dot{z}' \quad (1.26)$$

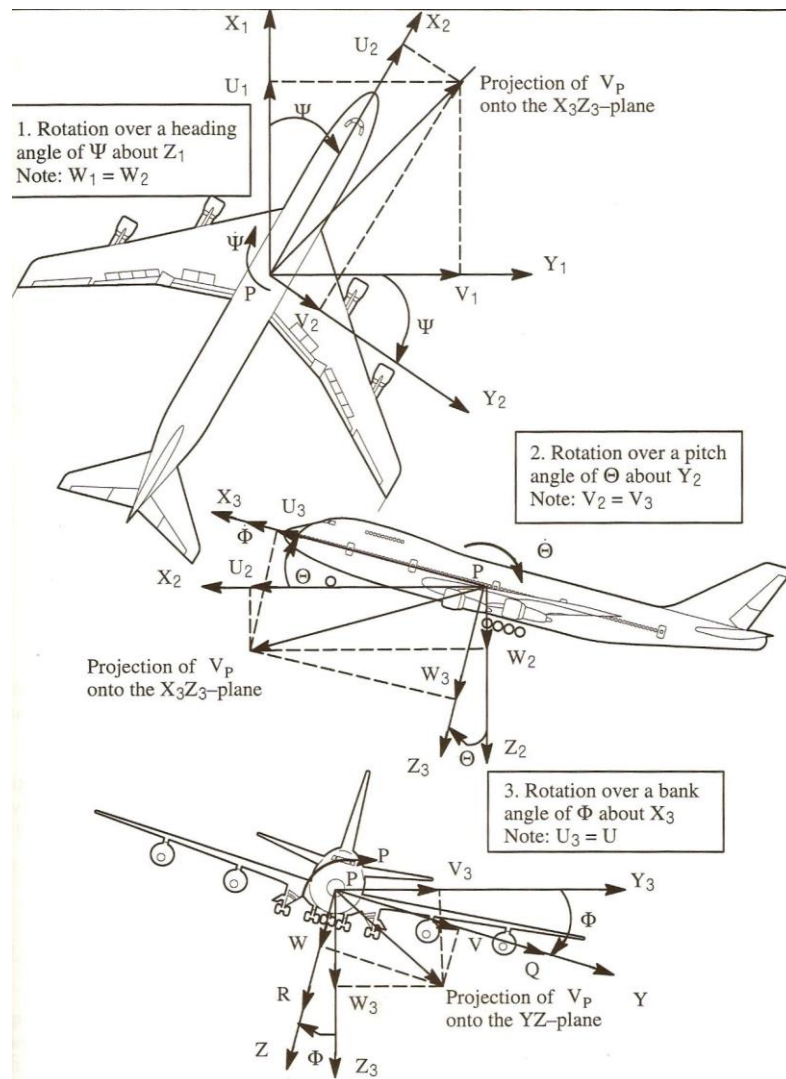


Figura 11. Relación de los vectores de velocidades lineal y angular

De la figura 11 se puede verificar que siguientes relaciones entre U_1, V_1, W_1 y U_2, V_2, W_2 .

$$\begin{bmatrix} U_1 \\ V_1 \\ W_1 \end{bmatrix} = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_2 \\ V_2 \\ W_2 \end{bmatrix} \quad (1.27)$$

Con una transformación ortogonal similar es posible relacionar

$$\begin{Bmatrix} U_2 \\ V_2 \\ W_2 \end{Bmatrix} = \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix} \begin{Bmatrix} U_3 \\ V_3 \\ W_3 \end{Bmatrix} \quad (1.28)$$

Finalmente, se establece

$$\begin{Bmatrix} U_3 \\ V_3 \\ W_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix} \begin{Bmatrix} U \\ V \\ W \end{Bmatrix} \quad (1.29)$$

Sustituyendo 1.29 en 1.28

$$\begin{Bmatrix} U_1 \\ V_1 \\ W_1 \end{Bmatrix} = \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix} \begin{Bmatrix} U \\ V \\ W \end{Bmatrix} \quad (1.30)$$

La ecuación 1.30 proporciona las relaciones deseadas entre los componentes de velocidad en el sistema X'Y'Z' respecto a Tierra y los del sistema XYZ centrado en el avión.

La trayectoria del avión en términos de $x'(t), y'(t), z'(t)$ puede hallarse por la integración de 1.30. Para esto, los ángulos de Euler deben ser conocidos, sin embargo estos en si mismos son función del tiempo, por lo que se debe satisfacer la igualdad:

$$\vec{\omega} = iP + jQ + kR = \vec{\Psi} + \vec{\Theta} + \vec{\Phi} \quad (1.31)$$

Como Ψ representa una variación angular respecto al eje Z_1 se ve de la figura 11 que:

$$\vec{\Psi} = k_1 \dot{\Psi} = k_2 \dot{\Psi} \quad (1.32)$$

de manera similar $\vec{\Theta}$ representa una variación angular sobre el eje Y_2 , por lo que de la misma figura se ve que:

$$\vec{\Theta} = j_2 \dot{\Theta} = j_3 \dot{\Theta} \quad (1.33)$$

finalmente $\vec{\Phi}$ representa una variación angular sobre el eje X_3 , por lo que se tiene:

$$\vec{\Phi} = i_3 \dot{\Phi} = i \dot{\Phi} \quad (1.34)$$

Por sustitución de 1.32, 1.33 y 1.34 en 1.32, tenemos:

$$\omega = k_2 \dot{\Psi} + j_3 \dot{\Theta} + i \dot{\Phi} \quad (1.35)$$

Luego, usando transformaciones similares a la ecuación 1.29 se puede demostrar que:

$$k_2 = -i_3 \sin \Theta + k_3 \cos \Theta = -i \sin \Theta + \cos \Theta (j \sin \Phi + k \cos \Phi) \quad (1.36)$$

$$j_3 = j \cos \Phi - k \sin \Phi \quad (1.37)$$

Después de sustituir 1.36 y 1.37 en 1.35 y reacomodar se tiene

$$\omega = i(-\dot{\Psi} \sin \Theta + \dot{\Phi}) + j(\dot{\Psi} \cos \Theta \sin \Phi + \dot{\Theta} \cos \Phi) + k(\dot{\Psi} \cos \Theta \cos \Phi - \dot{\Theta} \sin \Phi) \quad (1.38)$$

Por comparación con 1.31 ahora tenemos las denominadas ecuaciones cinemáticas del avión:

$$\begin{aligned} P &= \dot{\Phi} - \dot{\Psi} \sin \Theta \\ Q &= \dot{\Theta} \cos \Phi + \dot{\Psi} \cos \Theta \sin \Phi \\ R &= \dot{\Psi} \cos \Theta \cos \Phi - \dot{\Theta} \sin \Phi \end{aligned} \quad (1.39)$$

Para el problema de integración de la trayectoria de vuelo se invierten las ecuaciones para lograr:

$$\begin{aligned} \dot{\Phi} &= P + Q \sin \Theta \sin \Phi + R \cos \Theta \sin \Phi \\ \dot{\Theta} &= Q \cos \Phi - R \sin \Phi \\ \dot{\Psi} &= (Q \sin \Theta + R \cos \Theta) \sin \Phi \end{aligned} \quad (1.40)$$

1.2.4 Componentes de la fuerza de gravedad

En las ecuaciones 1.19 se vio que aparecieron las tres componentes de la fuerza de gravedad: mg_x, mg_y, mg_z . Haciendo referencia a la figura 8, las componentes de la aceleración gravitacional pueden escribirse como:

$$\vec{g} = k_1 \vec{g} = k_1 g \equiv i g_x + j g_y + k g_z \quad (1.41)$$

Estas componentes se pueden poner en función de los ángulos de Euler reconociendo que $k_1 = k_2$. Dado que k_2 ya fue expresada en términos de i, j y k por 1.36 se deduce que:

$$i(-g \sin \theta) + j(g \sin \theta \cos \Phi) + k(g \cos \theta \cos \Phi) \equiv i g_x + j g_y + k g_z \quad (1.42)$$

Viéndose que:

$$\begin{aligned} g_x &= -g \sin \theta \\ g_y &= g \sin \theta \cos \Phi \\ g_z &= g \cos \theta \cos \Phi \end{aligned} \quad (1.43)$$

1.2.5 Ecuaciones de movimiento

Ecuaciones de fuerzas en el sistema de coordenadas XYZ fijado al cuerpo de la aeronave

$$\text{Fuerza en el eje x: } m(U - VR + WQ) = -mg_x + F_{Ax} + F_{Tx}$$

$$\text{Fuerza en el eje y: } m(V + UR - WP) = mg_y + F_{Ay} + F_{Ty}$$

$$\text{Fuerza en el eje z: } m(W - UQ + VP) = mg_z + F_{Az} + F_{Tz}$$

Ecuaciones de momentos en el sistema de coordenadas XYZ fijado al cuerpo de la aeronave

$$\text{Momento de alabeo respecto a X: } I_{xx} \dot{P} - I_{xz} \dot{R} - I_{xz} PQ + (I_{zz} - I_{yy}) RQ = L_A + L_T$$

$$\text{Momento de cabeceo respecto a Y: } I_{yy} \dot{Q} + (I_{xx} - I_{zz}) PR + I_{xz} (P^2 - R^2) = M_A + M_T$$

$$\text{Momento de guiñada respecto a Z: } I_{zz} \dot{R} - I_{xz} \dot{P} + (I_{yy} - I_{xx}) PQ + I_{xz} QR = N_A + N_T$$

Para las ecuaciones cinemáticas:

$$\text{Tasa de alabeo respecto a X: } P = \dot{\Phi} - \dot{\Psi} \sin \Theta$$

$$\text{Tasa de cabeceo respecto a Y: } Q = \dot{\Theta} \cos \Phi + \dot{\Psi} \cos \Theta \sin \Phi$$

$$\text{Tasa de guiñada respecto a Z: } R = \dot{\Psi} \cos \Theta \cos \Phi - \dot{\Theta} \sin \Phi$$

ECUACIONES DE MOVIMIENTO

Ecuaciones de fuerzas en el sistema de coordenadas XYZ fijado al cuerpo de la aeronave

$$\text{Fuerza en el eje x: } m(U - VR + WQ) = -mg \sin \Theta + F_{Ax} + F_{Tx}$$

$$\text{Fuerza en el eje y: } m(V + UR - WP) = mg \cos \Theta + F_{Ay} + F_{Ty}$$

$$\text{Fuerza en el eje z: } m(W - UQ - VP) = mg \cos \Theta + F_{Az} + F_{Tz}$$

Ecuaciones de momentos en el sistema de coordenadas XYZ fijado al cuerpo de la aeronave

$$\text{Momento de alabeo respecto a X: } I_{xx} \dot{P} - I_{xz} \dot{R} - I_{xz} PQ + (I_{zz} - I_{yy}) RQ = L_A + L_T$$

$$\text{Momento de cabeceo respecto a Y: } I_{yy} \dot{Q} + (I_{xx} - I_{zz}) PR + I_{xz} (P^2 - R^2) = M_A + M_T$$

$$\text{Momento de guiñada respecto a Z: } I_{zz} \dot{R} - I_{xz} \dot{P} + (I_{yy} - I_{xx}) PQ + I_{xz} QR = N_A + N_T$$

Para las ecuaciones cinemáticas:

$$\text{Tasa de alabeo respecto a X: } P = \dot{\Phi} - \dot{\Psi} \sin \Theta$$

$$\text{Tasa de cabeceo respecto a Y: } Q = \dot{\Theta} \cos \Phi + \dot{\Psi} \cos \Theta \sin \Phi$$

$$\text{Tasa de guiñada respecto a Z: } R = \dot{\Psi} \cos \Theta \cos \Phi - \dot{\Theta} \sin \Phi$$

Tabla 2. Ecuaciones de movimiento

1.3 La percepción del movimiento espacial en el ser humano

El ser humano percibe el movimiento en el espacio a través de tres canales:

- El sistema vestibular, compuesto por el otolito y los canales semicirculares. Provee la sensación de movimiento por aceleraciones traslacionales y rotacionales de la aeronave. Los canales semicirculares perciben las aceleraciones angulares, mientras que el otolito responde a aceleraciones lineales (traslacionales). Este sistema tiene un umbral de aceleración que debe ser superado para ser percibido por el piloto, el cual para aceleraciones lineales se estima en 10 cm/s^2 [2]. Para las aceleraciones angulares puede considerarse entre 0.44 y 0.80 grados/ s^2 , sin considerar aplicación simultánea de estímulo visual, en presencia del cual este umbral pasa a estar entre 0.10 y 0.12 grados/ s^2 .

Los canales semicirculares cuentan con una masa de fluido líquido que añade inercia, por lo cual los efectos sobre este exceden la duración del estímulo, llegando a generar desorientación espacial en el piloto.

- La piel percibe vibraciones, lo cual es aprovechado por algunos dispositivos de simulación que aplican efectores en la espalda, glúteos y muslos del piloto para reproducir condiciones propias de los disturbios.
- El sistema propioceptivo detecta esfuerzo muscular y posición de las articulaciones, esta última debido a la acción directa del vehículo sobre el cuerpo (por ejemplo las partes en contacto con un asiento), mientras que otras tienden a moverse como producto de la inercia. El esfuerzo muscular se presenta para contrarrestar movimientos relativos no deseados, y existen sensaciones de movimiento que son percibidas por la piel.

En un vehículo real, estas informaciones están interrelacionadas de una forma natural y perfecta [3]. Se conoce ahora que los mecanismos de la percepción humana no son perfectos y esto puede ser usado para que la información visual y de movimiento simulado, cuya interrelación es imperfecta producto de su naturaleza, sea aceptada como válida. La más útil *“imperfección”* es la incapacidad humana de percibir directamente la velocidad lineal por otra vía que no sea la visual. Una simulación puede presentar una sensación de movimiento simplemente con una escena mientras el simulador en sí permanece inmóvil.

Imperfecciones menos conocidas son los umbrales bajo los cuales el movimiento no puede ser detectado. Esto se emplea después de un período de movimiento del simulador para devolver la cabina a una posición neutral con aceleraciones que no son percibidas.

Asimismo, otra importante consideración en el diseño de sistemas de movimiento es el efecto sobre los ocupantes: aunque muchas personas sufren desordenes al ser sometidos a estímulos visuales y motores consistentes a bordo de un vehículo, la ausencia de movimiento también puede ser perjudicial en tareas de entrenamiento especializado.

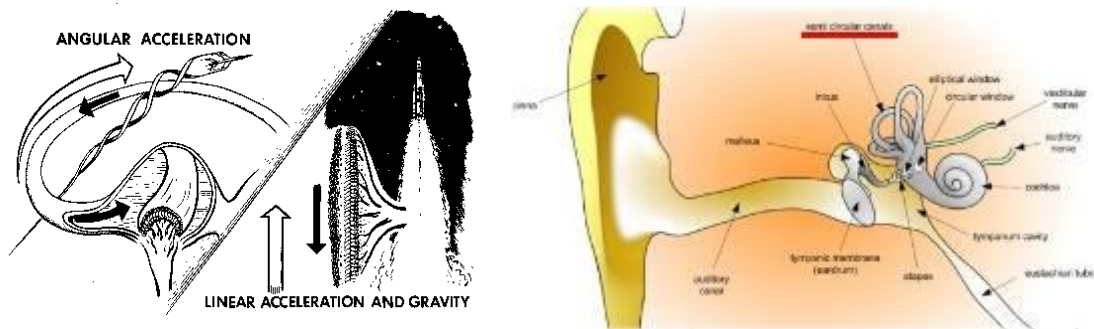


Figura 12. Percepción de aceleración angular y lineal en el oído interno

1.4 La simulación del vuelo

La simulación es ampliamente empleada en el entrenamiento de tripulaciones aéreas, debido a que permite probar a los pilotos en condiciones límite sin poner en riesgo a la aeronave. Algunos conceptos sobre simulación son:

“Representación imitativa del funcionamiento de un sistema o proceso a través del funcionamiento de otro” [4].

“Representación de objetos, sistemas o procesos en términos matemáticos y el uso de estos sustitutos para representar la dinámica de los equivalentes en el mundo real” [5].

“Creación de una representación dinámica del comportamiento de una aeronave de una forma que permite al operador humano interactuar con la simulación como parte de la simulación” [6].

1.4.1 Objetivos de un Simulador:

El objetivo básico es hacer creer al ser humano que esta experimentando una dinámica de movimientos continuos en un entorno infinito. Sin embargo, esto debe lograrse en un espacio limitado, mediante una combinación de estímulos visuales y de movimiento que genera entradas al sistema fisiológico. Para asegurar que el sistema sensorial humano recibe información coherente, debe atenderse a varios factores:

- a) Los datos de audio, visión y movimiento deben ser sincronizados. Esto requiere un sistema de movimiento que:
 - Proporcione amplitud de movimiento consistente con la imagen visual.
 - Proporcione dinámica de vuelo consistente con el rango de respuesta del cuerpo humano
- b) El entorno debe ayudar a lograr la experiencia deseada:
 - Una cabina consistente con el objetivo.
 - Simulación visual envolvente
 - Presentación de audio realista
- c) Deben incluirse disturbios del vuelo
 - Ruido
 - Vibraciones

1.4.2 Evolución histórica de la simulación

Debido a que los vuelos reales son peligrosos para pilotos sin un entrenamiento previo, desde los primeros días de la aviación, diversos esquemas fueron usados para que los pilotos pudieran sentir la sensación de volar sin ser realmente aerotransportados. Por ejemplo el "*Sander Teacher*" fue un aeroplano completo montado en una articulación universal que era orientado hacia el viento

con capacidad de girar y de inclinarse libremente. Otro simulador de vuelo de los primeros días fue desarrollado en 1910 a partir de un barril montado en un marco.



Figura 13. “Simulador” de vuelo de comienzos del siglo XX.

Un buen número de dispositivos electromecánicos fueron probados durante la Primera Guerra Mundial y posteriormente. El más conocido es el "*Link Trainer*", el que en 1930 sólo simulaba movimientos mecánicos aunque posteriormente se le incluyeron Instrumentos de control (avión) y fue utilizado por algunos países durante la Segunda Guerra Mundial e incluso después.

El "*Celestial Navigation Trainer*" de 1941 fue una enorme estructura de 13,7 m capaz de acomodar en su interior una tripulación entera de un bombardero para aprender como volar en misiones nocturnas. En 1940 las computadoras analógicas fueron usadas para resolver las ecuaciones de vuelo, dando lugar a los primeros simuladores electrónicos.

En 1948, Curtiss-Wright desarrolló un simulador para el Stratocruiser de Pan American, el primer simulador de vuelo completo utilizado por una aerolínea. Aunque no se había simulado movimiento ni vistas exteriores, la cabina entera y sus instrumentos funcionaban de forma que la tripulación lo encontró muy efectivo. A partir de ese momento los sistemas de movimiento llegaron y se instalaron a finales de los años 50.



Figura 14 Simuladores de combate de la Segunda Guerra Mundial

Los primeros sistemas *"full motion"* solían simular el terreno usando un modelo a escala del mismo y haciendo elevar una cámara sobre él con la misma posición en la cual se supone que debía estar la aeronave. El resultado eran imágenes proyectadas en un televisor. Naturalmente el área simulada era muy pequeña y habitualmente se limitaba a los alrededores del aeropuerto, un sistema similar al que utilizaron los militares para simular bombardeos. El uso de computadoras digitales para la simulación de vuelo empezó en los años 60.

En 1954 General Precision Inc., posteriormente parte de Singer Corporation, desarrolló un simulador con movimiento el cual contenía una cabina dentro de un marco metálico. En 1969 fueron desarrollados simuladores de aerolínea con actuadores hidráulicos controlando cada eje de movimiento. Los simuladores empezaron a ser construidos con seis ejes de movimiento (cabeceo, guiñada y alabeo para los movimientos angulares y movimientos horizontales, verticales y laterales). A partir de 1977, los simuladores de aerolínea empezaron a adoptar las nuevas cabinas en donde los computadores estaban instalados en el interior de éstas.



Figura 15. Simulador de avión BAC Comet, con sistema hidráulico de movimiento

A partir de este momento los mayores avances se realizaron en tecnologías de visualización. En 1972 Singer desarrolló una lente policromada utilizando un espejo esférico que proyectaba las vistas del exterior de la cabina. Esto mejoró sustancialmente la sensación de vuelo. Sin embargo solamente ofrecía un campo de visión de 28 grados. En 1976 se introdujeron las imágenes enfocadas. Hacia los 80 las imágenes proyectadas de alta definición se convirtieron en el estándar de los simuladores de aerolínea

1.4.3. Tipos de sistemas

Contrariamente a la creencia popular, hoy en día los simuladores modernos no son tan utilizados para el entrenamiento de pilotos en como volar, como para enseñar al piloto procedimientos normales y de emergencia en pleno vuelo. Utilizando los simuladores, los pilotos pueden practicar situaciones que no pueden ser practicadas de forma segura en una aeronave real. Estas situaciones incluyen desprendimientos de superficies de vuelo y pérdidas completas de potencia. Hoy en día las aeronaves están basadas en sistemas complejos de computadoras y para poderla operar de forma eficiente los pilotos deben poseer un buen nivel técnico así como habilidades de pilotaje.

La mayoría de las instituciones gubernamentales tales como la estadounidense Administración Federal de Aviación (FAA) emiten normas

reguladoras para certificar las características de los dispositivos [7]. Los pilotos comerciales estadounidenses pueden anotarse las horas de vuelo requeridas únicamente en simuladores certificados por la FAA. Para que un simulador sea certificado debe demostrar que sus características de vuelo coinciden con las del avión que está siendo simulado. Los requerimientos a testear están detallados en las guías denominadas Guías de Test de Aprobación (ATG) o Guías de Test de Calificación (QTG).

