



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

ÁREA DEPARTAMENTAL DE ENGENHARIA DE SISTEMAS DE POTÊNCIA E AUTOMAÇÃO

ENGENHARIA ELECTROTÉCNICA (AUTOMAÇÃO E ELECTRÓNICA INDUSTRIAL)

**MODELIZAÇÃO DE UM SISTEMA DE  
SIMULAÇÃO DE UMA AERONAVE APLICADO A  
UMA PLATAFORMA ELECTROMECHANICA DE  
GOUGH-STEWART**

Nelson Filipe Pereira dos Santos

(Licenciado pré-Bolonha)

DISSERTAÇÃO PARA OBTENÇÃO DO GRAU DE MESTRE  
EM ENGENHARIA ELECTROTÉCNICA

Orientador:

Doutor José Carlos Lourenço Quadrado

Co-Orientadora:

Mestre Rita Marcos Fontes Murta Pereira

Júri:

Presidente: Doutor Elmano da Fonseca Margato

Vogais:

Doutor Ângelo Manuel Rego e Silva Martins

Doutor José Carlos Lourenço Quadrado

Mestre Rita Marcos Fontes Murta Pereira

SETEMBRO DE 2010

## **Resumo**

Esta tese apresenta o desenvolvimento de um simulador de voo, através da aplicação de um modelo dinâmico de uma aeronave e de sistemas de realidade virtual a uma plataforma electromecânica de Gough-Stewart. Para o efeito, utiliza-se o modelo da aeronave Cessna-172. Neste sentido, apresenta-se o estudo e a análise do sistema de realidade virtual utilizado, confirma-se o funcionamento do modelo matemático da aeronave em ferramenta de cálculo e apresenta-se o modelo da cinemática inversa da plataforma, cuja validação é realizada recorrendo-se a um sensor inercial. Desenvolve-se programação específica para a comunicação entre o *software* de cálculo numérico e a plataforma electromecânica de Gough-Stewart.

## **Palavras-chave**

Manipulador Paralelo; Plataforma Gough-Stewart; Simulação; Realidade Virtual; *Real-Time*; Comunicação Industrial.

## **Abstract**

This thesis presents the development of a flight simulator, applying a dynamic model of airplane and virtual reality systems, to the electromechanical Gough-Stewart platform. In this case, was been used the Cessna-172 aircraft model and present the study and analysis of virtual reality system. Confirm the operation of mathematical aircraft model in software calculation tool and presents the inverse kinematics model of the platform with validation of the performed using an inertial sensor. Develop specific program for communication between software calculation tool and electromechanical Gough-Stewart platform.

## **Keywords:**

Parallel Manipulator; Gough-Stewart Platform; Simulation; Virtual Reality; Real Time; Industrial Communication.

## **Agradecimentos**

Finalizada uma etapa particularmente importante da minha vida, não poderia deixar de expressar o mais profundo agradecimento a todos aqueles que me apoiaram nesta longa caminhada e contribuíram para a realização deste trabalho.

Ao orientador, Doutor José Carlos Quadrado, o meu maior agradecimento por toda a disponibilidade, pelos conselhos científicos, pelo apoio incondicional e compreensão que sempre manifestou. Desejando-lhe sucesso a nível profissional e pessoal.

À co-orientadora, Mestre Rita Marcos Pereira pelos conselhos científicos, técnicos, disponibilidade, confiança e amizade demonstradas ao longo deste trabalho. Desejando-lhe as maiores felicidades a nível académico, profissional e pessoal.

Ao Eng. Ricardo Luís, Eng. Pedro Fonte, Eng. Armando Cordeiro e Eng. Graça Almeida pelos conselhos técnicos. Desejando-lhes sucesso a nível académico, profissional e pessoal.

Aos docentes da Secção de Electrometria, pela compreensão demonstrada, permitindo a conclusão deste trabalho. Desejando-lhe sucesso a nível académico, profissional e pessoal.

Aos meus familiares por todo apoio, compreensão e motivação que me proporcionaram ao longo do percurso académico.

À Sandra Sales, por fazer parte da minha vida, por partilhar comigo todos os momentos, sempre com incentivo e compreensão.