Los entrenadores del sistema (flight training devices o FTD's) son usados para enseñar a los pilotos como operar apropiadamente varios sistemas de la aeronave. Habitualmente no son usados para enseñar a volar ni para entrenamiento de procedimientos de emergencia.



Figura 7 . Flight training device

Una vez que los pilotos se familiarizan con los sistemas operacionales se realiza la transición a los entrenadores de procedimientos de cabina (cockpit procedure trainer o CPT's). Estos sistemas fijos son copias exactas de una aeronave y son usadas para entrenar a las tripulaciones de vuelo en procedimientos normales y de emergencia. Estos reproducen el entorno atmosférico en el que la aeronave volará, simulando vientos, temperaturas y turbulencias. CPTs simulan también la variedad de sonidos provocados por la aeronave como por ejemplo los motores, trenes de aterrizaje y otros. Algunos son equipados con sistemas visuales. A pesar

de todo estos no son equipados con sistemas que simulen los movimientos de la aeronave.



Figura 17. Cockpit procedure trainer

Un full motion Simulator (simulador de movimiento completo) o full-flight Simulator (FFS), replica todos los aspectos de una aeronave y su entorno, incluyendo los movimientos básicos de la aeronave. Este tipo de simuladores pueden generar sacudidas momentáneas de forma que sus ocupantes en el simulador deban abrocharse los cinturones de los asientos tal y como harían en una aeronave real. Dado que el movimiento de cualquier simulador es mucho más restrictivo que los movimientos de una aeronave real, el sistema de movimientos no calca los movimientos y actitud del avión. En lugar de ello realiza las llamadas señales de movimiento las cuales engañan las sensaciones del piloto y le hacen creer que está volando. Para poder hacer esto apropiadamente, el conocimiento de los órganos de los sentidos del ser humano, particularmente del sistema vestibular, es empleado extensivamente. Esto convierte a la simulación de vuelo en un área de conocimientos intensivos.



Figura 18. Full motion simulator del AirbusA320

Los simuladores de vuelo son extensamente utilizados en investigación en diversas áreas aeroespaciales, particularmente en la dinámica de vuelo y en la interacción hombre-máquina. Tanto los simuladores estándar como los específicamente construidos para investigación se utilizan para este propósito. El rango va desde la gama más simple, parecida a los videojuegos, hasta los diseños específicos y extremadamente caros como LAMARS, instalados en la base Wright-Patterson Air Force, Ohio [8], el cual permite 5 ángulos de movimiento y un sistema visual con más de 180 grados de campo de visión en todas direcciones.



Figura 19. Large Amplitude Multi Mode Aerospace Research Simulator (LAMARS)

De acuerdo al Manual de Criterios para calificar los simuladores de vuelo de la Organización de Aviación Civil Internacional, un simulador de vuelo se define como *“una réplica de tamaño real de un tipo específico, o de una marca, modelo y serie de puesto de pilotaje de aeronave, incluido el ensamblaje de equipo y programas de computadora necesarios para representar las operaciones de la aeronave en tierra y en vuelo, junto con un sistema de visualización que proporciona la perspectiva desde el puesto de pilotaje y un sistema de movimientos con fuerzas de referencia. En este se satisfacen las normas mínimas para la calificación del simulador:*

- a) *Simulador de vuelo de nivel I. Debe proporcionarse un sistema de movimientos respecto a seis ejes. En la simulación de sonidos deben incluirse los de precipitaciones atmosféricas y otros ruidos importantes de la aeronave que perciba el piloto y deben reproducirse los sonidos de un aterrizaje accidentado. La respuesta a las acciones de mando no deben ser en un tiempo superior a 150 milisegundos respecto a los valores que se observan en la aeronave. Debe proporcionarse la simulación de cortante del viento.*

b) Simulador de vuelo de nivel II. El nivel máximo de actuación simulada. Además de los requisitos correspondientes a los simuladores de vuelo de nivel I, se requiere un sistema de visualización completa de condiciones diurnas/ crepusculares/ nocturnas y deben reproducirse con completa fidelidad sonidos y bataneo de movimientos.” [9]

1.4.4 Simulación de movimiento

El propósito de la simulación de movimientos y fuerzas es proporcionar información del entorno crítica para las tareas y cualquier otro componente requerido de incremento de la carga de trabajo en vuelo inducido por un estresor que puede estar presente en el vuelo. [10]

Las consideraciones de diseño para la simulación de movimiento son:

- El movimiento afecta la performance del piloto
- La replicación de movimiento rotacional y lineal de manera simultánea es imposible en un simulador fijo en tierra
- Enfatizar en la fidelidad sensorial y perceptual tiene mas beneficios potenciales que tratar de mantener la fidelidad física

1.4.5 Desarrollos de la Fuerza Aérea del Perú

El CEDEP ha construido simuladores para el avión CESSNA A-37B y Tucano EMB-312 (figura 20), los cuales tienen más de siete mil y tres mil quinientas horas de operación respectivamente. Adicionalmente ha construido entrenadores de vuelo para el avión Tucano T-27 y Zlin Z-242L (figura 21).



Figura 20. Simulador de avión Cessna A-37B (izq.) y Tucano T-27 (der.)



Figura 21. Entrenador de avión Tucano T-27 (izq.) y Zlin Z-242L (der.)

En el simulador del avión Tucano T-27 fabricado por el CEDEP, hasta el momento, han sido evaluadas siete promociones de pilotos, habiendo acumulado más de siete mil horas de vuelo. Este simulador ha permitido realizar el entrenamiento básico y avanzado, ya que tiene modelado el comportamiento de los sistemas del avión, tales como Sistemas de Radionavegación, de Combustible, Eléctrico, de Armamento y de Comunicaciones, permitiendo de esta manera efectuar los vuelos en condiciones visuales, instrumentales, operaciones nocturnas, misiones de navegación y misiones de combate.

1.5 Aspectos metodológicos del estudio a realizar

1.5.1 Planteamiento del problema

Para este proyecto se ha definido como requerimiento a cubrir el dotar de movimiento en seis grados de libertad a entrenadores de aeronaves de baja performance, debido a que sus movimientos son relativamente lentos en comparación a las aeronaves supersónicas, y no efectúan virajes bruscos, lo cual facilita la labor de control.

En la actualidad estas estructuras son estáticas y por tanto los pilotos no perciben como reales las sensaciones visuales, lo cual tiene efectos sobre su desempeño en vuelo real al sentirse sorprendidos por la respuesta de la cabina ante maniobras que, al ser practicadas en simuladores o entrenadores, no producen mayor acción física.

El problema a resolver es: diseñar e implementar el prototipo de una plataforma de movimiento de seis grados de libertad enlazada a un software de simulación de vuelo, que permita reproducir las sensaciones de movimiento en el espacio como si se estuviese al interior de una aeronave.

1.5.2 Hipótesis

Se plantea el diseño y construcción de una plataforma de movimiento de seis grados de libertad que permita dar movimiento a simuladores de aeronave de baja performance, para aumentar la percepción de realidad de dichos sistemas a través de la sensación de desplazamiento en el espacio por los pilotos.

Variable dependiente: La percepción de realidad que se logra a través de la simulación del vuelo, que está condicionada por los estímulos que reciben los sentidos del piloto en la medida que estos se acercan a los que presentan durante el vuelo.

Variable independiente: El movimiento a ser logrado a través de la plataforma propuesta, que se aplicará a una estructura de manera controlada y enlazada a un programa de simulación.

1.5.3 Objetivos de la investigación

- Proponer un sistema de control para una plataforma de movimiento de seis grados de libertad aplicable a simuladores de vuelo, con comprobación a través de enlace a software de simulación.
- Implementar un prototipo a escala que permita validar el diseño propuesto y que sirva de banco de prueba para este tipo de sistemas.

1.5.4 Metodología del estudio

El estudio que se propone es de tipo exploratorio, debido a que se plantea la solución a un problema poco estudiado o que no ha sido abordado antes en nuestro medio [11].

Parte de la recolección de datos se realizará a través de la búsqueda de patentes y desarrollos similares realizados en otros países, a fin de obtener otros puntos de vista y de ser posible, analizar las virtudes y defectos de cada sistema que permitan lograr un buen modelo.

Capítulo 2

Descripción de la plataforma Stewart

Pese a que la estructura del manipulador, a la cual esta plataforma debe su nombre, fue propuesta en 1965, ha evolucionado convirtiéndose en un tópico de investigación popular en la robótica recién a partir de los años 80. En los años 90, se manifestó un interés creciente en el campo de los manipuladores paralelos en general y de la plataforma Stewart en particular, como puede desprenderse de las publicaciones en revistas técnicas y conferencias relacionadas a robótica. Solamente en 1995, más de 50 artículos de divulgación fueron publicados sobre el tema.

2.1 Manipuladores paralelos

La ciencia y tecnología de los robots se originó con el propósito de desarrollar sistemas mecánicos que pudieran llevar a cabo tareas normalmente realizadas por el ser humano, por lo tanto es natural que el primer enfoque se orientase al uso de cadenas seriales de lazo abierto. Estos robots tenían la ventaja de cubrir amplios espacios de trabajo con gran maniobrabilidad similar al brazo de una persona, pero su capacidad de carga era pobre debido a su misma estructura, en consecuencia, para consideraciones de fuerza, las articulaciones difícilmente resistían grandes pesos y producían vibraciones al ser sometidas a grandes esfuerzos de peso y velocidad, perdiendo precisión en su posicionamiento.

Debido a esto, para aplicaciones donde se requiriese capacidad de carga, buen desempeño dinámico y posicionamiento de precisión, era deseable tener una alternativa a los manipuladores seriales convencionales. Entre las posibles soluciones, una tomó como ejemplo los mecanismos de la propia naturaleza notando que los cuerpos de los animales son más estables al ser soportados por múltiples piernas en paralelo comparados con los bípedos como el hombre, y que aun los seres humanos usan ambos brazos para manejar grandes pesos, mientras que procedimientos de gran coordinación como la escritura requieren la participación de tres dedos que actúan simultáneamente [12]. En general, era de esperarse que manipuladores robot que tuviesen el efector final conectado a su base con múltiples

cadena actuando en paralelo lograsen mayor rigidez y superior capacidad de posicionamiento, lo cual hizo a los manipuladores paralelos atractivos para ciertas aplicaciones que tuviesen estos requerimientos de carga y precisión.

El diseño de manipuladores paralelos se remonta a varias décadas atrás por ejemplo en 1962, cuando Gough y Witehall diseñaron un sistema paralelo para ser usado en una máquina de pruebas universales. Stewart en 1965 diseñó un manipulador de plataforma para ser usado en simuladores de vuelo. Hunt en 1983 realizó un estudio de la estructura cinemática de los manipuladores paralelos, y desde entonces se han venido desarrollando numerosos estudios por diversos investigadores.

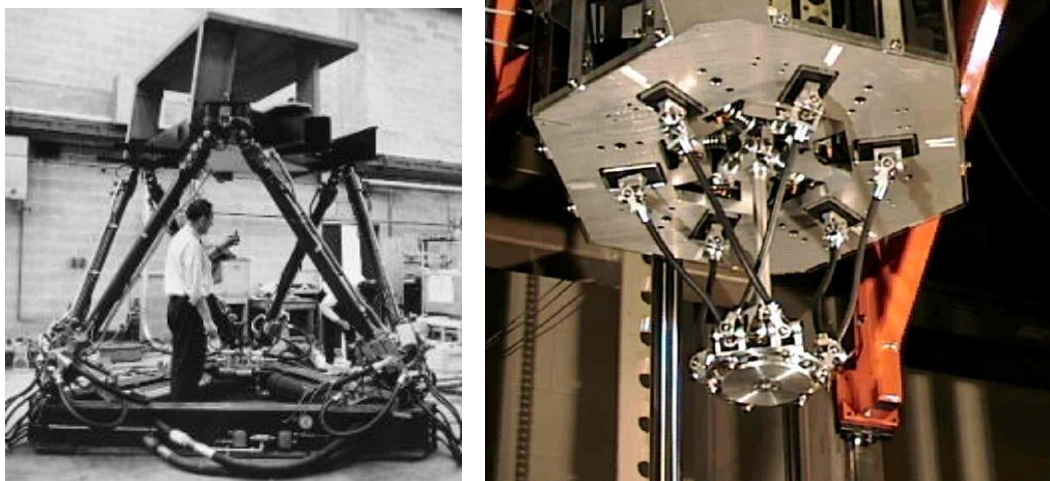


Figura 22. Aplicaciones de la plataforma de Stewart: Simulador de vuelo (izq.), robot ensamblador (der)

La figura 22 muestra una estructura de seis grados de libertad para movimiento de un simulador de vuelo, mientras que en la misma se aprecia también un robot paralelo del mismo número de grados de libertad que es empleado para el ensamble de piezas.

Cabe señalar que existen otras configuraciones como la delta, con tres grados de libertad inventada por Clavel en 1988 y es empleada en robots como el mostrado en la figura 23.



Figura 23. Robots paralelos de cuatro grados de libertad, basados en la plataforma Delta.

Los robots paralelos se han venido empleando para distintas tareas como en simuladores de vuelo, máquinas caminadoras, dispositivos de máquinas-herramientas, micro manipulación a alta frecuencia (telescopios) y recientemente para tareas de ensamble.

Una limitante importante que poseen estos tipos de manipuladores es su reducido espacio de trabajo y es precisamente ésta una de las tendencias en las investigaciones futuras de estos robots.

2.2 Descripción de la plataforma

En la figura 24 se muestra un diseño típico de una plataforma Stewart tomado de Mathworks®, en la cual podemos diferenciar sus partes:

- Plato inferior, fijo o base. Es aquel que soportará el peso de la estructura y que a su vez está anclado a tierra o a la superficie de referencia.
- Plato superior o móvil, de menor tamaño que el plato base y al cual se acopla el cuerpo al que se desea dotar de movimiento, en este caso una cabina de simulador.
- Piernas, que conectan ambos platos. Su longitud es variada con actuadores lineales los cuales son controlados simultáneamente (manipulador paralelo). Estas piernas están compuestas de una parte inferior y una superior, entre las cuales se produce una junta prismática de un grado de libertad (desplazamiento longitudinal).

El acople a los platos inferior y superior se produce con juntas del tipo universal o cardánico de dos grados de libertad (rotacionales).

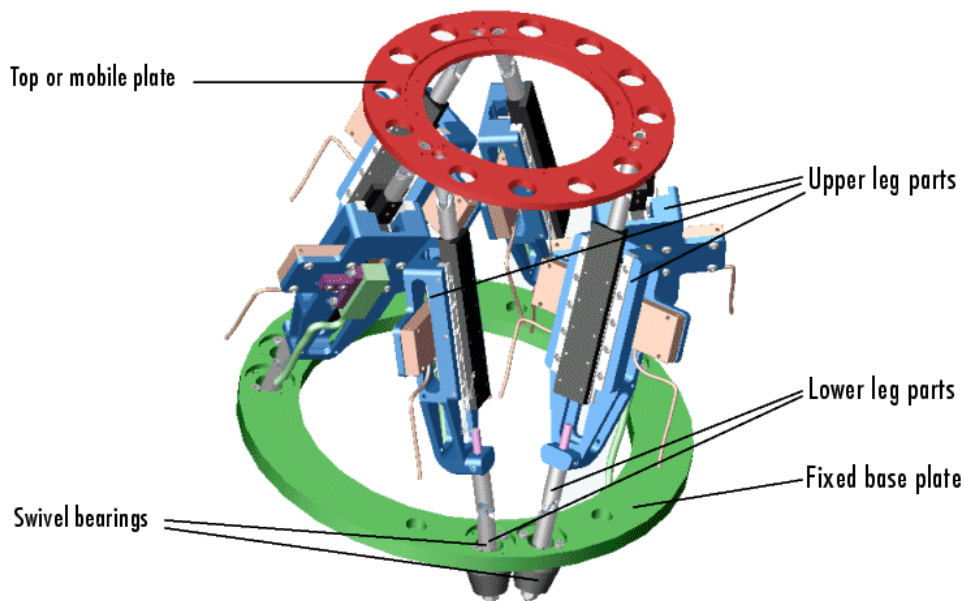


Figura 24. Modelo CAD de la plataforma Stewart

2.3 Análisis de grados de libertad

Haciendo uso del criterio de Kutzbach-Grubler [13] para sistemas tridimensionales, se efectúa el cálculo según la fórmula:

$$gdl = \lambda(l-j-1) + \sum_{i=1}^j g_i \quad (2.1)$$

Donde:

λ es el factor de movilidad (3 para cuerpos en el plano y 6 para cuerpos en el espacio)

l es el número de cuerpos rígidos (2 platos y 12 por las piernas consideradas por partes)

j es el número de uniones o juntas (6 esféricas, 6 universales y 6 prismáticas)

g_i es el número de grados de libertad en la junta i

Reemplazando se obtiene:

$$gdl = 6 (12+2 - 18 - 1) + 6(2) + 6(1) + 6(3)= 6$$

2.4 Cinemática inversa

Para resolver el problema de cinemática inversa, el vector de posición y la matriz de rotación del plato superior (U) con respecto al plato base (L) están dados y se debe encontrar la longitud de las piernas.

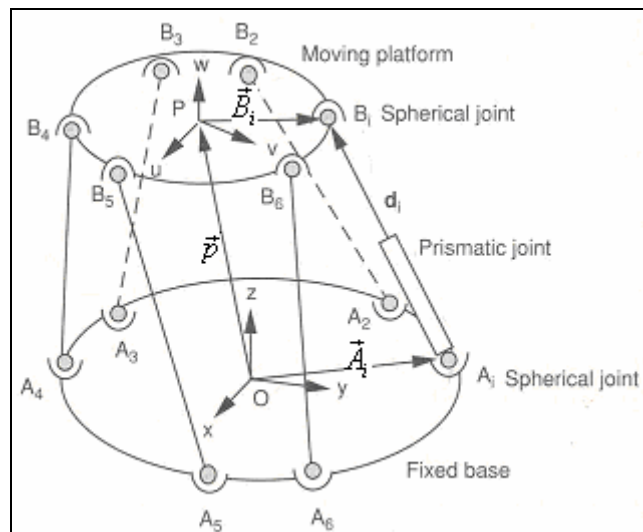


Figura 25. Esquema de la plataforma

Para analizar la cinemática inversa de la plataforma, se necesita tener dos sistemas de coordenadas cartesianas. Como se muestra en la figura 25, la base tiene el sistema de coordenadas x_w, y_w, z_w que es también el sistema de referencia, y el plato móvil tiene el sistema u_p, v_p, w_p . Para definir el lugar del plato superior, sus coordenadas deben ser transformadas al sistema de referencia, por lo tanto, el vector de posición p del centroide P y la matriz de rotación ${}^L R_U$ se usan para describir la posición y orientación. La matriz de transformación se convierte en:

$${}^L R_U = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \quad (2.2)$$

De acuerdo a la figura 20, $\vec{A}_i = [A_{ix} \ A_{iy} \ A_{iz}]^T$ y ${}^U \vec{B}_i = [B_{iu} \ B_{iv} \ B_{iw}]^T$ se eligen como vectores de posición, y el cálculo del vector está dado por:

$$AB_i = p^L R_U^U B_i - \bar{A}_i \quad (2.3)$$

Tomando el producto punto de dos vectores, la longitud de las piernas puede calcularse por:

$$d_i^2 = \left[p^L R_U^U B_i - \bar{A}_i \right]^T \left[p^L R_U^U B_i - \bar{A}_i \right] \quad (2.4)$$

Usando rotaciones con respecto a los ejes fijos, la matriz de rotación se construye como:

$$R = R_z(\Theta) R_y(\Phi) R_x(\Psi) \quad (2.5)$$

$$R = \begin{bmatrix} \cos\Theta \cos\Phi \cos\Psi - \sin\Theta \cos\Phi \sin\Psi + \cos\Theta \sin\Phi \cos\Psi & \sin\Theta \cos\Phi \cos\Psi + \cos\Theta \sin\Phi \cos\Psi & \sin\Theta \sin\Phi \cos\Psi \\ \cos\Theta \cos\Phi \sin\Psi + \sin\Theta \cos\Phi \cos\Psi & -\cos\Theta \cos\Phi \sin\Psi + \sin\Theta \cos\Phi \cos\Psi & \sin\Theta \sin\Phi \sin\Psi \\ -\sin\Theta & \cos\Theta & \sin\Theta \cos\Psi \end{bmatrix}$$

2.5 Ecuaciones dinámicas de la Plataforma Stewart

A fin de obtener un modelo dinámico, el sistema es separado en dos partes: la parte móvil (plato superior) y los seis actuadores con el plato base- Las fuerzas f_p en las juntas del plato superior se derivan usando la formulación de Lagrange.

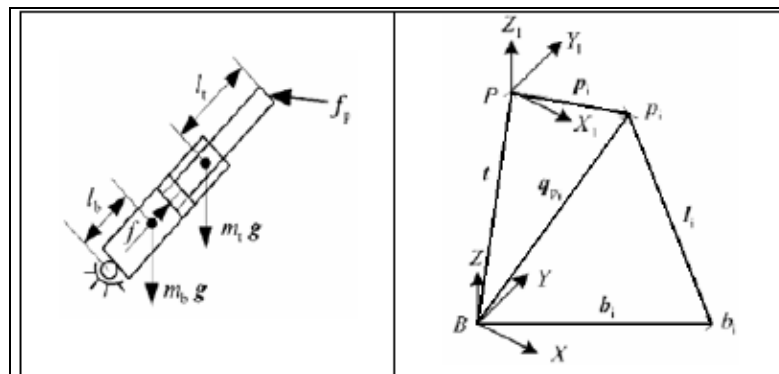


Figura 26. Izq. Diagrama de fuerzas. Der. Análisis de posición de una pierna

Considerando q_p y \dot{q}_p como las correspondientes coordenadas generalizadas y velocidades con la energía cinética, $T(q_p, \dot{q}_p)$, las ecuaciones de Lagrange del sistema están dadas por:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_p} \right) - \frac{\partial T}{\partial q_p} = Q \quad (2.6)$$

donde Q denota las fuerzas generalizadas proyectadas a lo largo de la variación de las coordenadas generalizadas q_p .

La energía cinética puede darse por:

$$\begin{aligned} T &= \frac{1}{2} \dot{v}_t^T m \dot{v}_t + \frac{1}{2} \omega_t^T (I_t + I_b) \omega_t \\ &= \frac{1}{2} \dot{q}_p^T \left(I + \frac{l_t \tilde{n}^2}{l} \right)^T m \left(I + \frac{l_t \tilde{n}^2}{l} \right) + (I_t + I_b) \tilde{n}^T \frac{\tilde{n}}{l^2} \dot{q}_p \\ &= \frac{1}{2} \dot{q}_p^T (M_1 + M_2) \dot{q}_p \end{aligned} \quad (2.7)$$

Donde se emplean las ecuaciones:

$$\omega_t = \frac{n \times \dot{q}_p}{l} = \frac{\tilde{n} \dot{q}_p}{l} \quad \text{velocidad angular del actuador (2.8)}$$

$$v_t = \dot{q}_p + \omega_t \times (-l_t n) = \left(I + \frac{l_t \tilde{n}^2}{l} \right) \dot{q}_p \quad \text{velocidad del centro de gravedad para las partes móviles del actuador (2.9)}$$

$$M_1 = \left(I + \frac{l_t \tilde{n}^2}{l} \right)^T m \left(I + \frac{l_t \tilde{n}^2}{l} \right) \quad (2.10)$$

$$M_2 = \frac{(I_t + I_b) \tilde{n}^T \tilde{n}}{l^2} \quad (2.11)$$

Considerando las ecuaciones 2.6 y 2.7, se derivan las siguientes relaciones:

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_p} \right) = \frac{d}{dt} ((M_1 + M_2) \dot{q}_p) = \frac{d(M_1 + M_2)}{dt} \dot{q}_p + (M_1 + M_2) \ddot{q}_p \quad (2.11a)$$

$$\begin{aligned} \frac{dM_1}{dt} &= \frac{2ml_t}{l^2} (n \dot{q}_p^T \tilde{n}^T \tilde{n} + n^T \dot{q}_p \tilde{n}^T \tilde{n} + \tilde{n}^T \tilde{n} \dot{q}_p n^T) \\ &\quad - \frac{ml_t^2}{l^3} (2n^T \dot{q}_p \tilde{n}^T \tilde{n} + n \dot{q}_p^T \tilde{n}^T \tilde{n} + \tilde{n}^T \tilde{n} \dot{q}_p n^T) \end{aligned} \quad (2.11b)$$

$$\frac{dM_2}{dt} = -\frac{I_t + I_b}{l^3} (2\dot{q}_p^T n \tilde{n}^T \tilde{n} + \tilde{n}^T \tilde{n} \dot{q}_p n^T + n \dot{q}_p^T \tilde{n}^T \tilde{n}) \quad (2.11c)$$

$$\begin{aligned} \frac{\partial T}{\partial \dot{q}_p} &= \frac{\partial}{\partial \dot{q}_p} \left(\frac{1}{2} \dot{q}_p^T (M_1 + M_2) \dot{q}_p \right) \\ &= ml_t \left(-\dot{q}_p^T \dot{q}_p \frac{\partial}{\partial \dot{q}_p} \left(\frac{1}{l} \right) + \frac{\partial}{\partial \dot{q}_p} \left(\frac{\dot{q}_p^T l \dot{q}_p^T l}{l^3} \right) \right) + \frac{1}{2} (ml_t^2 + I_t + I_b) \left(\dot{q}_p^T \dot{q}_p \frac{\partial}{\partial \dot{q}_p} \left(\frac{1}{l^2} \right) \right. \\ &\quad \left. - 2 \frac{\dot{q}_p^T l}{l^2} \frac{\partial}{\partial \dot{q}_p} \left(\frac{\dot{q}_p^T l}{l^2} \right) \right) \end{aligned} \quad (2.12)$$

Para simplificar la ecuación 2.12 se usan las siguientes expresiones de derivadas parciales:

$$\begin{aligned} \frac{\partial}{\partial \dot{q}_p} \left(\frac{1}{l} \right) &= -\frac{n}{l^2} \\ \frac{\partial}{\partial \dot{q}_p} \left(\frac{1}{l^2} \right) &= -2 \frac{n}{l^3} \\ \frac{\partial}{\partial \dot{q}_p} \left(\frac{\dot{q}_p^T l}{l^2} \right) &= \frac{\dot{q}_p}{l^2} - 2 \frac{n \tilde{n}^T \dot{q}_p}{l^2} \\ \frac{\partial}{\partial \dot{q}_p} \left(\frac{\dot{q}_p^T l \dot{q}_p^T l}{l^3} \right) &= \frac{1}{l^2} (2n^T \dot{q}_p \tilde{n}^T \tilde{n} \dot{q}_p - n \dot{q}_p^T n \tilde{n}^T \dot{q}_p) \end{aligned} \quad (2.13)$$

La ecuación 2.12 puede reescribirse como:

$$\begin{aligned} \frac{\partial T}{\partial \dot{q}_p} &= \frac{ml_t}{l^2} (n \dot{q}_p^T \dot{q}_p + 2n^T \dot{q}_p \dot{q}_p - 3n \dot{q}_p^T n \tilde{n}^T \dot{q}_p) \\ &\quad - \frac{ml_t^2}{l^3} (n \dot{q}_p^T \dot{q}_p + \dot{q}_p \dot{q}_p^T n - 2n \tilde{n}^T \dot{q}_p \dot{q}_p^T n) \\ &\quad - \frac{I_t + I_b}{l^3} (n^T \dot{q}_p \dot{q}_p + n \dot{q}_p^T \dot{q}_p - 2n \dot{q}_p - 2n \dot{q}_p n \tilde{n}^T \dot{q}_p) \end{aligned} \quad (2.14)$$

Como se muestra en la figura 26, hay muchas fuerzas externas actuando en la pierna, como las fuerzas gravitacionales debidas a la masa del actuador (m_i y m_b), la fuerza f generada por el actuador y la fuerza actuante en la articulación superior que a su vez actúa sobre la pierna. Tomando como base en el principio del trabajo virtual, dos sistemas de fuerzas son equivalentes si hacen el mismo trabajo virtual en sentido dinámico. La fuerza generalizada Q_f que se proyecta en la variación de coordenadas generalizadas (q_p) puede derivarse como sigue:

$$\delta W = Q_f \delta q_p = f \delta d = f n^T \delta q_p = (n f)^T \delta q_p \quad (2.15)$$

Donde la ecuación:

$$i = n^T \dot{q}_p = [n^T \quad n^T \tilde{p}_b^T] \dot{q} = [n^T \quad (\tilde{p}_b n)^T] \dot{q} = [n^T \quad (R p \times n)^T] \dot{q} \quad (2.16)$$

Similarmente, las fuerzas generalizadas debido al efecto de la gravedad pueden ser derivadas por:

$$Q_{m_g} = (I + \frac{l_i \tilde{n}^2}{l})^T m_i g = (I + \frac{l_i \tilde{n}^2}{l}) m_i g \quad (2.17)$$

$$Q_{m_{bg}} = (\frac{l_b \tilde{n}^T \tilde{n}}{l})^T m_b g = (\frac{l_b \tilde{n}^T \tilde{n}}{l}) m_b g$$

Donde se ha aplicado la ecuación de velocidad del centro de gravedad para las partes rotatorias del actuador:

$$v_b = w_i \times l_b n = (\frac{l_b \tilde{n}^T \tilde{n}}{l}) \dot{q}_p \quad (2.18)$$

La fuerza proyectada a lo largo de la variación de coordenadas q_p puede ser derivada como:

$$Q = Q_f + Q_{m_g} + Q_{m_{bg}} + f_p \quad (2.19)$$

Usando las ecuaciones 2.6, 2.11a, 2.11b, 2.14 y 2.19, la fuerza f_p en la unión al extremo de la pierna puede representarse como:

$$f_p = (M_1 + M_2) \ddot{q}_p + C_a \dot{q}_p - (Q_f + Q_{m_g} + Q_{m_{bg}}) \quad (2.20)$$

donde:

$$C_a = \frac{m l_t}{l^2} (\dot{n} \dot{q}_p \tilde{n}^T \tilde{n} + n^T \dot{q}_p \tilde{n}^T \tilde{n} + \tilde{n}^T \dot{n} \dot{q}_p n^T) - \frac{m l_t^2}{l^3} (n^T \dot{q}_p \tilde{n}^T \tilde{n} + \tilde{n}^T \dot{n} \dot{q}_p n^T) - \frac{2(I_t + I_b)}{l^3} (\tilde{n}^T \dot{n} \dot{q}_p n^T) \quad (2.21)$$

La ecuación 2.20 se reescribe como:

$$f_p = (M_1 + M_2) [I \quad R \tilde{P}^T R^T] \ddot{q} + C_a [I \quad R \tilde{P}^T R^T] \dot{q} + (M_1 + M_2) \tilde{w}^2 R p - (Q_f + Q_{ng} + Q_{mg}) \quad (2.22)$$

Donde se han usado las ecuaciones

$$\begin{aligned} \dot{q}_p &= \dot{i} + w \times R p = \dot{i} + (\tilde{p}_b)^T w = \dot{i} + R \tilde{P}^T R^T w \\ &= [I \quad R \tilde{P}^T R^T] \begin{bmatrix} \dot{i} \\ w \end{bmatrix} = [I \quad R \tilde{P}^T R^T] \dot{q} \end{aligned}$$

velocidad del punto superior de la junta q_p (2.23)

$$\ddot{q}_p = [I \quad R \tilde{P}^T R^T] \ddot{q} + \tilde{w}^2 R p$$

aceleración del punto superior de la junta q_p (2.24)

Capítulo 3

Simulación del vuelo

3.1 Herramientas de software empleadas

Para la simulación del vuelo se han empleado las siguientes herramientas de software:

3.1.1 FlightGear

Es un proyecto libre de simulación de vuelo. Está siendo desarrollado gracias a las contribuciones de código fuente y tiempo libre de programadores de todo el mundo.

La idea de FlightGear nació de la insatisfacción con los simuladores de vuelo comerciales actuales existentes para PC. Un gran problema con estos simuladores es el hecho de ser propietarios y su falta de extensibilidad. Mucha gente dedicada a la enseñanza y la investigación podrían usar un entorno de simulación de vuelo sobre el que construir sus propios proyectos; sin embargo, los simuladores comerciales no les permiten la modificación y mejora [14].

Hay una gran cantidad de gente interesada y participando en este proyecto. Es verdaderamente un esfuerzo global de contribuyentes de todos los continentes. Los intereses abarcan desde construir un simulador realista, la investigación universitaria y el uso instructivo, hasta simplemente una alternativa viable a los simuladores comerciales para PC.

Características:

Alto grado de Libertad

FlightGear es un proyecto de código abierto. Esto significa que mientras se cumplan los términos de la licencia GPL, se puede descargar y copiar FlightGear libremente. Hay acceso libre al último código fuente del desarrollo. Dado que es un proyecto de código abierto, los formatos de archivos son abiertos y fácilmente accesibles. Se utilizan formatos de modelado 3D estándar y gran parte de la

configuración del simulador está controlada mediante ficheros ASCII basados en XML. Crear extensiones adicionales para FlightGear (o directamente modificar el código fuente) es directo y requiere aplicar gran cantidad de ingeniería inversa. Esto hace a FlightGear una opción atractiva para su uso en proyectos privados, comerciales, de investigación o de hobby.

FlightGear funciona en plataformas Windows, Linux, Mac OS-X, FreeBSD, Solaris e IRIX, permitiendo al usuario utilizarlo en su plataforma preferida.

Modelos de Dinámica de Vuelo

Con FlightGear es posible elegir entre tres Modelos de Dinámica de Vuelo primarios. Es posible añadir nuevos modelos de dinámica o incluso conectar con modelos de dinámica de vuelo "propietarios" externos:

1. **JSBSim**: es un modelo de dinámicas de vuelo genérico y 6DoF para la simulación del movimiento de vehículos aéreos. Está escrito en C++. JSBSim se puede ejecutar en modo independiente para ejecuciones en batch, o puede hacer de controlador para un programa de simulación mayor que incluya subsistemas visuales (como FlightGear). En ambos casos, los aviones se modelan en un archivo de configuración XML, donde se definen las propiedades de masa, aerodinámica y control de vuelo.
2. **YASim**: Es parte integrante de FlightGear y usa una forma distinta a JSBSim simulando el efecto de la circulación del aire en distintas partes del avión. La ventaja de este modelo es que es posible realizar la simulación basada en la información de geometría y masa combinada con datos de funcionamiento disponibles más comunes para un avión. Esto permite construir rápidamente un avión con un comportamiento que coincida con los datos de funcionamiento publicados sin necesidad de realizar todo el test de datos tradicional.
3. **UIUC**: Este MDV está basado en LaRCsim escrito originalmente por la NASA. UIUC aumenta el código permitiendo archivos de configuración del avión en lugar de añadir código para la simulación del avión bajo condiciones de hielo.

Base de Datos del Escenario Mundial

- Alrededor de 20.000 aeropuertos reales incluidos en el set de escenario completo.
- Pistas, señalizaciones, emplazamientos y luces de aproximación correctas.
- Pistas de rodaje disponibles para muchos aeropuertos grandes (incluso se incluyen luces de la línea verde central cuando es necesario).
- Pistas inclinadas (las pistas cambian su inclinación tal y como ocurre en la vida real).
- Iluminación direccional de aeropuerto que cambian de intensidad según cambia la dirección de vista relativa.
- Terreno preciso de todo el mundo, basado en la publicación más reciente de los datos de terreno SRTM. 3 arco segundo de resolución (sobre 90m post espacio) para América del Norte y del Sur, Europa, Asia, África y Australia.
- El escenario incluye todos los lagos, ríos, carreteras, ferrocarriles, ciudades, pueblos, terrenos, etc.
- Buena iluminación del escenario nocturno con iluminación concentrada en la áreas urbanas (basado en mapas reales) y luces visibles en las principales autopistas. Esto permite vuelos nocturnos realistas con la posibilidad de reconocer pueblos y ciudades y seguir carreteras.
- Las porciones del escenario se paginan (cargan/descargan) en un hilo independiente para minimizar la tasa de fotogramas cuando se necesita cargar nuevas áreas.

Modelo del Cielo

FlightGear implementa un modelado horario extremadamente preciso con localizaciones correctas del sol, la luna, las estrellas y los planetas para la fecha y hora especificadas. FlightGear puede leer la hora actual del reloj de sistema para colocar correctamente el sol, la luna, las estrellas, etc. en su lugar actual correcto relativo a la tierra. El sol, la luna, las estrellas y los planetas siguen sus órbitas correctas en el cielo. Este modelado también tiene en cuenta los efectos estacionales, como días de 24 horas al norte del Círculo Polar ártico en verano, etc. Destaca también la correcta posición de la luna junto con la correcta posición del sol para conseguir la fase lunar correcta para la fecha/hora actual, como en la vida real.

Sistema de Modelado de Aviones

FlightGear brinda la posibilidad de modelar una amplia variedad de aviones. Actualmente se puede volar con el 1903 Wright Flyer, extraños "ornitópteros" aleteantes, un Boeing 747 y un Airbus A320, distintos jets militares y varios ligeros. FlightGear permite modelar esos aviones y casi cualquier cosa intermedia.

FlightGear posee una infraestructura que permite a los diseñadores de aviones construir cabinas totalmente animadas, totalmente operativas y totalmente interactivas (que se refrescan y muestran correctamente respecto a las vistas externas del avión). Modela de una forma realista el comportamiento de los instrumentos del mundo real. Los instrumentos que tiene demora en la vida real, también se demoran correctamente en FlightGear, el movimiento del giroscopio está implementado correctamente, la brújula está sujeta a las fuerzas del cuerpo del avión.

FlightGear reproduce de forma precisa los fallos de muchos sistemas e instrumentos. Si el sistema de presión falla, los giroscopios HSI giran hacia abajo lentamente con la correspondiente pérdida de respuesta.

Requerimientos de Hardware

El resultado es un simulador con un requerimiento de hardware moderado para funcionar con una tasa de fotogramas suave. Puede funcionar razonablemente bien en un computador de entre 500 y 1000 dólares, mientras se tengan buenos recursos para gráficos y RAM.

Propiedades Internas expuestas

FlightGear permite a los usuarios y diseñadores de aviones acceder a un gran número de variables de estado internas a través de numerosos mecanismos de acceso tanto internos como externos. Estas variables de estado están organizadas en un árbol de propiedades jerárquico.

Utilizando el árbol de propiedades es posible inspeccionar cualquier variable de estado interna de FlightGear. Es posible controlar de forma remota a FlightGear

desde un script externo. Se pueden crear animaciones de modelo, efectos sonoros, animaciones de instrumentos y protocolos de red para cualquier situación imaginable con tan sólo editar un pequeño número de ficheros de configuración. Este sistema tan potente hace que FlightGear sea inmensamente flexible, configurable y adaptable.

Opciones de Red

Varias opciones de red permiten a FlightGear comunicarse con otros programas FlightGear, con receptores GPS, con módulos externos de dinámicas de vuelo, pilotos automáticos externos o módulos de control externo, así como otro software como el proyecto Open Glass Cockpit y la utilidad de mapas Atlas.

Una opción de entrada/salida genérica permite un protocolo definido por el usuario para un fichero, puerto serie o cliente de red. FlightGear dispone de un protocolo multiusuario para red local en un entorno multiavión, por ejemplo para practicar vuelos en formación o para simulaciones de torre.

Las potentes opciones de red hacen posible sincronizar varias instancias de FlightGear permitiendo presentaciones multipantalla, si todas las instancias se están ejecutando con la misma tasa de fotogramas, es posible conseguir sincronizaciones realmente buenas entre las pantallas.

3.1.2 Simulink Aerospace Toolbox

Extiende el ambiente de desarrollo de Matlab proporcionando estándares de referencia, modelos de ambiente e importación de coeficientes aerodinámicos para realizar análisis aeroespaciales avanzados y evaluación de diseños [15]. Entre sus principales ventajas se encuentran:

- Proporciona modelos ambientales basados en estándares para atmósfera, gravedad y campos magnéticos.
- Convierte unidades y transforma sistemas de coordenadas y representación espacial.

- Implementa utilidades predefinidas para cálculo de parámetros aeroespaciales.
- A través de la interfase de FlightGear se permite visualizar la dinámica del vehículo en ambiente tridimensional.

3.1.3 Simulink Aerosim Toolbox

Proporciona un completo set de herramientas para el desarrollo rápido de modelos dinámicos de aeronaves en seis grados de libertad. Adicionalmente a los bloques básicos de dinámica de aeronaves, también están incluidos modelos completos que pueden ser adecuados al requerimiento del usuario de acuerdo a archivos de configuración [16].

3.2 Diagrama de Bloques en Simulink para la simulación del vuelo

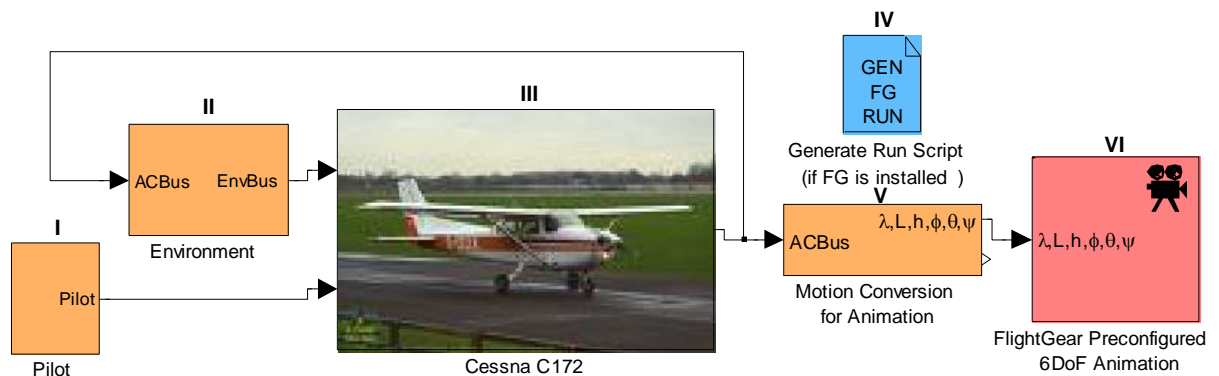


Figura 27. Esquema de simulación del vuelo con Simulink

Descripción de los bloques

3.2.1 Pilot

Interfase entre el operador y el software, a través de un joystick el usuario envía señales de comando a la aeronave.