## Lista de Símbolos

$Y_{T1}$	Coordenada do topo do vértice T1 segundo o eixo Y
$X_{T2}$	Coordenada do topo do vértice T2 segundo o eixo X
$Y_{T4}$	Coordenada do topo do vértice T4 segundo o eixo Y
$X_{T1}$	Coordenada do topo do vértice T1 segundo o eixo X
$Z_{T1}$	Coordenada do topo do vértice T1 segundo o eixo Z
$Y_{T2}$	Coordenada do topo do vértice T2 segundo o eixo Y
$Z_{T2}$	Coordenada do topo do vértice T2 segundo o eixo Z
$X_{T3}$	Coordenada do topo do vértice T3 segundo o eixo X
$Y_{T3}$	Coordenada do topo do vértice T3 segundo o eixo Y
$Z_{T3}$	Coordenada do topo do vértice T3 segundo o eixo Z
$X_{T4}$	Coordenada do topo do vértice T4 segundo o eixo X
$Z_{T4}$	Coordenada do topo do vértice T4 segundo o eixo Z
$X_{T5}$	Coordenada do topo do vértice T5 segundo o eixo X
$Y_{B5}$	Coordenada do topo do vértice T5 segundo o eixo Y
$Z_{T5}$	Coordenada do topo do vértice T5 segundo o eixo Z
$X_{T6}$	Coordenada do topo do vértice T6 segundo o eixo X
$Y_{T6}$	Coordenada do topo do vértice T6 segundo o eixo Y
$Z_{T6}$	Coordenada do topo do vértice T6 segundo o eixo Z
$Z_{B5}$	Coordenada da base do vértice B5 segundo o eixo Z
$Y_{B6}$	Coordenada da base do vértice B6 segundo o eixo Y
$Z_{B6}$	Coordenada da base do vértice B6 segundo o eixo Z
$\beta$	Ângulo de derrapagem

$C_L$	Coeficiente de sustentação da aeronave
$Y_{B1}$	Coordenada da base do vértice B1 segundo o eixo Y
$Z_{B1}$	Coordenada da base do vértice B1 segundo o eixo Z
$X_{B2}$	Coordenada da base do vértice B2 segundo o eixo X
$Y_{B2}$	Coordenada da base do vértice B2 segundo o eixo Y
$Z_{B2}$	Coordenada da base do vértice B2 segundo o eixo Z
$X_{B3}$	Coordenada da base do vértice B3 segundo o eixo X
$Y_{B3}$	Coordenada da base do vértice B3 segundo o eixo Y
$Z_{B3}$	Coordenada da base do vértice B3 segundo o eixo Z
$X_{B4}$	Coordenada da base do vértice B4 segundo o eixo X
$Y_{B4}$	Coordenada da base do vértice B4 segundo o eixo Y
$Z_{B4}$	Coordenada da base do vértice B4 segundo o eixo Z
$X_{B5}$	Coordenada da base do vértice B5 segundo o eixo X
$Y_{B5}$	Coordenada da base do vértice B5 segundo o eixo Y
$X_{B6}$	Coordenada da base do vértice B6 segundo o eixo X
$q_d$	Pressão dinâmica
$\dot{p}$	Aceleração angular em relação ao solo segundo $p$
$\dot{q}$	Aceleração angular em relação ao solo segundo $q$
$\dot{r}$	Aceleração angular em relação ao solo segundo $r$
$g$	Aceleração da gravidade para o local actual
$\dot{W}_{Wind}$	Aceleração do Vento
$a_x$	Aceleração segundo eixo $x$
$a_y$	Aceleração segundo eixo $y$
$a_z$	Aceleração segundo eixo $z$

$\delta_a$	<i>Aileron</i>
$Alt$	Altitude da aeronave
$\alpha$	Ângulo de ataque
$e_0$	Ângulo inicial de rotação da aeronave em relação ao solo
$e_x$	Ângulo inicial de rotação da aeronave em relação ao solo segundo eixo $x$
$e_y$	Ângulo inicial de rotação da aeronave em relação ao solo segundo eixo $y$
$e_z$	Ângulo inicial de rotação da aeronave em relação ao solo segundo eixo $z$
$S$	Área da Asa
$M_p$	Binário de hélice
$X_a$	Centro de massa em relação ao referencial da aeronave segundo o eixo $x$
$Y_a$	Centro de massa em relação ao referencial da aeronave segundo o eixo $y$
$Z_a$	Centro de massa em relação ao referencial da aeronave segundo o eixo $z$
$J$	Coefficiente $J$ da hélice
$C_Y$	Coefficiente de força lateral
$V_a$	Coefficiente de força lateral
$C_Y$	Coefficiente de força lateral
$C_T$	Coefficiente de impulso
$C_P$	Coefficiente de potência
$C_D$	Coefficiente de resistência
$C_D$	Coefficiente de resistência da aeronave
$C_L$	Coefficiente de sustentação
$C_{L0}$	Coefficiente de sustentação para o mínimo de coeficiente de resistência
$C_{L0}$	Coefficiente de sustentação para zero de ângulo de ataque
$C_m$	Coefficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa