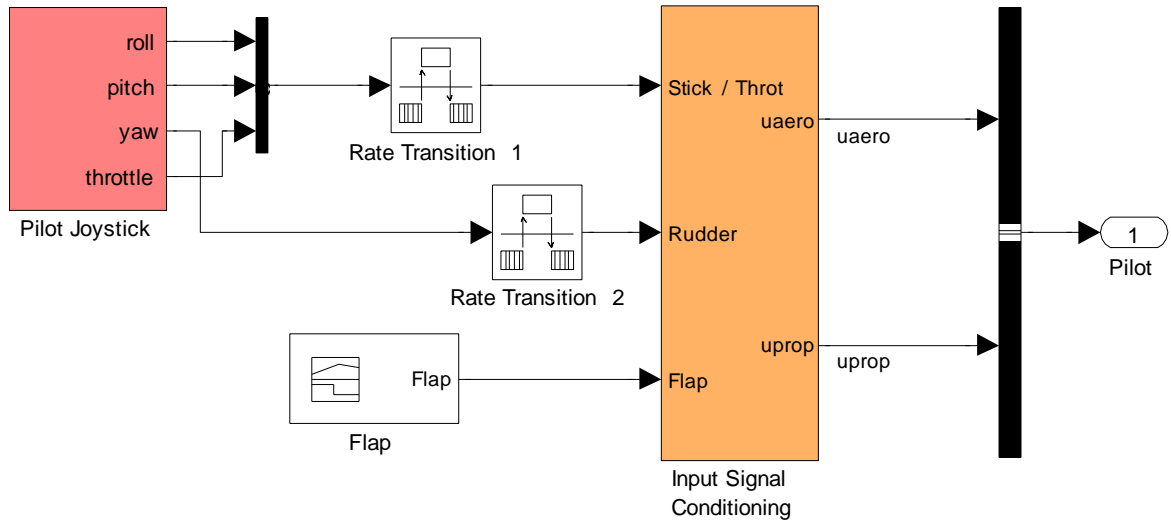


Figura 28. Detalle interno del bloque pilot

En la figura 29 se puede apreciar el contenido del bloque “*Input Signal Conditioning*”, que, como su nombre indica, se encarga de realizar el acondicionamiento de las señales de entrada recibidas del joystick a través de uno de los puertos USB. Para las señales de comando de movimiento del avión se consideran entradas analógicas en stick y throttle, mientras que el movimiento de rudders y flap debe ser generado por el usuario mediante teclado, ya que el sistema de mandos no los contempla.

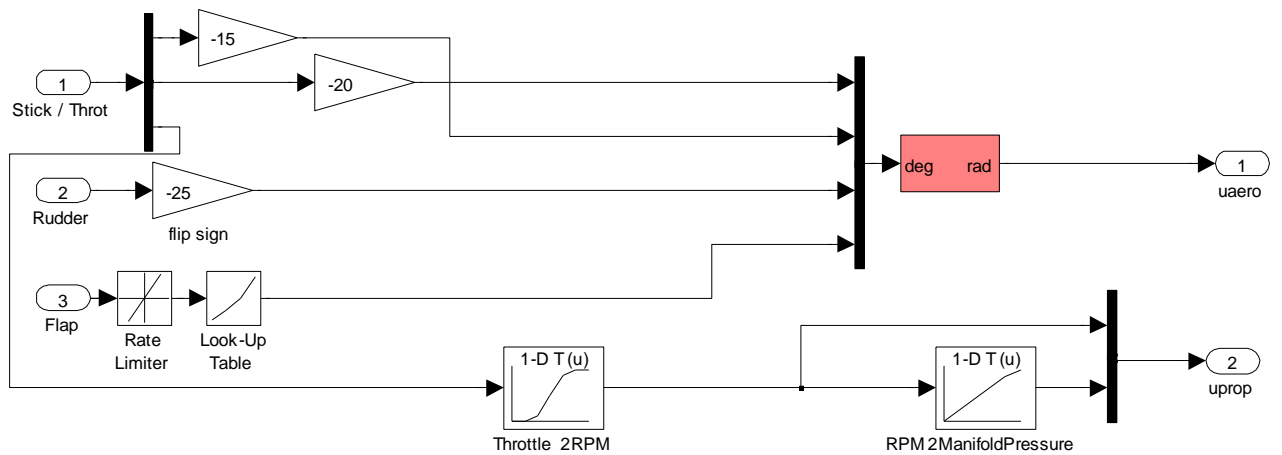


Figura 29. Detalle interno del bloque input signal conditioning

3.2.2 Environment

Se encarga de generar las condiciones climáticas en el entorno simulado, para lo cual considera la gravedad terrestre en el punto, el modelo de atmósfera correspondiente a la altitud del vuelo y los modelos de viento, todo esto propios del escenario seleccionado por el usuario y la configuración de fecha y hora cargadas al inicio de la simulación.

En la figura 30 se puede apreciar la composición interna del bloque, mientras que a continuación se hace una breve descripción de sus principales componentes

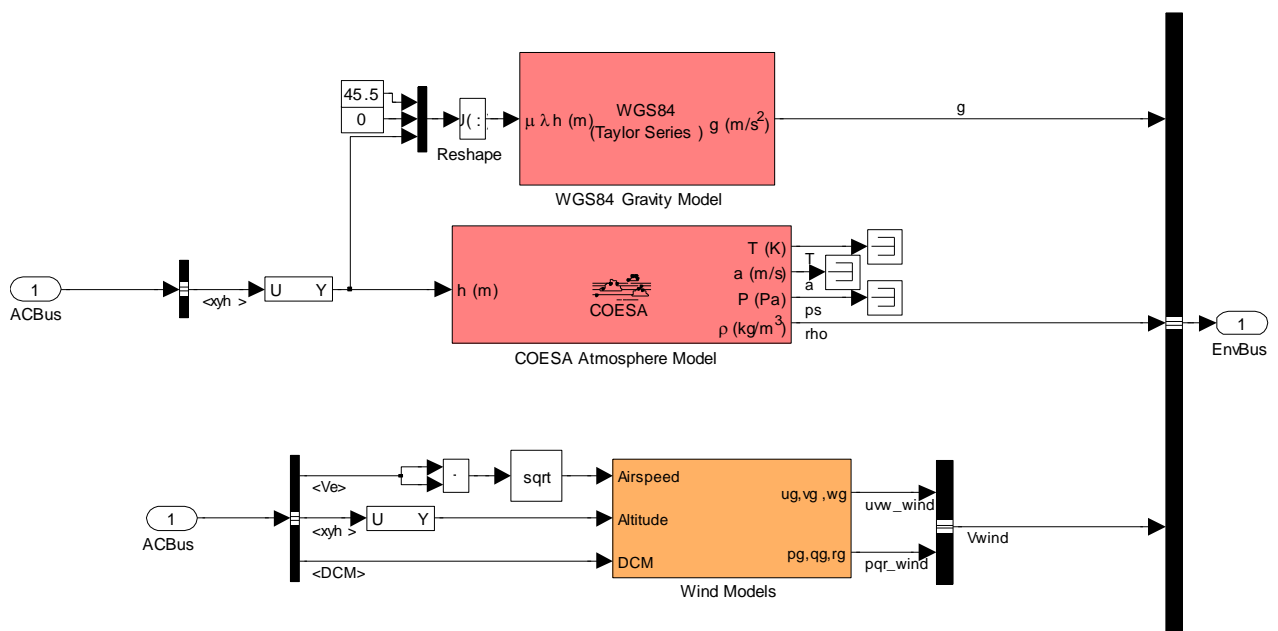


Figura 30. Detalle interno del bloque Environment

WGS84 Gravity Model

La función de este bloque es calcular el valor de la gravedad terrestre en una ubicación específica empleando el modelo de referencia World Geodetic System (WGS84). Este modelo considera a la tierra como un elipsoide, requiere como dato de entrada la ubicación en latitud, longitud y altitud.

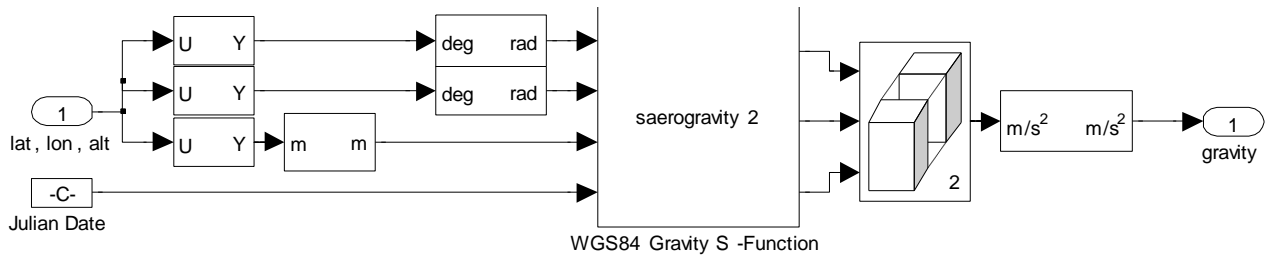


Figura 31. Detalle interno del bloque WGS84 Gravity Model

COESA Atmosphere Model

Es la implementación matemática de la atmósfera estándar con menores valores para temperatura, presión, densidad y velocidad de sonido absolutos de acuerdo a un valor de entrada de altitud. Fue establecida por el Comité de Extensión de la Atmósfera Estándar (COESA) de los Estados Unidos en el año 1976. Bajo los 32,000 metros la atmósfera COESA es idéntica a la atmósfera estándar empleada por la Organización de Aviación Civil Internacional (OACI).

En el caso de la simulación empleada, el único valor que se toma en cuenta de las salidas es el correspondiente a densidad del aire, mientras que los restantes, si bien son generados, no son considerados en los cálculos posteriores.

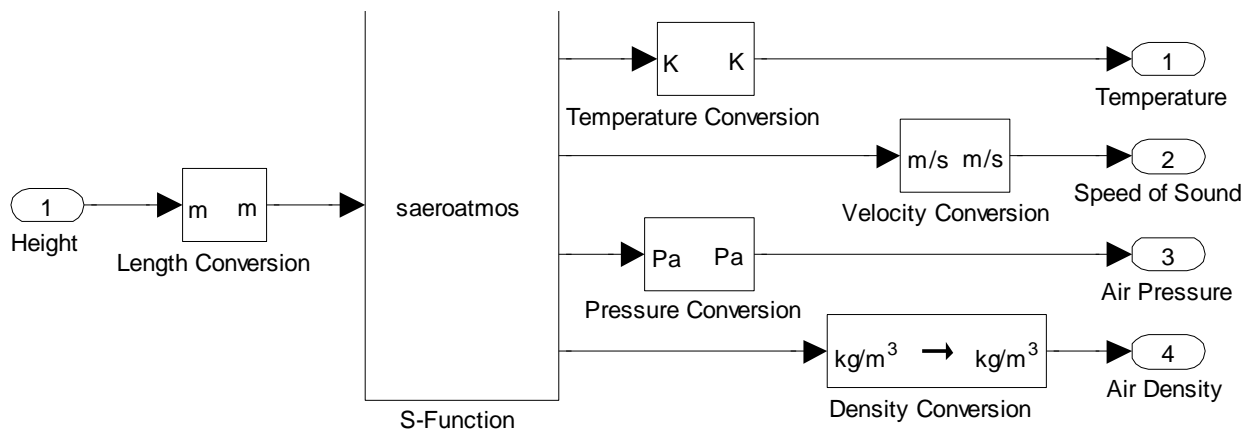


Figura 32. Detalle interno del bloque COESA Atmosphere Model

Wind Models

Se encarga de generar las corrientes de viento correspondientes a las condiciones de vuelo a las que se ha configurado la simulación, las cuales a su vez obedecen a modelos matemáticos. En este bloque se consideran:

- Modelos de viento wind shear (o viento cruzado), corresponde a un tipo de corriente de alta velocidad que se presenta a bajas alturas, y que representa una de las permanentes amenazas para el vuelo sobre todo al momento del aterrizaje. Tiene como entrada la altitud y la matriz de transformación para ejes centrados en el cuerpo respecto a ejes centrados en la tierra; y como salida la velocidad del viento wind shear.
- Modelo de turbulencia de viento de Dryden, que se encarga de generar turbulencias atmosféricas basado en el empleo de ruido blanco como generador de señales aleatorias. Tiene como entradas la altitud y velocidad del avión, y la matriz de transformación para ejes centrados en el cuerpo respecto a ejes centrados en la tierra. Tiene como salida la velocidad en m/s de la turbulencia y las velocidades angulares en rad/s.
- Modelo de generación de ráfagas de viento discretas, que se encarga de generar estos fenómenos teniendo como entrada la velocidad de la aeronave y como salida la velocidad de la ráfaga.

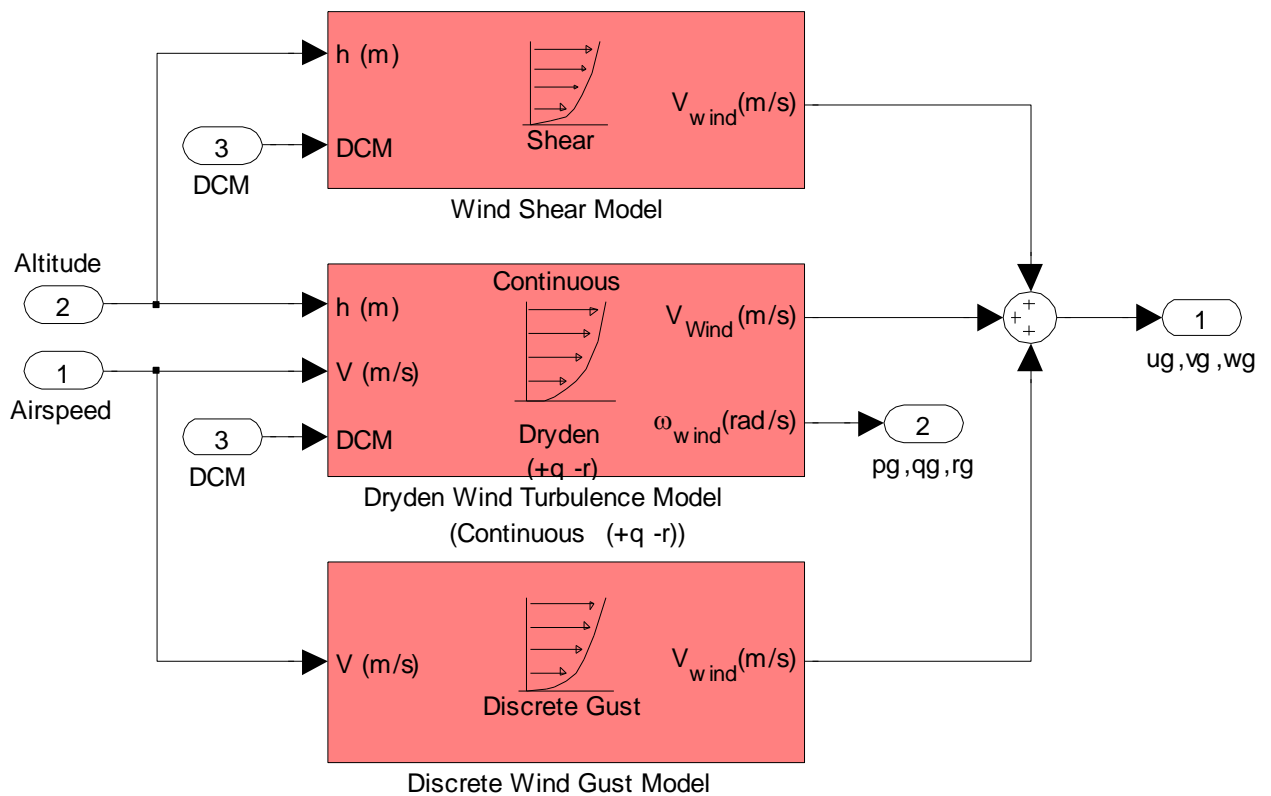


Figura 33. Detalle interno del bloque Wind Models

3.2.3 Cessna 172

Este bloque representa la dinámica de la aeronave, que está influenciada por su diseño estructural, tamaño, material de construcción, peso, características del motor, etc. Recibe como entradas las señales de comando que han sido acondicionadas desde su ingreso por el usuario a través del joystick, y las características del ambiente.

Sus componentes principales son: el bloque de cálculo de parámetros y el bloque de dinámica de la aeronave propiamente dicho.

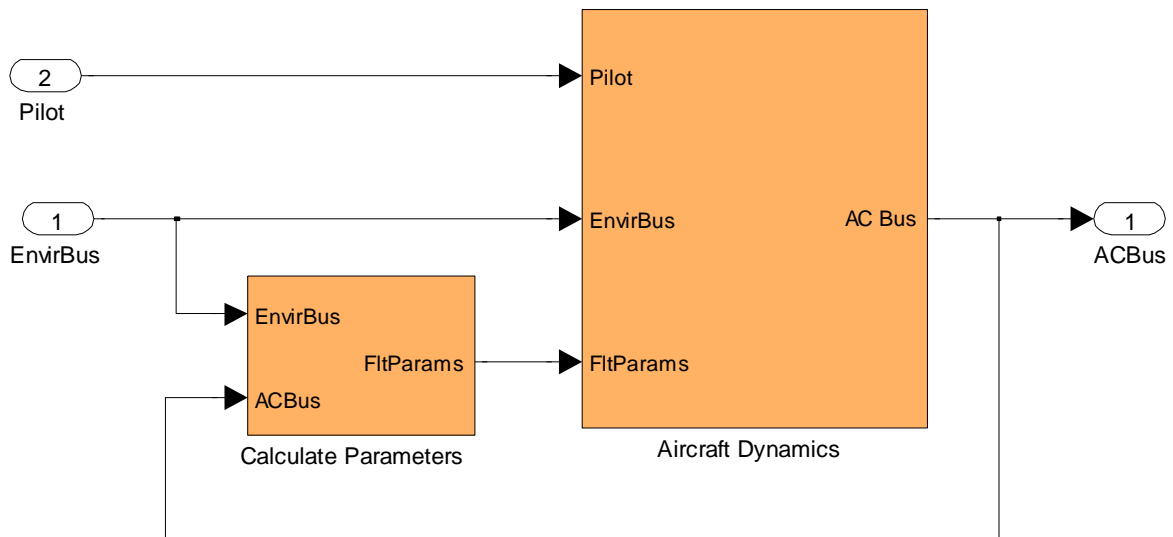


Figura 34. Detalle interno del bloque Cessna C172

Calculate parameters

Calcula los ángulos entre el cuerpo y el vector velocidad (ángulo de incidencia y de derrape lateral), así como la presión dinámica sobre el avión. Se refiere básicamente a las características del aire como fluido rodeando a la aeronave y actuando sobre ella.

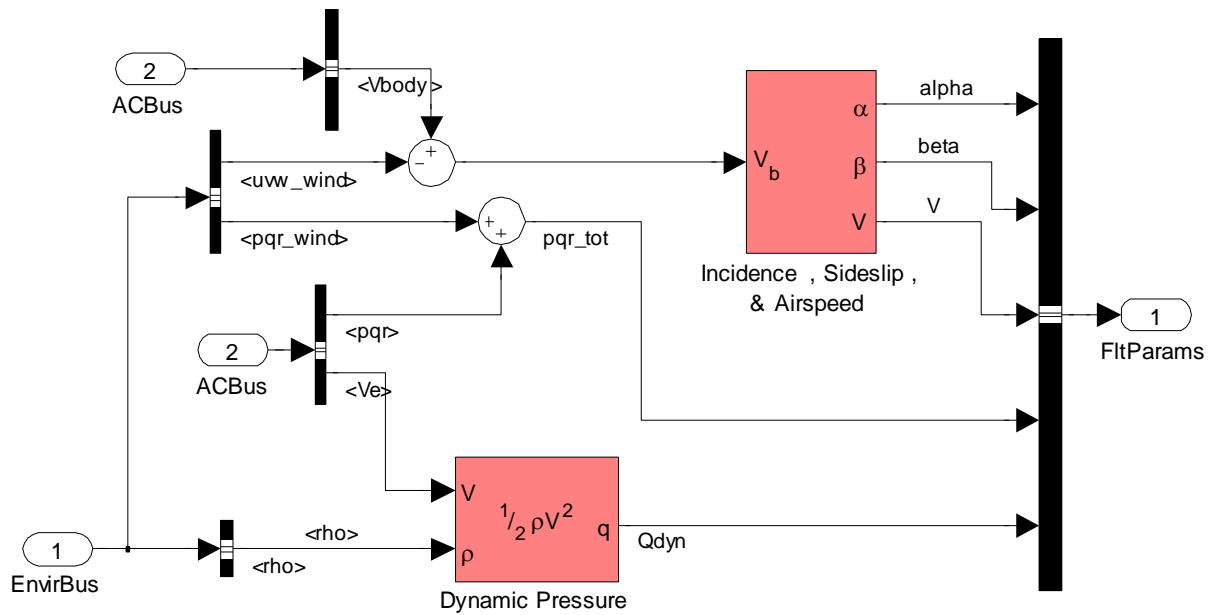


Figura 35. Detalle interno del bloque calculate parameters

Aircraft dynamics

Se encarga de calcular las fuerzas y momentos aerodinámicos usando coeficientes, datos de presión dinámica, centro de gravedad, centro de presiones y velocidad. Adicionalmente se encarga de implementar la representación del avión en seis grados de libertad con respecto a los ejes de movimiento del cuerpo.

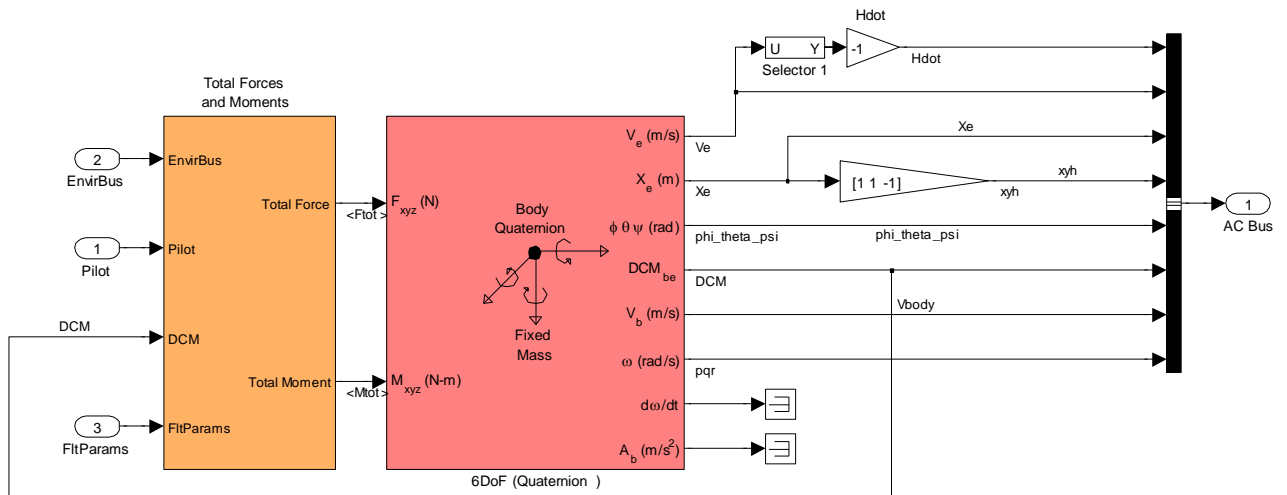


Figura 36. Detalle interno del bloque aircraft dynamics

3.2.4 Generate run script

Este bloque configura la simulación que se va a lanzar visualmente en el FlightGear desde Simulink. Genera un archivo de tipo script en el que se incluye la

ubicación del directorio base de FlightGear, el modelo geométrico de la aeronave a simular, puerto de comunicaciones a emplear, aeródromo de partida, pista de despegue, altitud, orientación geográfica, etc.

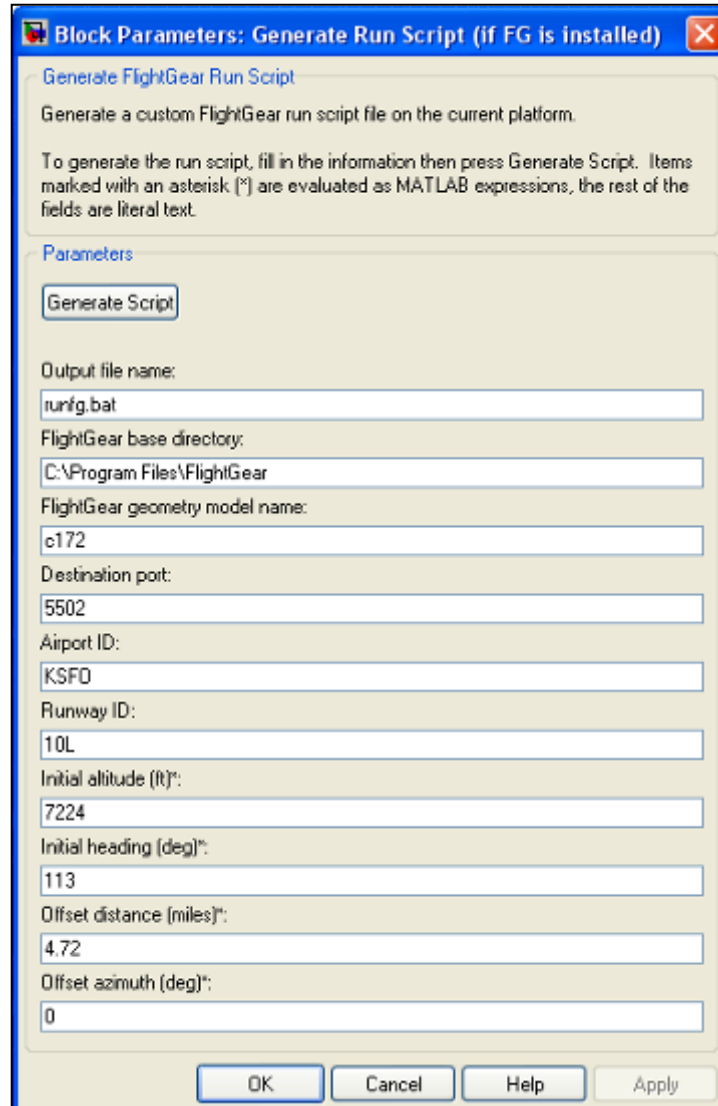


Figura 37. Ventana de configuración del bloque configurate run script

```

C:\WINDOWS\system32\cmd.exe - runfguncomputer
C:\Documents and Settings\GUILLERMO\My Documents\MATLAB>C:
C:\Documents and Settings\GUILLERMO\My Documents\MATLAB>cd C:\Program Files\FlightGear
C:\Program Files\FlightGear>SETI FG_ROOT=C:\Program Files\FlightGear\data
C:\Program Files\FlightGear>.\bin\win32\fgfs --aircraft=c172 --fdm=network,localhost,5501,5502,5503 --fog-fastest --disable-clouds --start-date-lat-2004:06:01:09:00:00 --disable-sound --in-air --enable-freeze --airport-id=KSFO --runway=10L --altitude=7224 --heading=113 --offset-distance=4.72 --offset-azimuth=0

```

Figura 38. Script generado para la ejecución de FlightGear

3.2.5 Motion conversion for animation

Se encarga de hacer la transformación de la información calculada para la aeronave a simular de manera que pueda ser visualizada de acuerdo a los parámetros de realidad virtual de FlightGear. Recibe información sobre la aeronave y realiza la conversión de un sistema considerando la tierra plana a un sistema de latitud, longitud y altitud, así como los ángulos de orientación en el espacio, enviando toda esta información a la interfase específica.

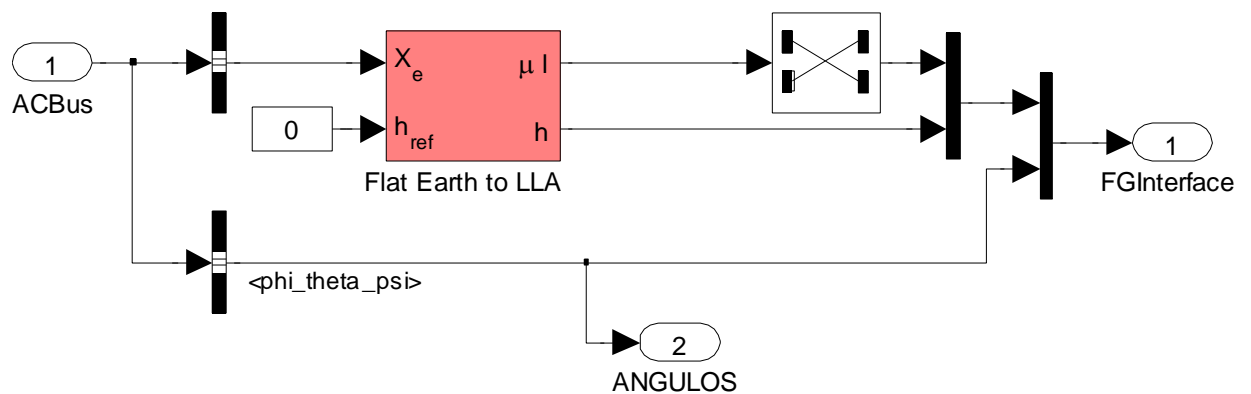


Figura 39. Detalle interno del bloque motion conversion for animation

Flat Earth to LLA

Realiza la conversión de las coordenadas correspondientes al modelo plano de la tierra al formato de latitud, longitud y altitud considerando el efecto de la curvatura del planeta.

Como entradas emplea las siguientes:

- X_e , vector conteniendo la posición en el marco del modelo plano
- ground height, valor de altitud de referencia

Como salidas se obtienen:

- μl , vector conteniendo la latitud y longitud en grados
- h , valor escalar de la altitud sobre el nivel de referencia

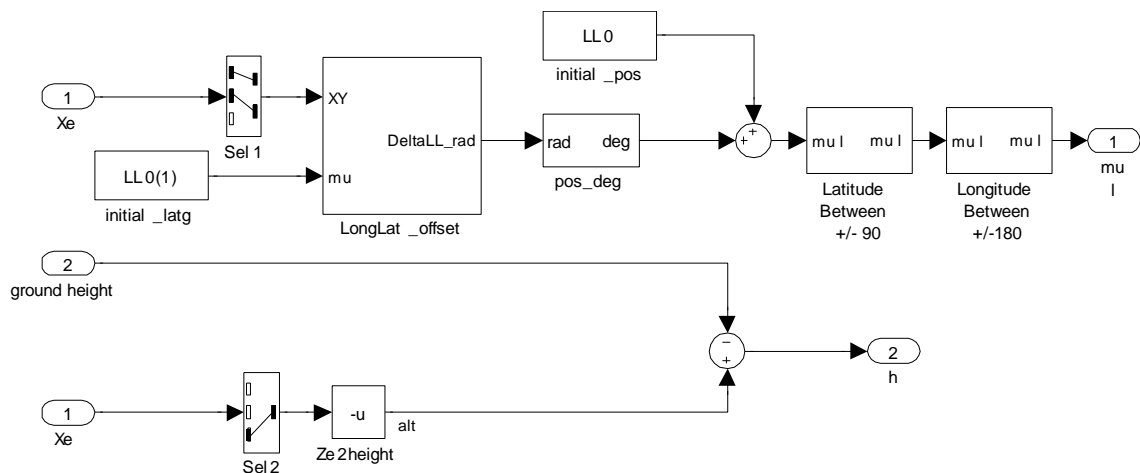


Figura 40. Detalle interno del bloque flat earth to lla

3.2.6 FlightGear preconfigured 6DoF Animation

Conecta el modelo al software a través de las características previamente establecidas. Permite llevar valores de posición y actitud de la aeronave a través de un flujo de valores numéricos de latitud, longitud, elevación, alabeo, cabeceo y guiñada. El bloque enmascara los componentes necesarios para la transmisión de datos a otro computador o su ejecución en el mismo, estableciendo las tasas de refresco de datos para una presentación visual coherente.

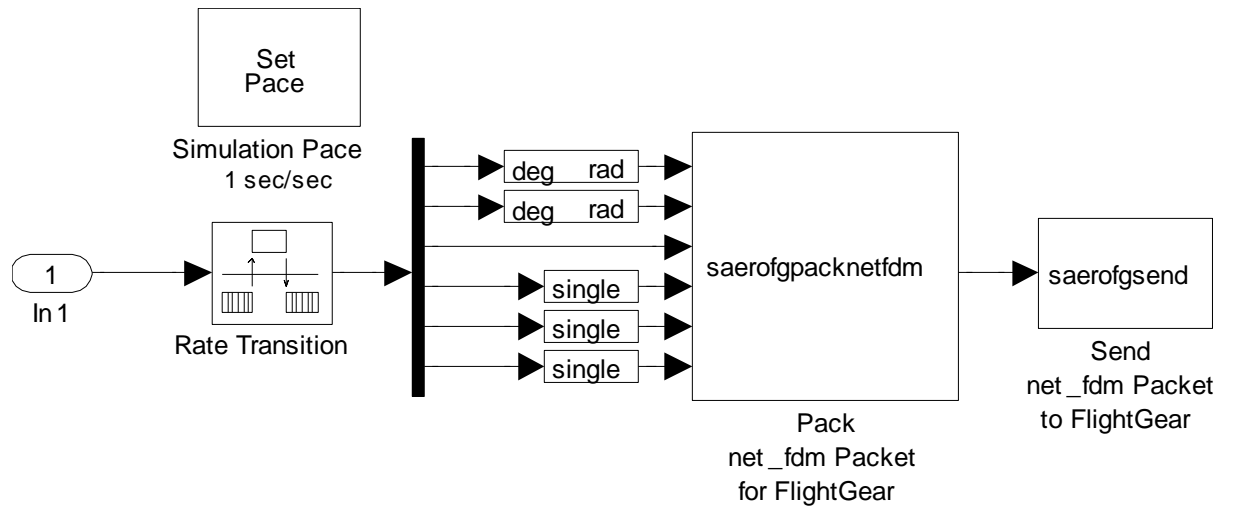


Figura 41. Detalle interno del bloque FlightGear preconfigured 6 DoF animation

Capítulo 4

Simulación de la Plataforma Stewart con Matlab/Simulink

4.1 Herramientas de software empleadas

4.1.1 Simulink SimMechanics Toolbox

Es una parte del modelamiento físico de Simulink. Está orientado al diseño en ingeniería y la simulación de sistemas mecánicos de cuerpos rígidos conectados por articulaciones, con la dinámica newtoniana estándar de fuerzas y torques.

SimMechanics simula movimientos rotacionales y traslacionales en tres dimensiones, proporcionando una suite de herramientas para cuerpos específicos y sus propiedades de masa, sus posibles movimientos, restricciones cinemáticas, sistemas de coordenadas y la forma de generar y medir movimientos. Permite representar un sistema mecánico a través de un diagrama de bloques interconectados como otros modelos de Simulink, y puede incluir subsistemas jerárquicos [17]. Las herramientas de visualización y animación permiten representaciones simplificadas del movimiento de los sistemas simulados.

4.2 Simulación

Se ha empleado el modelo de plataforma con trayectorias de referencia predefinidas disponible como caso de estudio por Mathworks, el cual ha sido modificado para recibir señales de entrada provenientes de la simulación de vuelo en lugar de las que el sistema generaba originalmente basado en formas de onda senoidales. Este se descompone en los siguientes bloques:

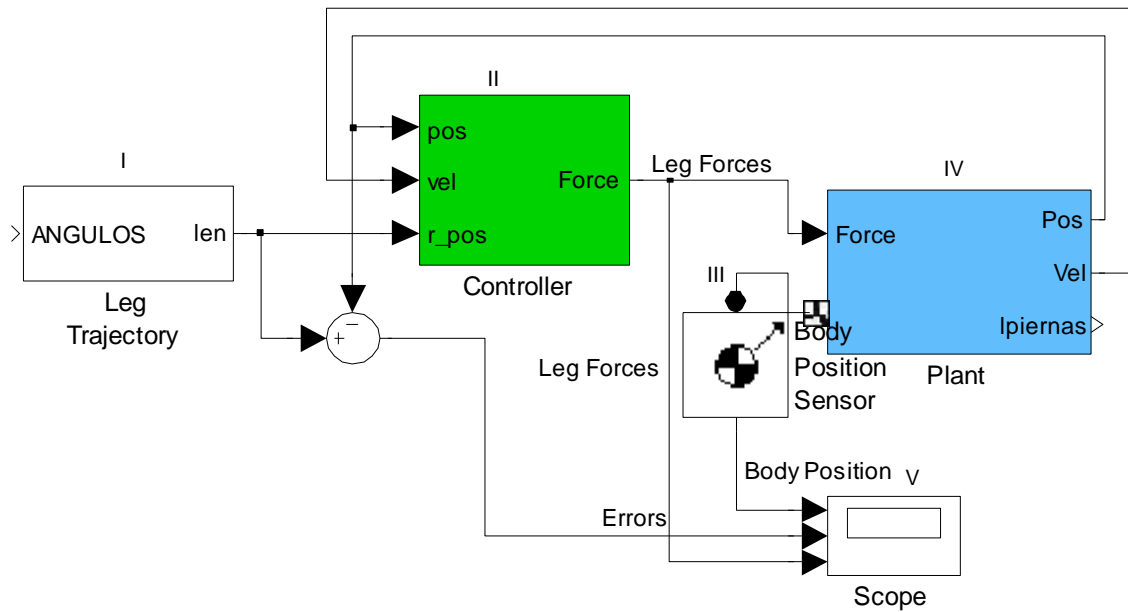


Figura 42. Esquema de simulación de la plataforma Stewart con Simulink

Descripción de los bloques

4.2.1 Bloque leg trajectory

Se encarga de generar la trayectoria a seguir por el conjunto móvil. El contenido del subsistema se puede apreciar en la Figura 43:

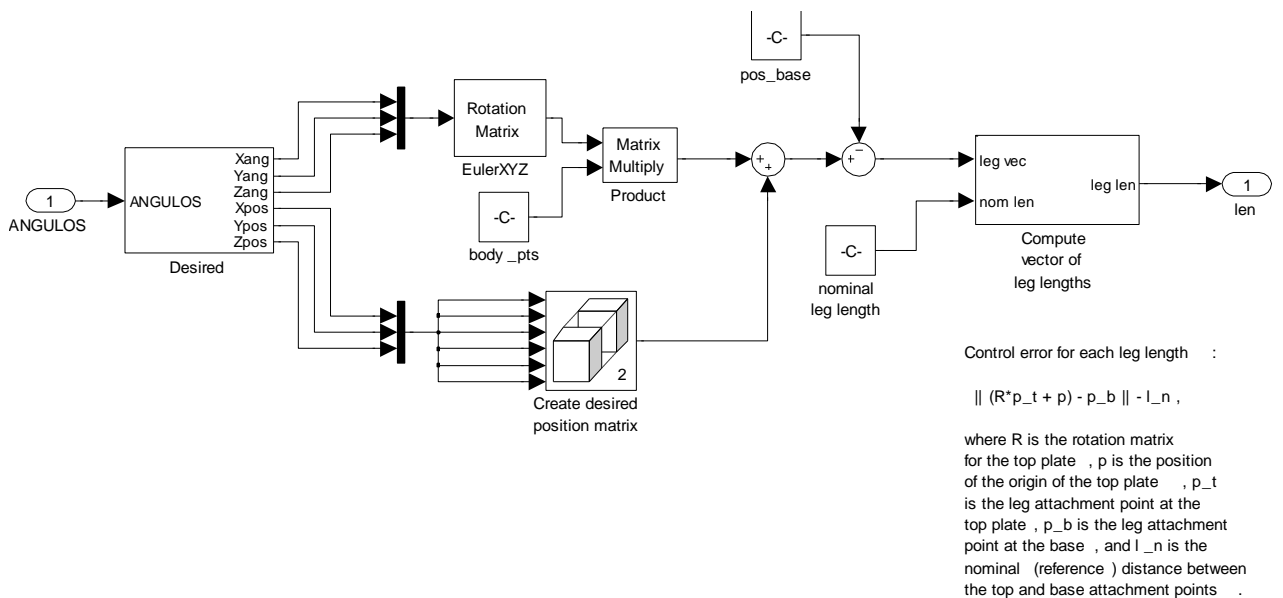


Figura 43. Detalle interno del bloque leg trajectory

Como entrada se recibe la información de los ángulos de orientación en el espacio o ángulos de Euler, que se obtienen en la simulación del vuelo. Estos son descompuestos en sus componentes en los tres ejes, y se generan datos tanto angulares como de posición. Posteriormente, a través de cálculo con matrices de rotación se arriba a las posiciones finales deseadas para el conjunto de la plataforma, cambiando los centros de referencia. Esto se diferencia de la posición base y se produce el cálculo de la longitud de cada pierna que permitirá llegar a la posición deseada, finalmente esta longitud se entrega como salida del bloque.

4.2.2 Bloque controller

El modelo de plataforma usa un controlador simple del tipo PID y un bloque sensor de posición para medir el movimiento. La implementación del control de trayectoria se realiza aplicando fuerzas a la planta proporcionales al error del movimiento.

Una ley de control PID es una combinación de la variable detectada por el sensor, su integral en el tiempo y su primera derivada [18]. Esta plataforma usa los errores de posición de cada pierna E_r y sus integrales y velocidades. La ley de control para cada pierna r tiene la forma:

$$F_{act,r} = K_p E_r + K_i \int_0^t E_r dt + K_d \left(\frac{dE_r}{dt} \right) \quad (4.1)$$

El controlador aplica la fuerza actuante $F_{act,r}$ a través de la pierna:

Si E_r es positiva, la pierna es muy corta, y $F_{act,r}$ es positiva (extensión).

Si E_r es negativa, la pierna es muy larga, y $F_{act,r}$ es negativa (retracción).

Si E_r es cero, la pierna tiene exactamente la longitud deseada, y $F_{act,r}$ es cero.

Las constantes K_p , K_i , y K_d son, respectivamente, las ganancias proporcional, integral y derivativa que actúan sobre las señales realimentadas por el sensor en la ley de control:

El primer término es proporcional al error de posición instantáneo de la pierna, o desviación de la referencia.

El segundo término es proporcional a la integral del error de posición de la pierna

El tercer término es proporcional a la derivada del error de posición de la pierna, en este caso vista como velocidad.

El resultado es $F_{act,r}$, la fuerza del actuador aplicada por el controlador a las piernas. Los términos proporcional, integral y derivativo tienden a hacer que los puntos de unión superiores de las piernas $p_{t,r}$ sigan las trayectorias de referencia suprimiendo el error de movimiento.