$C_{m0}$	Coeficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa para zero de ângulo de ataque
$C_m$	Coeficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa
$C_l$	Coeficiente do Momento de <i>Roll</i> em relação ao centro aerodinâmico da asa
$C_l$	Coeficiente do Momento de <i>Roll</i> em relação ao centro aerodinâmico da asa
$C_n$	Coeficiente do Momento de <i>Yaw</i> em relação ao centro aerodinâmico da asa
$C_n$	Coeficiente do Momento de <i>Yaw</i> em relação ao centro aerodinâmico da asa
$C_T$	Coeficientes $C_T$ da hélice
$C_P$	Coeficientes $C_P$ da hélice
$D$	Componente $D$ do vector de saída <i>Aerodynamic Force</i>
$Y$	Componente $Y$ do vector de saída <i>Aerodynamic Force</i>
$L$	Componente $L$ do vector de saída <i>Aerodynamic Force</i>
$L_a$	Componente $L_a$ do vector de saída <i>Aerodynamic Moment</i>
$M$	Componente $M$ do vector de saída <i>Aerodynamic Moment</i>
$N$	Componente $N$ do vector de saída <i>Aerodynamic Moment</i>
$X_{B1}$	Coordenada da base do vértice B1 segundo o eixo X
$c$	Corda média da aerodinâmica da asa
$\rho$	Densidade do ar
$\dot{Alt}$	Derivada da altitude da aeronave
$\dot{Lat}$	Derivada da latitude da aeronave
$\dot{Lon}$	Derivada da longitude da aeronave
$\dot{V}_D$	Derivada da velocidade em relação ao Geóide segundo D
$\dot{V}_E$	Derivada da velocidade em relação ao Geóide segundo E
$\dot{V}_N$	Derivada da velocidade em relação ao Geóide segundo N
$C_L^\alpha$	Derivada de coeficiente de sustentação em ordem a $\alpha$

$C_L^{\delta_f}$	Derivada de coeficiente de sustentação em ordem a $\delta_f$
$C_L^{\delta_e}$	Derivada de coeficiente de sustentação em ordem a $\delta_e$
$C_L^{\dot{\alpha}}$	Derivada de coeficiente de sustentação em ordem a $\dot{\alpha}$
$C_L^q$	Derivada de Coeficiente de sustentação em ordem a $q$
$C_L^M$	Derivada de Coeficiente de sustentação em ordem a $M$
$\dot{\alpha}$	Derivada do Ângulo de ataque
$\dot{\beta}$	Derivada do Ângulo de derrapagem
$C_Y^\beta$	Derivada do coeficiente de força lateral em ordem a $\beta$
$C_Y^{\delta_a}$	Derivada do coeficiente de força lateral em ordem a $\delta_a$
$C_D^{\delta_r}$	Derivada do coeficiente de força lateral em ordem a $\delta_r$
$C_Y^p$	Derivada do coeficiente de força lateral em ordem a $p$
$C_Y^r$	Derivada do coeficiente de força lateral em ordem a $r$
$C_D^{\delta_f}$	Derivada do coeficiente de resistência em ordem a $\delta_f$
$C_D^{\delta_e}$	Derivada do coeficiente de resistência em ordem a $\delta_e$
$C_D^a$	Derivada do coeficiente de resistência em ordem a $a$
$C_D^r$	Derivada do coeficiente de resistência em ordem a $r$
$C_D^M$	Derivada do coeficiente de resistência em ordem a $M$
$C_m^\alpha$	Derivada do coeficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa em ordem a $\alpha$
$C_m^{\delta_f}$	Derivada do coeficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa em ordem a $\delta_f$
$C_m^{\delta_e}$	Derivada do coeficiente do Momento de <i>Pitch</i> em relação ao centro aerodinâmico da asa em ordem a $\delta_e$

- $C_m^{\dot{\alpha}}$  Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $\dot{\alpha}$
- $C_m^q$  Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $q$
- $C_m^M$  Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $M$
- $C_l^\beta$  Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\beta$
- $C_l^{\delta_a}$  Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\delta_a$
- $C_l^{\delta_r}$  Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\delta_r$
- $C_l^p$  Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $p$
- $C_l^r$  Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $r$
- $C_n^\beta$  Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\beta$
- $C_n^{\delta_a}$  Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\delta_a$
- $C_n^{\delta_r}$  Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\delta_r$
- $C_n^p$  Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $p$
- $C_n^r$  Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $r$