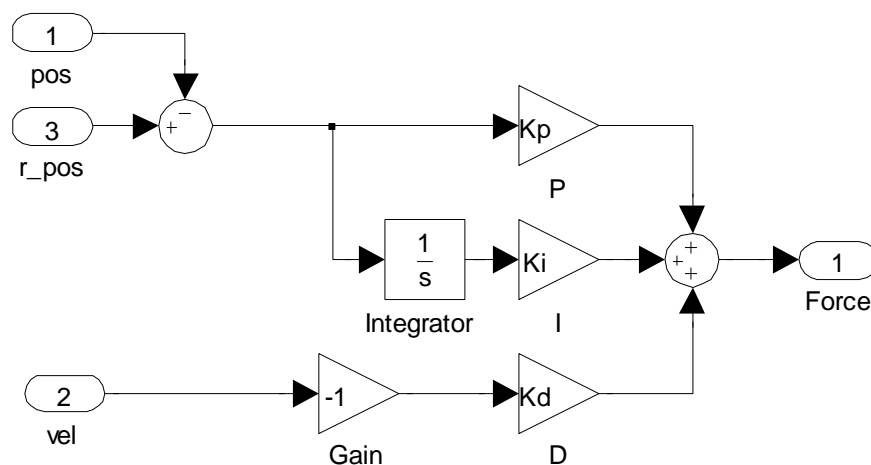


Figura 8. Detalle interno del bloque controller

Los valores por defecto configurados en el controlador para las constantes K_i , K_p , K_d son:

$$K_i = 10000$$

$$K_p = 2000000$$

$$K_d = 45000$$

Se trata del método de control más simple empleado en el modelo inicial. Existen otros algoritmos más sofisticados que son en su mayoría variaciones de la teoría del uso de filtros del tipo washout [19].

4.2.3 Bloque body position sensor

El control PID requiere las derivadas con respecto al tiempo del movimiento actual y el referencial. Para brindar una buena calidad de realismo, la planta de la plataforma usa un bloque sensor para detectar solo la actual posición de la plataforma, dejando que la velocidad sea calculada por el controlador. Encontrar la velocidad actual y de la referencia requiere tomar derivadas numéricas de las trayectorias actuales, cada una de las cuales es una función del tiempo.

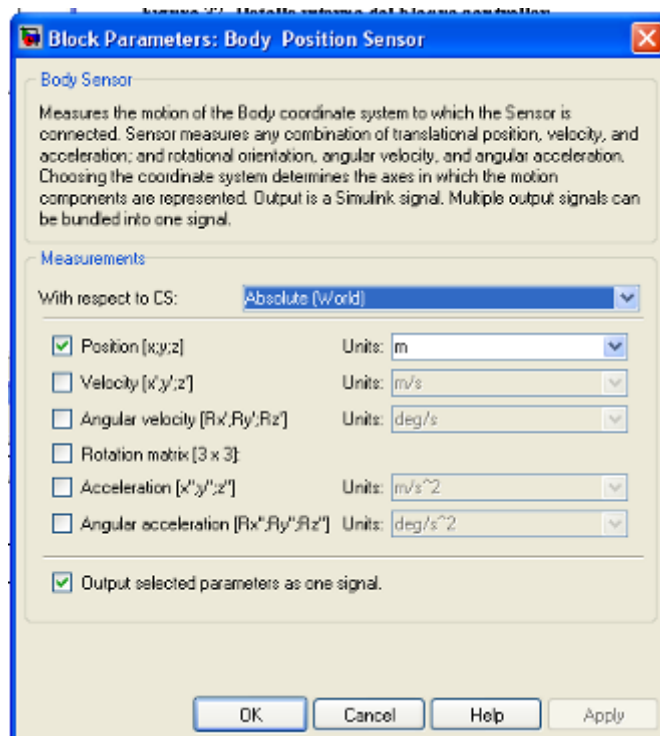


Figura 45. Ventana de configuración del bloque body position sensor

4.2.4 Plant

Consiste en el modelamiento de planta de la plataforma Stewart, en este caso empleando herramientas de software en vez de los tradicionales procedimientos matemáticos que usan ecuaciones diferenciales. Está compuesta por los platos superior e inferior, unidos por las piernas, como puede apreciarse en la figura 47.

En el caso del plato superior, que representa finalmente el cuerpo que se va a mover, se cuenta con una ventana de configuración para sus sistemas de

coordenadas y centro de gravedad (figura 46) las cuales arrastran todo el proceso posterior y posibilitan el cálculo de la dinámica, tarea que anteriormente hubiese demandado gran capacidad de recursos de cálculo y que ahora se ha simplificado significativamente.

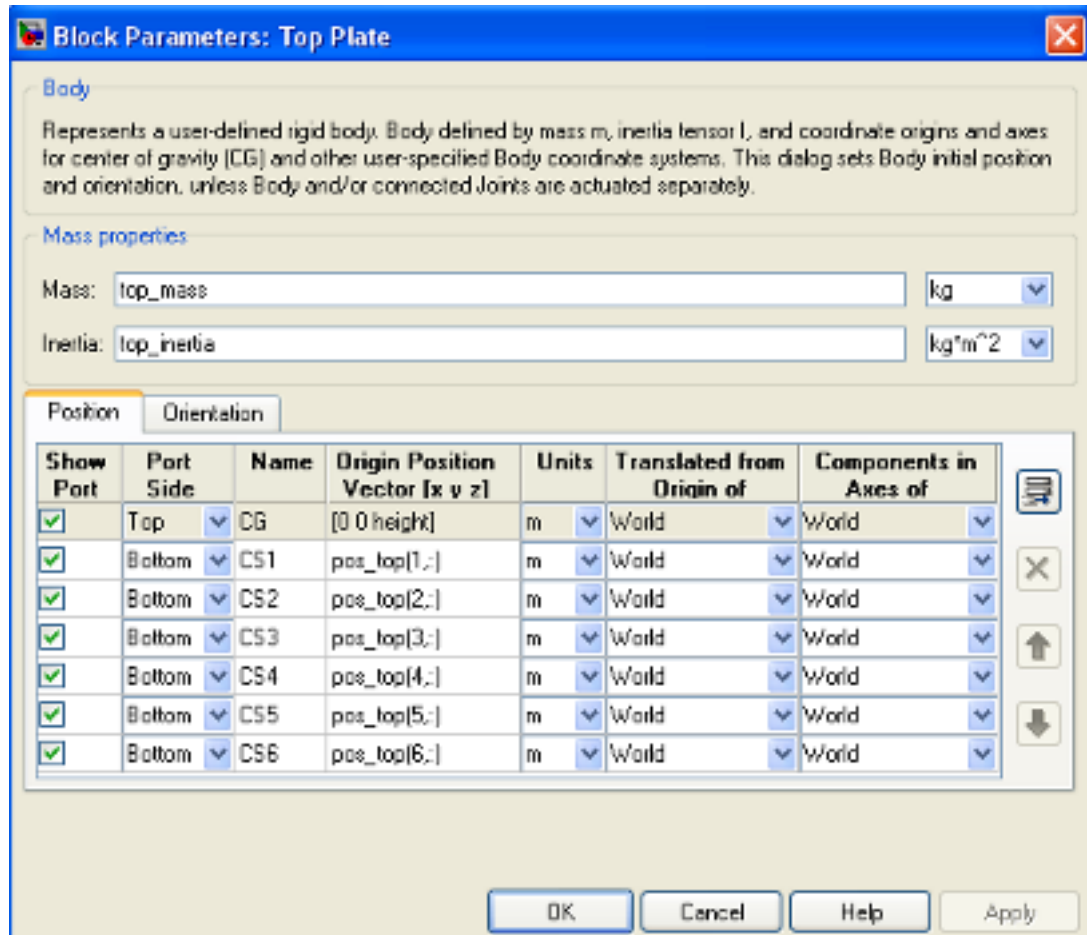


Figura 46. Ventana de configuración del plato superior

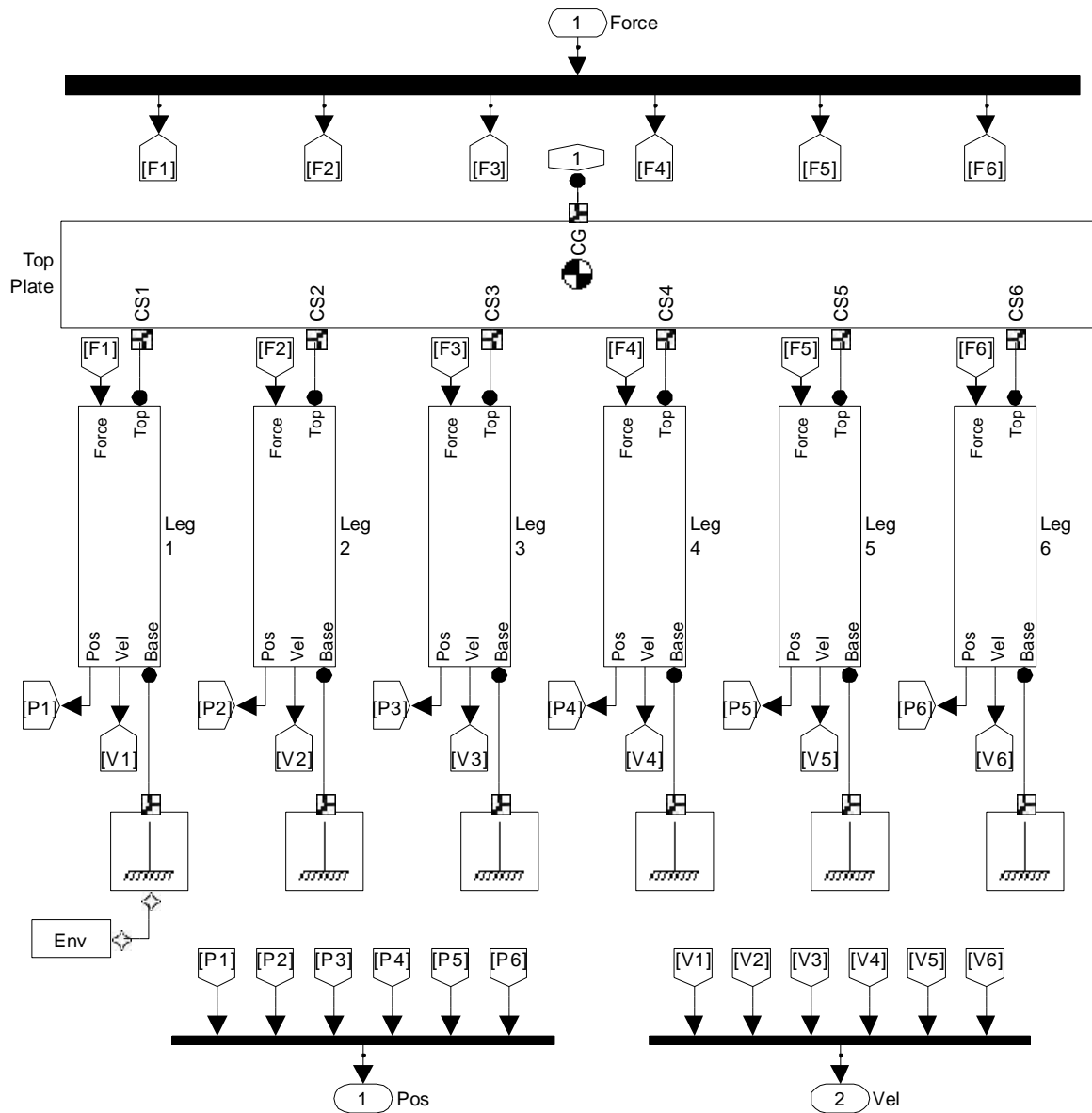


Figura 47. Detalle interno del bloque plant

Descomponiendo cada pierna se obtiene el siguiente esquema, dentro del cual se aprecian las características de las juntas empleadas: universal entre las piernas y los platos superior e inferior, y cilíndrica entre el actuador y el resto del conjunto. En este ultimo caso, si bien el esquema corresponde a una articulación cilíndrica, en la realidad se trata de desplazamiento únicamente longitudinal sin ningún tipo de rotación.

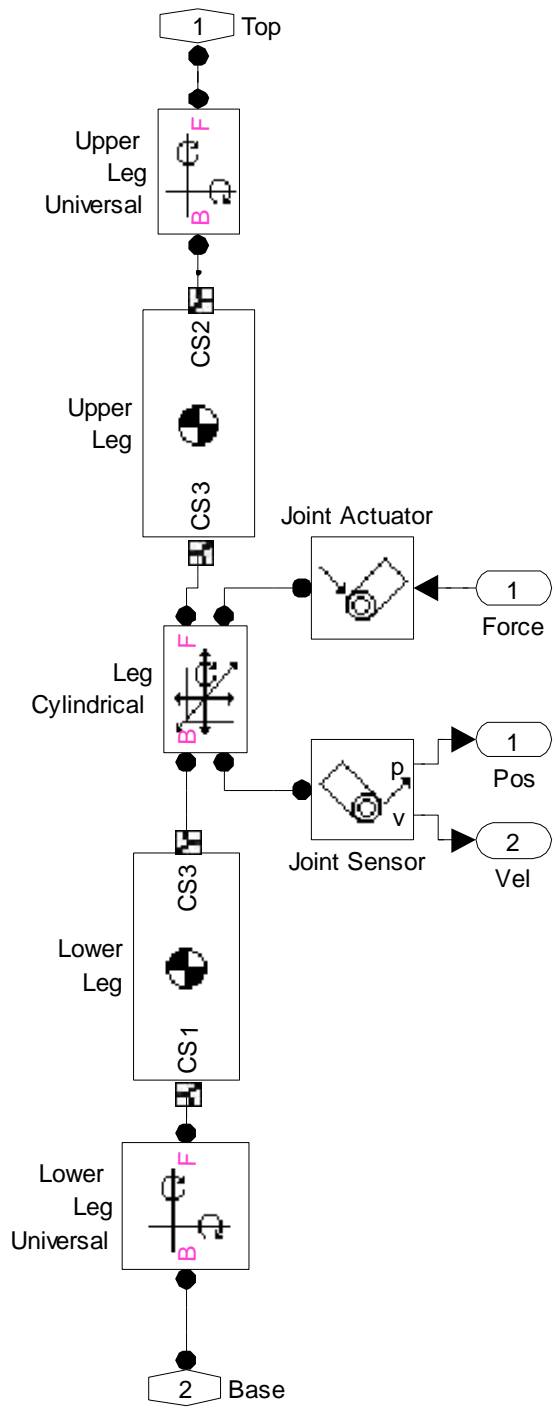


Figura 48. Detalle interno del subsistema pierna

Asimismo, el modelo considera un sensor de posición y velocidad para el sistema, en el caso de la aplicación desarrollada se ha restringido el monitoreo a solamente posición

4.3 Definición del problema de control

Se ha tomado como caso de referencia el análisis realizado por Mathworks para la plataforma Stewart, donde se define el problema como *“encontrar un método para accionar las seis piernas y llevar a una posición y orientación específica el plato móvil con respecto a la base fija”*. Estos valores deseados deben poder variar en el tiempo, siendo el modelo de planta no lineal. Para la estrategia de control se debe considerar el accionamiento de las piernas y el sensado de longitud y velocidad.

4.4 Modelamiento de componentes físicos con SimMechanics

Se ha construido un modelo de planta que consiste en el plato superior móvil, el plato inferior fijo y seis piernas que conectan ambos platos. El sistema final tiene seis grados de libertad, cada subsistema “pierna” contiene dos cuerpos conectados con una junta cilíndrica, mientras que los extremos de las piernas se conectan a los platos con juntas de tipo universal.

El accionamiento lineal de la plataforma se obtiene variando la longitud de las piernas, para lo cual se emplea un bloque actuador cuyo fin es controlar el grado de libertad trasnacional de la junta cilíndrica (el grado rotacional se considera libre).

Para la orientación espacial se emplea la teoría de ángulos de Euler. No se presentan cálculos específicos por cuanto el sistema permanentemente varía su posición, corrigiendo el resultado final según la formula:

$$\|(R^*p_{i,i} + p) - p_{b,i}\| - l_{n,i} \quad (4.2)$$

Donde:

\mathbf{R} es la matriz de rotación entre la orientación del plato superior respecto al inferior

$\mathbf{p}_{t,i}$ es el punto de unión de la pierna i en el plato superior respecto a su sistema de coordenadas.

\mathbf{p} , posición del plato superior respecto al inferior

$\mathbf{p}_{b,i}$ es el punto de unión de la pierna i en el plato inferior respecto a su sistema de coordenadas.

$l_{n,i}$ es la longitud nominal de la pierna i .

Capítulo 5

Implementación del modelo a escala

5.1 Modelo Simulink

Para la realización del modelo a escala se empleó el siguiente esquema en Simulink, que consiste en integrar los módulos previamente descritos a través de la toma de información de la animación del vuelo simulado hacia la simulación de la plataforma Stewart.

BANCO DE PRUEBA VIRTUAL PARA PLATAFORMA STEWART

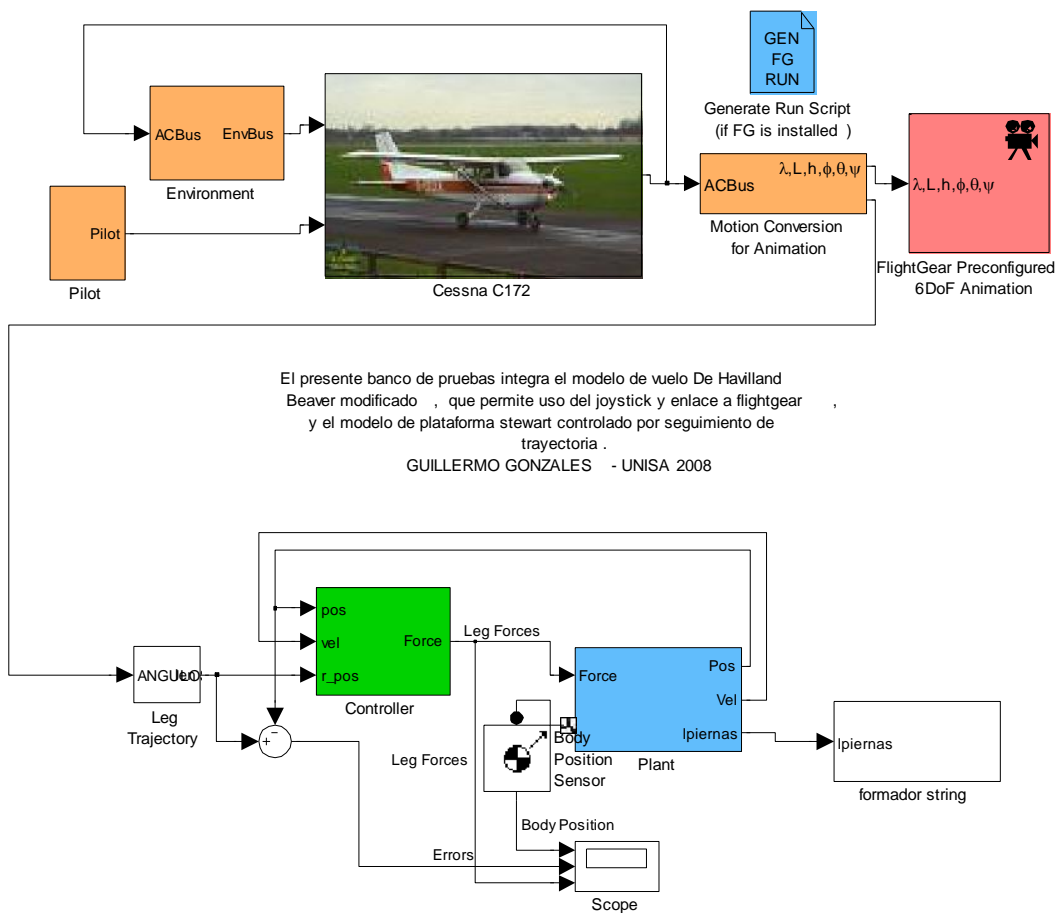


Figura 49. Esquema completo en Simulink para el proyecto

En este caso, la señal que se toma a la salida del bloque de simulación del vuelo es la de los ángulos de orientación de la aeronave, a partir de los cuales se calcula su movimiento en el espacio. Esta señal se introduce en el generador de trayectorias de referencia de la plataforma a fin de enlazar ambos subsistemas. Luego, se efectúa el cálculo de la longitud correspondiente a cada una de las piernas para producir el movimiento, lo cual a su vez se envía a un bloque llamado “formador de string”, que se encarga de conformar los comandos que serán enviados por el puerto serial a la controladora de servos y que a su vez harán posible el movimiento de los actuadores.

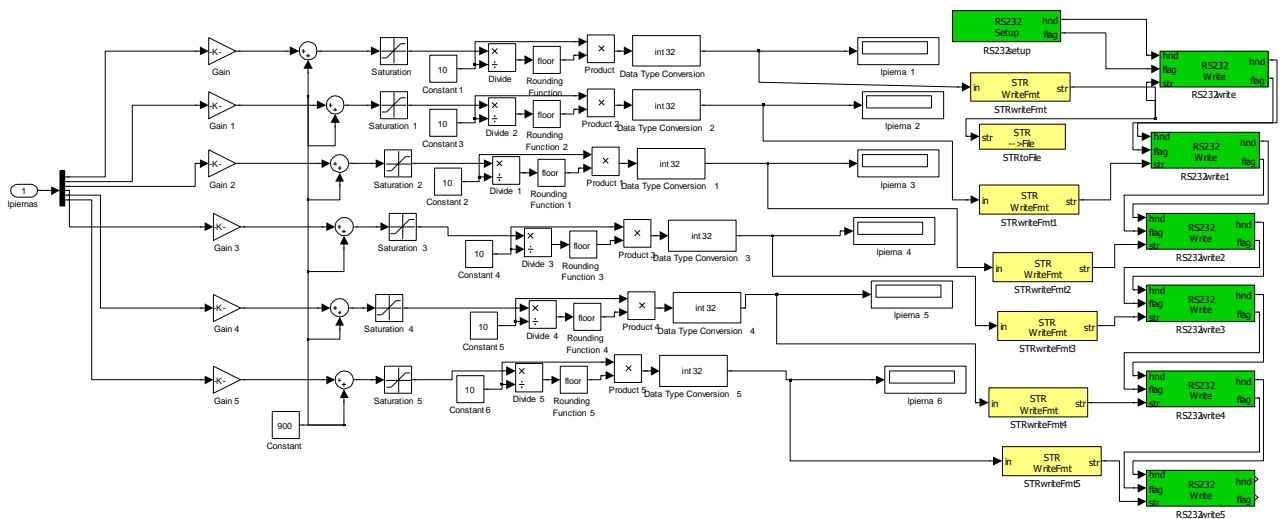


Figura 50. Subsistema formador de strings para control de servos

El bloque formador de string toma la información correspondiente a la longitud de cada pierna, que de acuerdo a la simulación de la plataforma varía entre un mínimo de 0.3m y un máximo de 1,1m, multiplica esta cantidad por 1000, añade una constante de 900 y lo aplica a un bloque de saturación con límite inferior de 1200 y límite superior de 2000. Este proceso asegura que se obtenga un valor numérico adecuado para la cantidad de microsegundos que emplea como comando la tarjeta controladora de servos.