$R_{p2n}$	<i>Direction Cosine Matrix Body (DCM)</i>
$\delta_e$	<i>Elevator</i>
$b$	Envergadura da asa
$\delta_f$	<i>Flap</i>
$F_p$	Força de impulso da hélice
$AConGnd$	Indicador de Aeronave no solo
$\phi$	Latitude da aeronave
$Lat$	Latitude da aeronave
$Lon$	Longitude da aeronave
$C_{D0}$	Mínimo coeficiente de resistência da aeronave
$J_x$	Momento de inércia referenciado ao centro de massa actual $x$
$J_y$	Momento de inércia referenciado ao centro de massa actual $y$
$J_z$	Momento de inércia referenciado ao centro de massa actual $z$
$J_{xz}$	Momento de inércia referenciado ao centro de massa actual $xz$
$R$	Raio da hélice
$R_{equiv}$	Raio equivalente
$R_{normal}$	Raio normal
$M$	Relação entre Velocidade do ar e Velocidade do som à altitude actual
$\delta_r$	<i>Rudder</i>
$r_{Wind}$	Vector aceleração angular causado pelo vento.
$\dot{v}_{Wind}$	Vector aceleração da turbulência
$q_{Wind}$	Vector aceleração do vento
$\Omega$	Velocidade angular da rotação da hélice (rad/s)
$\omega_{in}^p$	Velocidade angular de rotação da Terra.

$p$	Velocidade angular em relação ao solo segundo $p$
$q$	Velocidade angular em relação ao solo segundo $q$
$r$	Velocidade angular em relação ao solo segundo $r$
$u_{aircraft}$	Velocidade da aeronave
$C_D$	Velocidade do ar
$V_{body}$	Velocidade do centro de massa em relação ao solo
$a$	Velocidade do som à altitude actual
$W_{body}$	Velocidade do vento em relação ao centro de massa
$V_D$	Velocidade em relação ao Geóide segundo D
$V_E$	Velocidade em relação ao Geóide segundo E
$V_N$	Velocidade em relação ao Geóide segundo N
$a$	Distância entre juntas superiores e a topo plataforma electromecânica
$agl$	Altura em relação ao solo
$b$	Distância entre juntas inferiores e a base da plataforma electromecânica
B1	Coordenada do vértice da base da plataforma Gough-Stewart B1
B2	Coordenada do vértice da base da plataforma Gough-Stewart B2
B3	Coordenada do vértice da base da plataforma Gough-Stewart B3
B4	Coordenada do vértice da base da plataforma Gough-Stewart B4
B5	Coordenada do vértice da base da plataforma Gough-Stewart B5
B6	Coordenada do vértice da base da plataforma Gough-Stewart B6
$c$	Distância lateral da base entre B1 e B2
$d$	Distância lateral da base entre B2 e B3
$e$	Distância lateral da topo entre T2 e T3
$f$	Distância lateral da topo entre T1 e T2
$F_s$	Frequência de amostragem
L1	Distância da “perna” L1

L2	Distância da “perna” L2
L3	Distância da “perna” L3
L4	Distância da “perna” L4
L5	Distância da “perna” L5
L6	Distância das pernas L6
msl	Altura em relação ao nível médio do mar
nc <sub>i</sub>	Nível de combustível
px	Posição central da topo da plataforma segundo eixo x
py	Posição central da topo da plataforma segundo eixo y
pz	Posição central da topo da plataforma segundo eixo z
$R_{meridian}$	Raio do meridiano
T1	Coordenada do vértice do topo da plataforma Gough-Stewart T1
T2	Coordenada do vértice do topo da plataforma Gough-Stewart T2
T3	Coordenada do vértice do topo da plataforma Gough-Stewart T3
T4	Coordenada do vértice do topo da plataforma Gough-Stewart T4
T5	Coordenada do vértice do topo da plataforma Gough-Stewart T5
T6	Coordenada do vértice do topo da plataforma Gough-Stewart T6
vm	Velocidade de rotação do motor de combustão interna
vm <sub>i</sub>	Velocidade inicial do motor de combustão interna
X	Sistema de eixo de referência X
x	Sistema de eixo móvel x
y	Sistema de eixo de móvel y
Y	Sistema de eixo de referência Y
z	Sistema de eixo de móvel z
Z	Sistema de eixo de referência Z
$\alpha$	Ângulo de rotação segundo eixo x
$\beta$	Ângulo de rotação segundo eixo y

$\gamma$       Ângulo de rotação segundo eixo z

## **Abreviaturas**

USB – Universal Serial Bus

TCP/IP – Protocolo de comunicação da rede *Ethernet*

## **Estrangeirismos**

Ao longo do texto são utilizados estrangeirismos sobretudo pela sua generalização no uso e no contexto em que são empregues. Como, por exemplo, os termos ingleses *Roll*, *Pitch*, *Heave*, *Yaw*, *Surge* e *Sway* utilizados na aeronáutica.

Todos os estrangeirismos utilizados ao longo do texto serão assinalados em itálico.

# Índice

1 – Introdução.....	1
1.1 – Objectivos.....	1
1.2 – Motivação.....	2
1.3 – Enquadramento do Trabalho .....	3
1.4 – Estrutura do Trabalho.....	3
2 – Estado da Arte .....	4
2.1 – Simuladores de Voo .....	4
2.2 – Realidade Virtual.....	11
3 – Projecto do Sistema .....	15
3.1 – Introdução.....	15
3.2 – Plataforma de Gough-Stewart .....	15
3.3 – Modelo Matemático da Aeronave .....	20
3.3.1 – Descrição do Modelo Aerodinâmico.....	23
3.3.2 – Descrição do Modelo de Propulsão.....	29
3.3.3 – Descrição do Modelo Atmosférico e do Modelo Inercial .....	31
3.3.4 – Descrição do Modelo da Terra .....	33
3.3.5 – Descrição do Modelo das Equações do Movimento .....	36
3.4 – <i>Software</i> de Simulação FlightGear®.....	41
3.5 – Realidade Virtual.....	44
3.5.1 – Conceitos Sobre Visualização .....	45
3.5.2 – Técnicas de Produção de Pares Estereoscópicos.....	47
3.5.3 – Dispositivos de Visualização.....	49
4 – Implementação .....	52
4.1 – Introdução.....	52
4.2 – Equipamento e <i>Software</i> .....	53

4.2.1 – <i>Joystick</i> .....	53
4.2.2 – Modelo Cessna-172 .....	55
4.2.3 – Óculos de Realidade Virtual.....	56
4.2.4 – Plataforma Electromecânica.....	56
4.3 – Simulações do Modelo Matemático Cessna-172 .....	59
4.4 – Ensaio Laboratoriais .....	64
4.5 – Sistema Global.....	69
5 – Conclusões.....	71
5.1 – Considerações Finais .....	71
5.2 – Perspectivas de Desenvolvimento Futuro .....	73
Referências Bibliográficas.....	75
Anexos.....	78
Anexo A – Listagem do Programa de Inicialização do <i>FlightGear</i> <sup>®</sup> .....	78
Anexo B – Listagem do Programa de Comunicação entre <i>FlightGear</i> <sup>®</sup> e o <i>MatLab/Simulink</i> <sup>®</sup> .....	78
Anexo C – Características do Sensor Inercial XSENS MTI.....	83

## Índice de Figuras

Figura 1.1 – Diagrama de blocos geral do simulador.....	2
Figura 2.1 – Simulador de Antoniette em 1910, [1].....	4
Figura 2.2 – Simulador de Edwin Link, [3].....	5
Figura 2.3 – Construção dos simuladores “ANT-18 Basic Instruments Trainer”, [4] .....	6
Figura 2.4 – Simulador Boieng 377 de Curtiss-Wright, [3].....	7
Figura 2.5 – Simulador Comet IV da Redifon, [5].....	8
Figura 2.6 – Sistema de visualização acoplado a um simulador, [1] .....	9
Figura 2.7 – Simulador de voo actual da companhia Aérea QANTAS, [6].....	10
Figura 2.8 – Sensorama de Norton L. Heilig, [7].....	11
Figura 2.9 – Capacete de realidade virtual de Ivan Sutherland, [9] .....	12
Figura 2.10 – Projecto VCASS de Thomas Fueness, [10] .....	13
Figura 2.11 – Óculos de realidade virtual Pisight da Sencics, [11].....	14
Figura 2.12 – Dispositivos para realidade virtual aumentada da Vuzix, [12].....	14
Figura 2.13 – Exemplo de imagem com realidade virtual aumentada, [13].....	14
Figura 3.1 – Configurações das Plataformas de Gough-Stewart, [14].....	16
Figura 3.2 – Plataforma de Gough-Stewart, [15] .....	17
Figura 3.3 – Caracterização dos sistemas de eixos e dos vértices ao nível das juntas universais da Plataforma de Gough-Stewart, [15].....	18
Figura 3.4 – Diagrama de blocos da cinemática inversa da Plataforma Gough-Stewart .....	19
Figura 3.5 – Bloco da cinemática inversa da Plataforma Gough-Stewart.....	19
Figura 3.6 – Bloco do modelo da aeronave Cessna-172 .....	21
Figura 3.7 – Diagrama representativo da estrutura interna simplificada do bloco do Cessna-172, adaptado de [17] .....	22
Figura 3.8 – Diagrama interno do bloco do modelo aerodinâmico .....	23
Figura 3.9 – Diagrama interno do bloco do modelo de propulsão .....	30