Luego se aplica un proceso de redondeo a fin de que los valores sean múltiplos de diez a fin de mejorar la velocidad de transmisión de datos, una conversión de tipo de datos y finalmente la conformación del comando incluyendo el número de servo sobre el que actúa cada señal, para su envío por el puerto correspondiente mediante librerías de comunicación específicas de Simulink [20][21].

5.2 Resultados obtenidos

Se ha logrado una representación virtual del movimiento deseado, empleando Matlab y FlightGear sobre la misma PC.

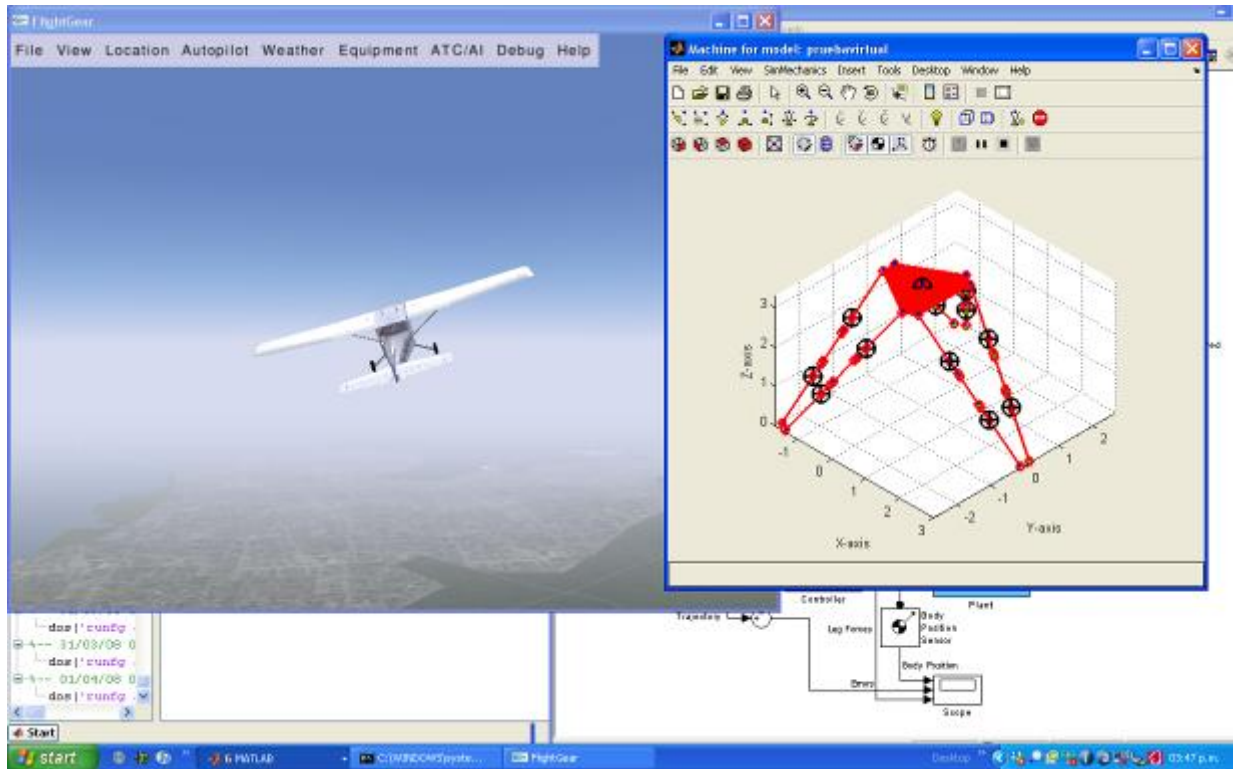


Figura 51. Simulación de plataforma con visualización en FlightGear

La prueba inicial enlazaba el resultado de la plataforma a la entrada de datos con el joystick, lo cual no representaba la orientación real del avión. Se efectuaron las correcciones y se enlazó el resultado de la plataforma a la dinámica de vuelo de la aeronave, mejorando significativamente el desempeño.

A fin de implementar el modelo propuesto en los módulos anteriores, se han seleccionado servomotores de aeromodelismo Futaba S-148, a los que se le ha acoplado el kit EMS Linear Servo Converter, que permite convertir el movimiento rotacional que realiza el servo originalmente, a un recorrido lineal a través del uso de un acople mecánico interno. De esta manera se obtiene un actuador lineal de bajo costo que cumple las siguientes características:

Control por modulación de ancho de pulso (PWM)
Alimentación: 6 VDC 1.6 Amp (consumo de 6 servos)
Recorrido mecánico del actuador: 2 cm



Figura 52. Servomotor FutabaS148 (izq) y kit para movimiento lineal (der)

Para interfazar los servomotores a la computadora se emplea la tarjeta Lynxmotion SSC-32 con drivers de la versión 2.00XE, la cual es ampliamente usada por aficionados a la electrónica y la robótica. Tiene la posibilidad de generar señales de control hasta para 32 servos simultáneamente y cuenta con una interfase serial del tipo RS232 para comunicación.

Características Técnicas:

Microcontrolador	Atmel ATMEGA8-16PI
EEPROM	24LC32P
Velocidad	14.75Mhz
Puerto serial	RS232 o TTL, 2400, 9600, 38.4k, 115.2k
Salidas	32 (Servo o TTL)
Consumo	31mA
Interfaz a PC	DB9F
Control de Servos	Hasta 32
Servos soportados	Futaba o Hitec
Rango de trabajo	cerca 170°
Resolución de Servos	1uS, 0.11°
Velocidad de servos	1uS/Segundo
Tipo de control	Inmediato, temporizado, velocidad o sincronizado
Dimensiones	7.60 x 5.85 mm



Figura 53. Tarjeta controladora Lynxmotion SSC-32

Se basa en el procesador Atmega 168, el cual ya viene con el firmware de control instalado de fabrica con lo cual se mejora el rendimiento del sistema. Tiene un entorno de pruebas amigable

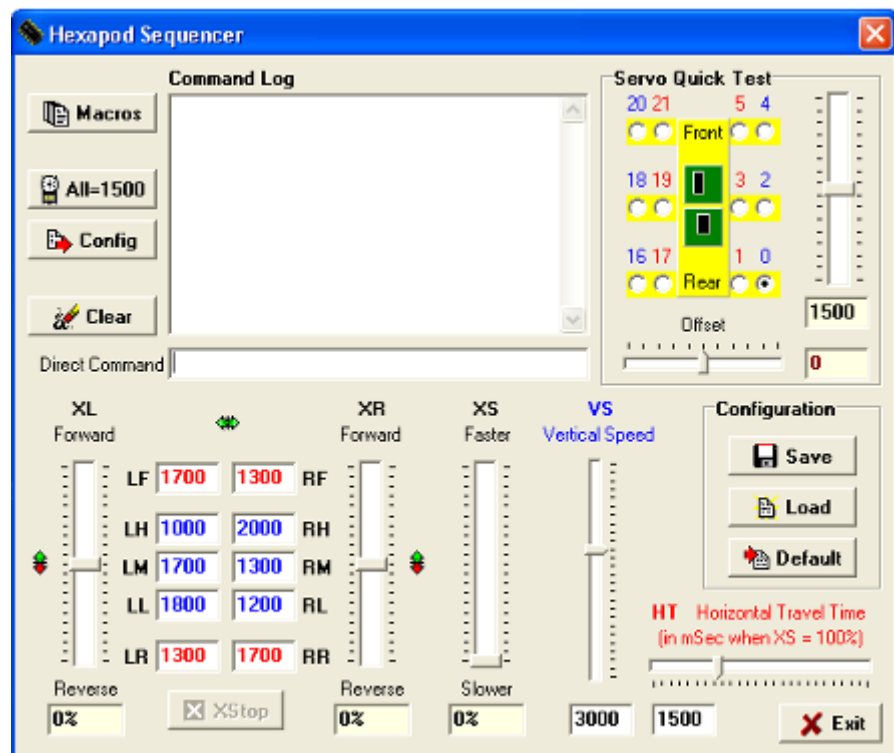


Figura 54. Pantalla de presentación de GUI suministrada por Lynxmotion

Para efectos del prototipo se viene realizando una interfaz gráfica de usuario a través de Matlab, la cual en un principio permitirá realizar el modulo de pruebas y posteriormente reemplazar las señales generadas por el usuario con los datos en el entorno en que corre la simulación del vuelo.

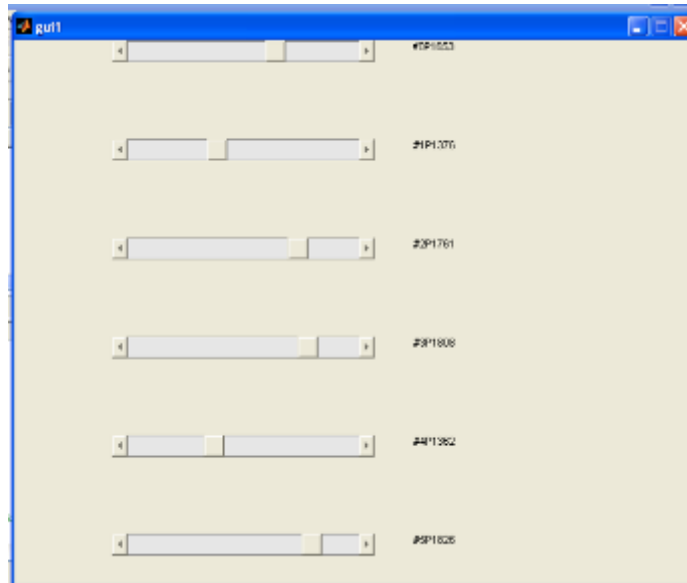


Figura 55. GUI desarrollada para el empleo de servomotores desde Matlab

De acuerdo a los resultados obtenidos, los servomotores originalmente se controlan con pulsos de periodos entre los 750 y 2500 microsegundos, sin embargo al acoplar el adaptador lineal estos rangos se restringen y se considera emplear entre 1200 y 2000 milisegundos para las posiciones extremas, en este rango es que se realizara el control para generar el desplazamiento de la plataforma.

Para la realización de la interfaz gráfica con controles independientes por servo se emplea la función `elslider_gui`, cuyo código en Matlab se incluye en el Anexo B.

Finalmente se logró generar el movimiento de prototipo enlazado a la simulación gráfica, empleando uno o dos computadores. Se apreció un rango de movimiento limitado atribuible al corto desplazamiento mecánico de los actuadores.

5.3 Prototipo

Se construyeron tres estructuras, en diversos materiales y dimensiones a fin de evaluar la implementación final del prototipo:

Estructura inicial en cartón corrugado: realizada únicamente como maqueta, permitió bosquejar la ubicación de cada uno de los componentes, así como las distancias aproximadas



Figura 56. Maqueta en cartón corrugado

Estructura con base y plato superior en espuma sintética: se eligió este material por su bajo peso y facilidad de trabajo, permitió montar los actuadores lineales construidos y las articulaciones universales. Al efectuar pruebas funcionales se evidenciaron las siguientes limitaciones:

- Al no haberse anclado las uniones de las articulaciones con los platos superior e inferior, el movimiento de los actuadores tendía a desgastar el material con lo cual se perdía capacidad de movimiento.
- La distancia entre las piernas en las uniones con el plato superior era muy corta, lo cual hacía que se interfiriesen al momento de moverse y restaba maniobrabilidad a la plataforma.

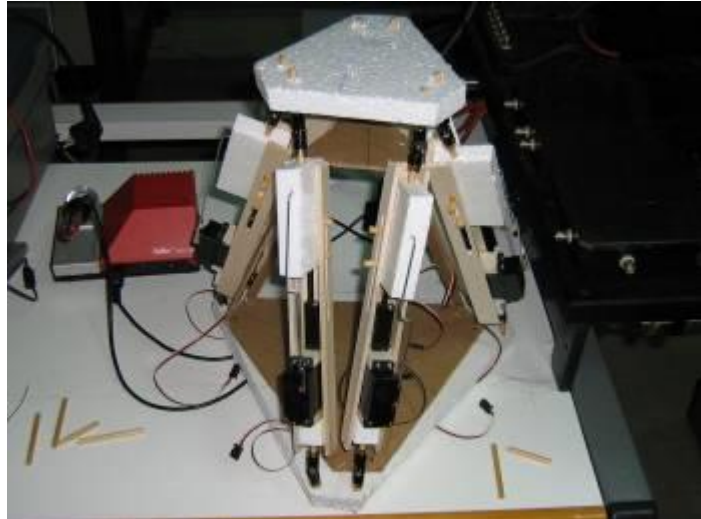


Figura 57. Estructura con platos en espuma sintética

Estructura realizada en madera balsa: este material presenta a la vez un bajo peso y buena solidez, es el mismo que se emplea para construir aeromodelos y maquetas. La fijación de las articulaciones a los platos superior e inferior se realizó mediante pernos zincados y arandelas, con lo que se asegura una correcta fijación y se evita desplazamientos durante el movimiento. Al efectuarse las pruebas los actuadores tuvieron un mejor desempeño y se corrigieron las interferencias.

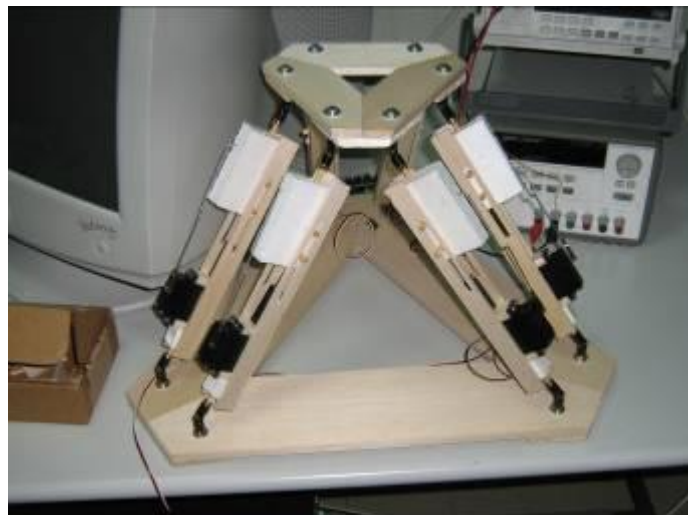


Figura 58. Estructura en madera balsa con pernos de sujeción

Conclusiones

El objetivo de la presente tesis ha sido el diseño e implementación del prototipo de una plataforma de movimiento de seis grados de libertad aplicable a simuladores de vuelo, que permitiera reproducir las sensaciones de movimiento espacial que se producen durante la operación de una aeronave, con miras a aplicar este modelo a los simuladores y entrenadores de vuelo con que actualmente cuenta la Fuerza Aérea del Perú.

En particular se ha planteado el empleo de la plataforma Stewart, manipulador paralelo que ha sido, desde su aparición, sujeto de estudio por sus grandes posibilidades de aplicación en la industria de la simulación de vuelo, así como otras tareas que requieran de posicionamiento de precisión y manipulación de grandes cargas.

Durante el presente estudio se ha enlazado el modelo de la plataforma Stewart, puesto a disposición de la comunidad académica por Mathworks®, con la simulación de vuelo generada desde Matlab® y visualizada a través de Flight Gear. Para el control de los actuadores se ha empleado una tarjeta de la firma Lynxmotion, la cual ha sido empleada con éxito en proyectos de robótica a nivel de laboratorio, contándose con diversos foros para solucionar dudas por parte de los desarrolladores. Como se aprecia, se ha hecho uso de herramientas disponibles en el mercado a fin de reducir los tiempos de desarrollo y asegurar el resultado final, aprovechando la gran cantidad de información y de investigaciones realizadas en estos temas [22].

Los primeros capítulos de la tesis sirven como marco teórico, dada la complejidad del tema de la simulación del vuelo y el control de la plataforma Stewart. Podemos conocer así el estado del arte actual en simulación de vuelo con reproducción de movimiento, las características de la plataforma y el software a emplear así como algunas normativas vigentes aplicables a este campo. En cuanto a la plataforma propiamente dicha, se hace un breve repaso de su historia así como sus características, lo cual permite establecer la idoneidad de la elección efectuada.

En los capítulos finales se hace uso intensivo de las herramientas de software para implementar el diseño planteado, a través de la simulación primero y después la construcción de los prototipos que se plantearon. El buen desempeño logrado en las simulaciones se vio reflejado en el prototipo real, mientras que los prototipos permitieron percibir importantes aspectos de la construcción a tener en cuenta en futuras implementaciones.

Para el presente trabajo se han empleado actuadores lineales eléctricos en razón de las dimensiones del prototipo final y los costos de desarrollo, mientras que en la mayoría de dispositivos en uso actualmente en simuladores se emplean actuadores hidráulicos, producto de los primeros diseños. Sin embargo, se recomienda para la implementación en mayor escala el empleo de las nuevas tecnologías de motores eléctricos en función de su mayor facilidad de instalación y mantenimiento, adicionalmente a las facilidades de modelamiento que brindan, dado que los actuadores hidráulicos presentan no linealidades que aumentan su complejidad. De acuerdo a especialistas y firmas fabricantes [23], la tendencia en los futuros desarrollos de plataformas de movimiento está dada por los motores y servomotores eléctricos, que si bien pueden ser más costosos resultan más beneficiosos por cuanto reducen notablemente los costos de operación.

Agradecimientos

El autor desea expresar su agradecimiento al Alto Mando de la Fuerza Aérea del Perú y al personal militar y civil del Centro de Desarrollo de Proyectos de la FAP, con cuya experiencia en el campo de la simulación de vuelo y sistemas de control de movimiento hicieron posible el desarrollo de la presente tesis.

Finalmente el agradecimiento al Ms. Pedro Huamaní Navarrete por la asesoría brindada, e igualmente al Director de la Escuela Profesional de Ing. Electrónica, Dr. Oscar Penny Cabrera, por el apoyo brindado con su asesoramiento tanto en el tema investigado como en las gestiones ante la Universidad para la presentación de esta tesis.

APENDICE A: CARACTERISTICAS DEL CESSNA 172

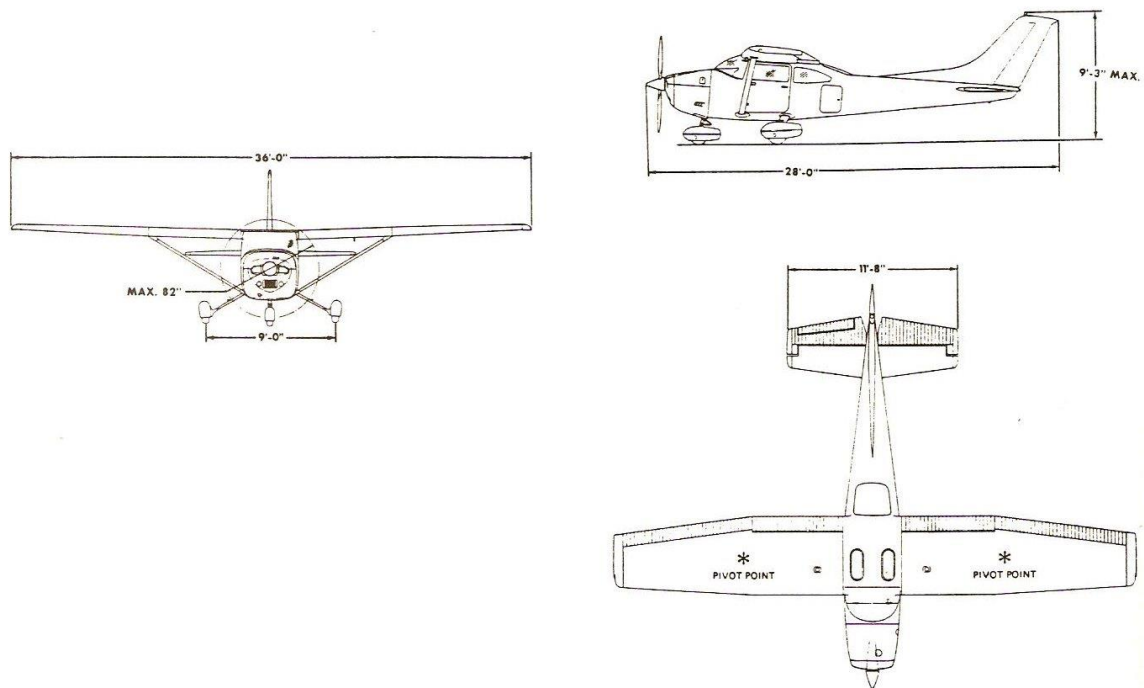


Figura 59. Vistas generales del Cessna C172

Características generales

- Tripulación: 1
- Capacidad: 3 pasajeros
- Longitud: 8,28 m
- Envergadura: 11 m
- Altura: 2,72 m
- Superficie alar: 16,20 m²
- Peso vacío: 743 kg
- Peso máximo de despegue: 1.110 kg
- Planta motriz: 1x motor de cuatro cilindros opuestos enfriados por aire Lycoming IO-360-L2A, 120 kW (160 HP)
- Hélices: bipala de velocidad constante, por motor.