Figura 3.10 – Diagrama interno do bloco do modelo atmosférico.....	31
Figura 3.11 – Diagrama interno do bloco do modelo inercial.....	33
Figura 3.12 – Bloco do modelo da Terra.....	33
Figura 3.13 – Superfície terrestre, com representação do Geóide e da Elipsóide, [21] .....	35
Figura 3.14 – Diagrama interno do bloco do modelo da terra.....	35
Figura 3.15 – Bloco das equações do movimento referente ao modelo Cessna-172 .....	36
Figura 3.16 – Diagrama interno do bloco das equações de movimento.....	36
Figura 3.17 – Diagrama de blocos do modelo da aeronave Cessna-172 com controlo de estabilidade.....	40
Figura 3.18 – Bloco Modelo Cessna-172 .....	40
Figura 3.19 – Visão do piloto numa aeronave Cessna-172 ao aproximar-se da pista de aterragem do aeroporto internacional de São Francisco no <i>software FlightGear</i> <sup>®</sup> , [22] .....	42
Figura 3.20 – Bloco do <i>MatLab/simulink</i> <sup>®</sup> para a escrita de variáveis no <i>software FlightGear</i> <sup>®</sup> .....	43
Figura 3.21 – Painel do programa “fg_simulink.exe” de leitura da altura do solo em relação ao nível médio do mar .....	44
Figura 3.22 – Bloco de comunicação UDP em <i>MatLab/simulink</i> <sup>®</sup> de leitura da variável de altura do solo em relação ao nível médio do mar.....	44
Figura 3.23 – Representação do campo de visão do ser humano, [24] .....	45
Figura 3.24 – Tipo de paralaxe: a) zero, b) divergente, c) positiva, d) negativa, [8] .....	46
Figura 3.25 – Projecções <i>off-axis</i> e <i>on-axis</i> , [8] .....	48
Figura 3.26 – Ponto P e suas projecções ( $P_E$ e $P_D$ ) relativas aos centros CPE e CPD, [8]... ..	48
Figura 3.27 – Óculos de realidade virtual <i>eMagin</i> <sup>®</sup> no modelo z800, [26].....	50
Figura 4.1 – Diagrama de blocos “geral” do simulador da aeronave Cessna-172 .....	52
Figura 4.2 – Diagrama de blocos representando a comunicação utilizada entre dispositivos .....	52
Figura 4.3 – Interface humana: joystick de entrada do simulador, [27].....	54
Figura 4.4 – Diagrama interno do bloco DeadZone .....	54
Figura 4.5 – Diagrama de bloco utilizado para simulação do modelo matemático Cessna-172 .....	59

Figura 4.6 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Roll</i> do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente .....	60
Figura 4.7 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Roll</i> do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente....	61
Figura 4.8 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Pitch</i> do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente .....	61
Figura 4.9 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Pitch</i> do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente....	62
Figura 4.10 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Yaw</i> do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente .....	62
Figura 4.11 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Yaw</i> do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente....	63
Figura 4.12 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Throttle</i> do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente .....	63
Figura 4.13 – Resposta temporal do Modelo Cessna-172 para um movimento <i>Throttle</i> do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente .....	64
Figura 4.14 – Diagrama de bloco da cinemática inversa para comando plataforma electromecânica .....	65
Figura 4.15 – Sensor inercial <i>Xsens</i> <sup>®</sup> Mti, [33] .....	65
Figura 4.16 – Resposta temporal da plataforma electromecânica para um <i>Roll</i> do tipo quadrada. ....	66
Figura 4.17 – Resposta temporal do Modelo Cessna-172 para um <i>Roll</i> do tipo sinusoidal.	66
Figura 4.18 – Resposta temporal do Modelo Cessna-172 para um <i>Pitch</i> do tipo quadrada.	67
Figura 4.19 – Resposta temporal do Modelo Cessna-172 para um <i>Pitch</i> do tipo sinusoidal. ....	67
Figura 4.20 – Resposta temporal do Modelo Cessna-172 para um <i>Yaw</i> do tipo quadrada. .	68
Figura 4.21 – Resposta temporal do Modelo Cessna-172 para um <i>Yaw</i> do tipo sinusoidal.	68
Figura 4.22 – Monitorização do funcionamento do simulador Cessna-172.....	70
Figura 4.23 – Fotografia da plataforma electromecânica (a) e a imagem respectiva visualizada nos óculos de realidade virtual (b).....	70