Rendimiento

- Velocidad nunca a exceder (Vne): 163 nudos (302 km/hora)
- Velocidad máxima operativa (Vno): 123 nudos (228 km/hora)
- Alcance en vuelo: 1.270 km
- Techo de servicio: 13.500 ft ()
- Tasa de ascenso: 720 ft/min ()

Funciones de transferencia para el eje longitudinal

F.T. polinomial ángulo de ataque con respecto al elevador

$$\frac{-449854s - 7798686 - 355298 - 335164}{22055s^4 + 198952s^3 + 626286s^2 + 329825s + 18576}$$

F.T. factorizada ángulo de ataque con respecto al elevador

$$\frac{-449584 + 173302s^2 + 00454s + 00424}{22055(s^2 + 88994 + 277798s^2 + 0044s + 0029)}$$

Ganancia $K_{\text{gain}} = -1.830343$ para la F.T. ángulo de ataque a elevador

F.T. polinomial velocidad con respecto al elevador

$$\frac{-875615s + 9613807s + 49832852}{22055s^4 + 198952s^3 + 626286s^2 + 329825s + 18576}$$

F.T. factorizada velocidad con respecto al elevador

$$\frac{-875615s - 1147866s + 49602}{22055(s^2 + 88994 + 277798s^2 + 0044s + 0029)}$$

Ganancia $K_{\text{gain}} = 2760.037863$ para la F.T. velocidad a elevador

F.T. polinomial actitud de cabeceo con respecto a elevador

$$\frac{-7713340s - 1586000s - 908451}{22055s^4 + 198952s^3 + 626286s^2 + 329825s + 18576}$$

F.T. factorizada actitud de cabeceo con respecto a elevador

$$\frac{-7713340 + 19982s + 00589}{22055(s^2 + 88994 + 277798s^2 + 0044s + 0029)}$$

Ganancia $K_{\text{gain}} = -5.029704$ para la F.T. actitud de cabeceo a elevador

F.T. para el eje lateral-direccional

F.T. polinomial derrape a alerón

$$\frac{694302s^2 + 17900258s + 2689498}{22075s^2 + 3162196s + 6213579 + 3026188s + 5396173}$$

F.T. factorizada derrape a alerón

$$\frac{694302s+255517s+0.1508}{220752s+130127s+0.0179s^2+1.3405+10528s}$$

Ganancia $K_{\text{gain}} = -5.029704$ para la F.T. derrape a alerón

F.T. polinomial derrape a rudder

$$\frac{195634s+249841s+2971270s-512714s}{220752s+316219s+621257s+3026188s+539688s}$$

F.T. factorizada derrape a rudder

$$\frac{195634s-0.0179s+1.44019s+1.3294s}{220752s+130127s+0.0179s^2+1.3405+10528s}$$

Ganancia $K_{\text{gain}} = -0.950926$ para la F.T. derrape a rudder

F.T. polinomial alabeo a alerón

$$\frac{16517989s+21473354s+13277201s}{220752s+316219s+621257s+3026188s+539173s}$$

F.T. factorizada alabeo a alerón

$$\frac{16517989s^2+1.3001+8.0389s}{220752s+130127s+0.0179s^2+1.3405+10528s}$$

Ganancia $K_{\text{gain}} = 246.259658$ para la F.T. alabeo a alerón

F.T. polinomial alabeo a rudder

$$\frac{1060487s-3902643s-58493958s}{220752s+316219s+621257s+3026188s+539173s}$$

F.T. factorizada alabeo a rudder

$$\frac{1060487s-9.4949s+5.8106s}{220752s+130127s+0.0179s^2+1.3405+10528s}$$

Ganancia $K_{\text{gain}} = -108.488972$ para la F.T. alabeo a rudder

F.T. polinomial cabeceo a alerón

$$\frac{-7508413s^4 - 15814284s^3 - 3308966s^2 + 19037166s}{220752s^5 + 3162190s^4 + 6213579s^3 + 3026188s^2 + 539173s}$$

F.T. factorizada cabeceo a alerón

$$\frac{-7508413s^4 - 0.9773s^3 + 207903s^2 + 1.2479s}{220752s^5 + 130127s^4 + 0.0179s^3 + 1.3405s^2 + 10528s}$$

Ganancia $K_{\text{gain}} = 35.309434$ para la F.T. cabeceo a alerón

F.T. polinomial cabeceo a rudder

$$\frac{-2240954s^4 + 131775s^3 + 0.0720s^2 + 0.2865s}{220752s^5 + 130127s^4 + 0.0179s^3 + 1.3405s^2 + 10528s}$$

Ganancia $K_{\text{gain}} = -15.699819$ para la F.T. cabeceo a rudder

APENDICE B: CÓDIGO MATLAB PARA PRUEBA DE LOS SERVOMOTORES

Código de funciones desarrollado de acuerdo a manuales de Mathworks [24]

```
function elslider_gui
fig = figure(1);
sh1 = uicontrol(fig,'units','normal','pos',[.2,.7,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider1_callback);
eth1 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh1,'Value')),...
    'units','normal','pos',[.05,.7,.1,.05],...
    'Callback',@edittext1_callback);
val = 1600;
sh2 = uicontrol(fig,'units','normal','pos',[.2,.6,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider2_callback);
eth2 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh2,'Value')),...
    'units','normal','pos',[.05,.6,.1,.05],...
    'Callback',@edittext2_callback);
sh3 = uicontrol(fig,'units','normal','pos',[.2,.5,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider3_callback);
eth3 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh3,'Value')),...
    'units','normal','pos',[.05,.5,.1,.05],...
    'Callback',@edittext3_callback);
sh4 = uicontrol(fig,'units','normal','pos',[.2,.4,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider4_callback);
eth4 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh4,'Value')),...
    'units','normal','pos',[.05,.4,.1,.05],...
    'Callback',@edittext4_callback);
sh5 = uicontrol(fig,'units','normal','pos',[.2,.3,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
```

```

        'SliderStep',[0.05 0.2],...
        'Callback',@slider5_callback);
eth5 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh5,'Value')),...
    'units','normal','pos',[.05,.3,.1,.05],...
    'Callback',@edittext5_callback);
sh6 = uicontrol(fig,'units','normal','pos',[.2,.2,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider6_callback);
eth6 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh6,'Value')),...
    'units','normal','pos',[.05,.2,.1,.05],...
    'Callback',@edittext6_callback);
% -----
% Set the value of the edit text component String property
% to the value of the slider.

```

```

function slider1_callback(hObject,eventdata)
    Ser1=serial('COM8','BaudRate',115200,'Terminator','CR/LF');
    fopen(Ser1);
    val = round(get(hObject,'Value'));
    set(eth1,'String',num2str(val));
    temporary = sprintf('#0P%d',val); %store using sprintf function the
    fprintf(Ser1,temporary); % send the instruction to the SSC32
    fclose(Ser1)
end

```

```

% -----
% -----
% Set the value of the edit text component String property
% to the value of the slider.

```

```

function slider2_callback(hObject,eventdata)
    Ser1=serial('COM8','BaudRate',115200,'Terminator','CR/LF');
    fopen(Ser1);
    val = round(get(hObject,'Value'));
    set(eth2,'String',num2str(val));
    temporary = sprintf('#1P%d',val); %store using sprintf function the
    fprintf(Ser1,temporary); % send the instruction to the SSC32
    fclose(Ser1)
end

```

```

% -----
% -----
% Set the value of the edit text component String property
% to the value of the slider.

```

```

function slider3_callback(hObject,eventdata)
    Ser1=serial('COM8','BaudRate',115200,'Terminator','CR/LF');
    fopen(Ser1);
    val = round(get(hObject,'Value'));

```



```

    set(eth3, 'String', num2str(val));
    temporary = sprintf('#2P%d', val); %store using sprintf function the
    fprintf(Ser1, temporary); % send the instruction to the SSC32
    fclose(Ser1)
end
% -----
% -----
% Set the value of the edit text component String property
% to the value of the slider.

function slider4_callback(hObject,eventdata)
    Ser1=serial('COM8', 'BaudRate', 115200, 'Terminator', 'CR/LF');
    fopen(Ser1);
    val = round(get(hObject, 'Value'));
    set(eth4, 'String', num2str(val));
    temporary = sprintf('#3P%d', val); %store using sprintf function the
    fprintf(Ser1, temporary); % send the instruction to the SSC32
    fclose(Ser1)
end
% -----
% -----
% Set the value of the edit text component String property
% to the value of the slider.

function slider5_callback(hObject,eventdata)
    Ser1=serial('COM8', 'BaudRate', 115200, 'Terminator', 'CR/LF');
    fopen(Ser1);
    val = round(get(hObject, 'Value'));
    set(eth5, 'String', num2str(val));
    temporary = sprintf('#4P%d', val); %store using sprintf function the
    fprintf(Ser1, temporary); % send the instruction to the SSC32
    fclose(Ser1)
end
% -----
% -----
% Set the value of the edit text component String property
% to the value of the slider.

function slider6_callback(hObject,eventdata)
    Ser1=serial('COM8', 'BaudRate', 115200, 'Terminator', 'CR/LF');
    fopen(Ser1);
    val = round(get(hObject, 'Value'));
    set(eth6, 'String', num2str(val));
    temporary = sprintf('#5P%d', val); %store using sprintf function the
    fprintf(Ser1, temporary); % send the instruction to the SSC32
    fclose(Ser1)
end
% -----
end

```

Código MATLAB para pruebas conjuntas de servomotores

```
function multislider
fig = figure(1);
sh1 = uicontrol(fig,'units','normal','pos',[.2,.7,.5,.05],...
    'Style','slider',...
    'Max',2000,'Min',1200,'Value',1600,...
    'SliderStep',[0.05 0.2],...
    'Callback',@slider1_callback);
eth1 = uicontrol(fig,'Style','edit',...
    'String',num2str(get(sh1,'Value')),...
    'units','normal','pos',[.05,.7,.1,.05],...
    'Callback',@edittext1_callback);
val = 1600;
function slider1_callback(hObject,eventdata)
    Ser1=serial('COM8', 'BaudRate', 115200,'Terminator','CR/LF');
    fopen(Ser1);
    val = round(get(hObject,'Value'));
    set(eth1,'String',num2str(val));
    temporary = sprintf('#0P%d',val); %store using sprintf function the
    fprintf(Ser1,temporary); % send the instruction to the SSC32
    temporary1=sprintf('#1P%d',val);
    fprintf(Ser1,temporary1);
    temporary2 = sprintf('#2P%d',val); %store using sprintf function the
    fprintf(Ser1,temporary2); % send the instruction to the SSC32
    temporary3=sprintf('#3P%d',val);
    fprintf(Ser1,temporary3);
    temporary4 = sprintf('#4P%d',val); %store using sprintf function the
    fprintf(Ser1,temporary4); % send the instruction to the SSC32
    temporary5=sprintf('#5P%d',val);
    fprintf(Ser1,temporary5);
    fclose(Ser1)
end
```

APENDICE C: PROCEDIMIENTO PARA PRUEBAS USANDO UN SOLO COMPUTADOR

El equipo empleado cuenta con las siguientes características:

Marca y modelo	: HP Pavilion dv6420la Notebook
Procesador	: AMD Turion 64x2 TL-56 1.81 Ghz
Memoria RAM	: 1.5 Gb
Disco duro	: 160 GB
Gráficos	: Nvidia GForce Go6150
Sistema Operativo	: Windows XP service pack 2
Software instalado	: FlightGear v. 0.9.8 Matlab R2007a Simulink con los siguientes toolboxes: Aerospace Aerosim SimMechanics Librerías de comunicación por puerto serial SSC32 de Lynxmotion
Joystick	: Logitech dual action
Adicionales	: Adaptador USB-serial

Preparación de la plataforma:

1. Conectar los servomotores a la tarjeta controladora SSC32, empleando de preferencia los puertos numerados del 0 al 5.
2. Conectar la alimentación de la tarjeta a la de los servos mediante el Jumper respectivo, y luego los cables de alimentación a la fuente. Emplear solo fuentes de alimentación confiables, se requiere 6 VDC 1 Amp para hacer funcionar el sistema.
3. Conectar el puerto serial a la computadora mediante el adaptador.
4. Alimentar.

Preparación del computador

1. Iniciar el computador, cerrando la mayor cantidad de aplicaciones prescindibles en el arranque a fin de liberar recursos.
2. Conectar el adaptador USB-serial, verificar que el número de puerto COM corresponda al empleado para el programa.
3. Verificar el funcionamiento de los actuadores empleando el driver SSC32.

4. Abrir Matlab y cargar los demos de la Plataforma Stewart con seguimiento de trayectoria y De Havilland Beaver.
5. Abrir el programa correspondiente para pruebas con un solo computador, en la opción "*Simulation-Configuration parameters*" desactivar las opciones correspondientes a SimMechanics para evitar sobrecarga del computador.
6. Ejecutar desde Matlab el script de carga de FlightGear.
7. Iniciar la ejecución de la simulación

APENDICE D: PROCEDIMIENTO PARA PRUEBAS USANDO DOS COMPUTADORES

El equipo fuente cuenta con las siguientes características:

Marca y modelo : HP Pavilion dv6420la Notebook
Procesador : AMD Turion 64x2 TL-56 1.81 Ghz
Memoria RAM : 1.5 Gb
Disco duro : 160 GB
Gráficos : Nvidia GForce Go6150
Sistema Operativo : Windows XP service pack 2
Software instalado : FlightGear v. 0.9.8
Matlab R2007a
Simulink con los siguientes toolboxes:
Aerospace
Aerosim
SimMechanics
Librerías de comunicación por puerto serial
SSC32 de Lynxmotion
Joystick : Logitech dual action
Adicionales : Adaptador USB-serial, cable cross-over para conexión de red

El equipo target posee las siguientes características:

Marca y modelo : Compaq Evo
Procesador : Intel Pentium 4 1.5 Ghz
Memoria RAM : 1.0 Gb
Disco duro : 18,6 GB
Gráficos : Nvidia Vanta 16 Mb (Integrada)
Sistema Operativo : Windows XP service pack 2
Software instalado : FlightGear v 0.9.8

Configuración de red punto a punto

Para la conexión de ambos equipos se emplea un cable cross-over o par cruzado.

Se ha establecido las siguientes IP:

Fuente:

IP address	192.168.0.2
Mascara	255.255.255.0
Gateway	en blanco

Target:

IP address	192.168.0.3
Máscara	255.255.255.0
Gateway	192.168.0.2

Para ambas se crea el grupo de trabajo “PROYECTO”, y se desactivan los firewalls, luego de lo cual se hacen pruebas de comunicación mediante el comando ping.

Preparación de la plataforma:

1. Conectar los servomotores a la tarjeta controladora SSC32, empleando de preferencia los puertos numerados del 0 al 5.
2. Conectar la alimentación de la tarjeta a la de los servos mediante el Jumper respectivo, y luego los cables de alimentación a la fuente. Emplear solo fuentes de alimentación confiables, se requiere 6 VDC 1 Amp para hacer funcionar el sistema.
3. Conectar el puerto serial a la computadora (fuente) mediante el adaptador.
4. Alimentar.

Preparación de los computadores

1. Iniciar el computador fuente, cerrando la mayor cantidad de aplicaciones prescindibles en el arranque a fin de liberar recursos.
2. Conectar el adaptador USB-serial, verificar que el número de puerto COM corresponda al empleado para el programa.
3. Verificar el funcionamiento de los actuadores empleando el driver SSC32.
4. Conectar el computador target con el cable cross-over, e iniciar el computador target. Verificar la comunicación de datos.
5. Abrir en el computador fuente Matlab y cargar los demos de la Plataforma Stewart con seguimiento de trayectoria y De Havilland Beaver.
6. Abrir en el computador fuente el programa correspondiente para pruebas con dos computadores. Verificar que en los bloques de preconfigured 6DoF animation figure la dirección IP del target.
7. Generar en el computador fuente un script que refleje la ubicación de los archivos de FlightGear en el computador target, y copiarlo al computador target.
8. Iniciar FlightGear en el computador target
9. Iniciar la ejecución de la simulación en el computador fuente

APENDICE E: MATRICES DE ESTADO DE LA PLANTA USANDO MATLAB

A partir del modelo en SimMechanics se extrae el modelo lineal de la plataforma para hacer la evaluación matemática del modelo. Para ello se selecciona el modelo en situación de equilibrio correspondiente al estado inicial de reposo, que se encuentra en el archivo `StewartPlatformEquilibrium`, y se simula:

```
>>sim('StewartPlatformEquilibrium');u=Forces';
```

Con lo que se simula el sistema, para este caso u corresponde a datos de entrada del Workspace de Matlab que se van a aplicar al modelo. Luego se efectúa la linealización alrededor del punto de equilibrio:

```
>>[A,B,C,D]=linmod('StewartPlatformPlant',[ ],u);
```

Esto genera un modelo de espacio de estados LTI desde un modelo SimMechanics, que retorna el número de estados del sistema (52 en total). Como se trata de un modelo cerrado se puede hacer una reducción usando la función *minreal*:

```
>>[A,B,C,D]=minreal(A,B,C,D);
```

Con lo que se obtienen las siguientes matrices de estados

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 9.0418 & 24682 & -20513 & -28144 & 1.4778 & -4.7826 & 0 & 0 & 0 & 0 & 0 & 0 \\ 24682 & 9.0418 & -4.7826 & 1.4778 & -28144 & -20513 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.4778 & -4.7826 & 9.0418 & 24682 & -20513 & -28144 & 0 & 0 & 0 & 0 & 0 & 0 \\ -28144 & -20513 & 24682 & 9.0418 & -4.7826 & 1.4778 & 0 & 0 & 0 & 0 & 0 & 0 \\ -20513 & -28144 & 1.4778 & -4.7826 & 9.0418 & 24682 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4.7826 & 1.4778 & -28144 & -20513 & 24682 & 9.0418 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Referencias bibliográficas

[1] Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale. *“Aircraft Systems. Flight Control Systems”*. Lecture Notes, Version 2004.

[2] P. Grant, B. Yam, R. Hosman y J. Schroeder. *“Effect of simulator motion on pilot behavior and perception”*. Journal of aircraft. Vol 43, N° 6. Noviembre-Diciembre 2006.

[3] D. Berger, J. Schulte-Pelkum, H. Bülthoff. *“Simulating believable forward accelerations on a Stewart motion platform”*. Max Planck Institut für biologische Kybernetik. Technical report No. 159. Febrero 2007.

[4] Watson School of Engineering. *“Flight and Ground Vehicle Simulation Update”*. State University of New York 1998.

[5] J. Flach *“Using Simulators for Skill Development”*. Wright State University.

[6] A. Lee. *“Flight Simulation: Virtual Environments in Aviation”*. Ashgate Publishing Company. 2005

[7] Federal Aviation Administration. *“Airplane flight training device qualification – Advisory circular AC 120-45A”*. U.S. Department of Transportation, Febrero 1992.

[8] E. Martin. *“Simulation of the Motion & Force environment using whole-body motion simulators”*. United States Air Force Research Laboratory, Patterson, Ohio.

[9] Organización de Aviación Civil Internacional (OACI). *“Manual para la calificación de simuladores de vuelo”* – Documento 9625.

- [10] T. Nash. *“Motion cueings Essentials”*. Simulation Technology, MT&SN Vol.6 Issue 1. Enero 2004.
- [11] C. Hernández, J. Fernández, G. Baptista. *“Metodología de la Investigación”* Segunda Edición. Editorial Mc Graw Hill, México 1999.
- [12] B. Dasgupta, T.S. Mruthyunjaya. *“The Stewart platform manipulator: a review”*. Mechanism and machine theory 35 (2000) 15-40.
- [13] G. Legnani. *“Robotica industriale”*. Editrice Ambrosiana. Milano, 2003.
- [14] <http://www.FlighGear.org>
- [15] The Mathworks. *“Aerospace toolbox 2. User’s guide”*. Marzo 2007.
- [16] Unmanned dynamics. *“Aerosim – aeronautical simulation blockset V. 1.2”*.
- [17] The Mathworks. *“SimMechanics for use with Simulink. User’s guide v1”*. Julio 2002.
- [18] K. Ogata. *“Ingeniería de Control Moderna”*. Tercera Edición. Editorial Prentice Hall, México 1998.
- [19] J. Gutridge. *“Three Degree of Freedom Simulator Motion Cueing Using Classical Washout Filters and Acceleration Feedback”*. Tesis para el Grado de Master of Science en Ingeniería Aeroespacial. Virginia Tech, Estados Unidos 2004.
- [20] L. Daga. *“RS232 Blockset for use with simulink”*. Italia, septiembre 2004.

[21] L. Daga. *“Communication & Strings Blockset for use with simulink”*. Italia, septiembre 2004.

[22] Z. Lei, J. Hangzhou, L. Hongren. *“PC Based high quality and low cost flight simulator”*. Proceedings of the IEEE International Conference on Automation and Logistics, Agosto 2007, Jinan- China.

[23] C. Bartel, D. Foster *“Life cycle motion base cost comparison: electric vs hydraulic”*. Industrial Control Division, Moog Inc.

[24] The Mathworks. *“Simulink 7 – Writing S Functions”*. Marzo 2007

[25] J. Roskam. *“Airplane flight dynamics and automatic flight controls”*. DAR Corporation, 1995. Estados Unidos de Norteamérica