## Índice de Tabelas

Tabela 3.1 – Parametrização do Bloco do modelo Cessna-172, [17].....	21
Tabela 3.2 – Variáveis de entrada do bloco do modelo Cessna-172.....	21
Tabela 3.3 – Variáveis de saída do bloco do modelo Cessna-172 .....	22
Tabela 3.4 – Variáveis de entrada do bloco do modelo terra .....	35
Tabela 3.5 – Variáveis de entrada do bloco do modelo terra .....	35
Tabela 3.6 – Variáveis de entrada do bloco equações de movimento.....	39
Tabela 3.7 – Variáveis de saída do bloco equações do movimento .....	39
Tabela 3.8 – Características de óculos de realidade virtual.....	51
Tabela 4.1 – Valores utilizados para o bloco DeadZone.....	55
Tabela 4.2 – Características dos movimentos da plataforma electromecânica, [15].....	55
Tabela 4.3 – Parâmetros do bloco Cinemática inversa plataforma 6DOF, [15].....	56
Tabela 4.4 – Estrutura da mensagem em ModBus TCP/IP ADU .....	58
Tabela 4.5 – Funções ModBus disponíveis no conversor de potência <i>Unidrive SP</i> , [28] ...	58
Tabela 4.6 – Estrutura do protocolo TCP/IP utilizado como suporte à rede <i>Ethernet</i> .....	58
Tabela 4.7 – Estrutura da mensagem <i>ModBus TCP/IP ADU</i> com indicação do valor dos <i>bytes</i> para o conversor de potência <i>Unidrive SP</i> .....	59
Tabela 4.8 – Condições iniciais para a simulação do modelo Cessna-172 .....	59

## 1 – Introdução

As plataformas do tipo Gough-Stewart caracterizam-se por disporem de seis graus de liberdade, de amplitude de movimentos considerável, e de uma frequência de operação, que contribuem para serem a escolha preferencial na aplicação de simuladores de movimentos, sejam eles ligados ao entretenimento ou a aplicações industriais. No ensino, é comum o seu uso no treino e instrução de pilotos de veículos, nomeadamente aéreos, terrestres e marítimos. A evolução das plataformas de Gough-Stewart surge da necessidade de se criarem simuladores cada vez mais sofisticados, com o intuito de completar a concepção, fabrico e optimização desses veículos. Torna-se, por esse motivo, numa ferramenta de engenharia, com custos reduzidos, indispensável na investigação.

Face às plataformas hidráulicas, a plataforma electromecânica apresenta a principal vantagem da portabilidade e da necessidade de um espaço de trabalho reduzido. Por razões económicas, é comum utilizarem-se, como accionamento, as máquinas eléctricas rotativas com adaptadores mecânicos para movimentos lineares.

A incorporação de ambiente de realidade virtual na plataforma electromecânica permite criar um simulador onde o utilizador pode interagir num ambiente virtual em tempo real, proporcionando sensações físicas associadas à dinâmica do veículo simulado, actuando, para o efeito, em dispositivos idênticos aos de comando do veículo real. Esta *interface* humana, utilizada em conjunto com óculos de realidade virtual, permite ao utilizador abstrair-se do mundo real e entrar num mundo virtual. Estes dispositivos, quando acoplados a sensores de movimento, permitem aos utilizadores uma maior liberdade de interacção. A realidade virtual é uma tecnologia indispensável em simuladores profissionais. Hoje em dia, com a evolução constante dos meios informáticos e do desenvolvimento de ferramentas de modelização, é possível desenvolverem-se simuladores com qualidade, a preço reduzido, onde a interactividade, associada à transmissão de sensações comparáveis aos sistemas reais, é um dos pontos fortes nestes sistemas.

### 1.1 – Objectivos

Pretende-se modelizar e controlar o sistema, constituído pelo modelo matemático de uma aeronave, aplicado à plataforma electromecânica, garantindo que esta responde às solicitações dinâmicas da aeronave simulada.

As entradas do sistema são fornecidas pelo utilizador através dos dispositivos de *interface* humana (*joystick*) que, associadas aos conversores electrónicos de potência, permitem o accionamento da plataforma electromecânica.

Neste sentido, pretende-se estudar e aplicar um modelo matemático de uma aeronave na cadeia de comando da plataforma electromecânica de Gough-Stewart, com o intuito de transmitir-se ao utilizador uma sensação de condução similar à de uma aeronave real. Para que exista coerência na simulação, torna-se necessário avaliar a trajectória da plataforma, imposta pela saída do modelo da aeronave, através da implementação de um sensor inercial.

Além da plataforma electromecânica, a interacção com o utilizador é, também, realizada com a incorporação de ambiente de realidade virtual, proporcionado pela utilização de óculos de realidade virtual e de *software* de simulação de aeronaves, que permita ao utilizador interagir em tempo real. Este trabalho tem, ainda, por objectivo a interligação de todos estes elementos, de forma a realizar um correcto funcionamento de todo o sistema.

Na figura 1.1 apresenta-se uma síntese dos elementos principais em forma de diagrama de blocos geral do simulador.

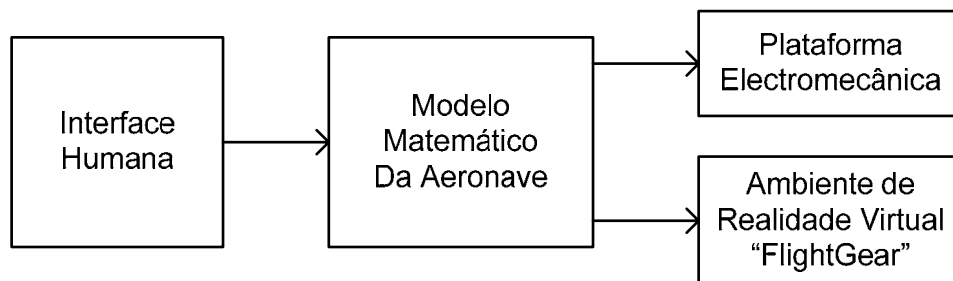


Figura 1.1 – Diagrama de blocos geral do simulador

## 1.2 – Motivação

A motivação deste trabalho tem em vista a obtenção do grau de mestre em Engenharia Electrotécnica e o desenvolvimento intelectual no que respeita ao percurso académico.

O tema deste trabalho vem no seguimento da investigação desenvolvida pelo autor na plataforma electromecânica. Por outro lado, por ser um tema em constante desenvolvimento na aérea dos simuladores de voo, nomeadamente no que diz respeito à utilização de sistemas de realidade virtual e de dispositivos de *interface* humana, com o objectivo de se alcançar maior interacção e realismo ao utilizador.

### 1.3 – Enquadramento do Trabalho

Esta dissertação vem no seguimento do trabalho desenvolvido no âmbito do projecto final de curso da Licenciatura em Engenharia Electrotécnica e Sistemas de Potência (Pré-Bolonha) intitulado “Emulação do comportamento de um helicóptero usando uma plataforma com sete graus de liberdade” [15], desenvolvido no Instituto Superior de Engenharia de Lisboa do Instituto Politécnico de Lisboa. O referido projecto final de curso teve como objectivos a colocação da plataforma electromecânica em funcionamento, o controlo dos actuadores electromecânicos, a implementação da *interface* humana, o desenvolvimento de um modelo matemático do manipulador e a implementação de modelos matemáticos de helicópteros em ferramenta de cálculo numérico.

O âmbito desta dissertação vem no seguimento destas temáticas e visa acrescentar inovações, nomeadamente a incorporação de ambiente de realidade virtual, a análise do comportamento da plataforma electromecânica com recurso de um sensor inercial, a aplicação de modelo matemático de uma aeronave (Cessna-172), o melhoramento do sistema de comunicação com a plataforma electromecânica e a interacção do utilizador com a plataforma electromecânica.

### 1.4 – Estrutura do Trabalho

Este trabalho encontra-se estruturado em cinco capítulos. Após este capítulo introdutório, o primeiro capítulo, segue-se o capítulo do estado da arte, ou seja, o segundo capítulo, onde se apresentam resumidamente factos históricos dos simuladores de voo e da realidade virtual.

No terceiro capítulo, é apresentado o projecto do sistema, onde se aborda a cinemática inversa da plataforma Gough-Stewart e se descreve o modelo matemático da aeronave Cessna-172, *software* de simulação gráfico e conceitos de realidade virtual.

No quarto capítulo, fundamenta-se a implementação do modelo matemático da aeronave Cessna-172 na plataforma Gough-Stewart para a simulação de voo, realizam-se simulações em ferramenta de cálculo numérico e valida-se a cinemática inversa da plataforma Gough-Stewart com o auxílio de um sensor inercial.

No último capítulo, apresentam-se as conclusões e perspectivas de desenvolvimento futuro. Os anexos A, B e C apresentam aspectos desenvolvidos no trabalho.

## 2 – Estado da Arte

### 2.1 – Simuladores de Voo

Os simuladores de voo desenvolveram-se à custa de aplicações militares, no entanto, a sua utilização tornou-se parte integrante da formação em companhias aéreas comerciais na década de 60, uma vez que não era sustentável o treino nas próprias aeronaves. Para a formação e aperfeiçoamento dos pilotos de aeronaves, tornou-se indispensável a utilização dos simuladores, nos quais os pilotos se familiarizam com os comandos da aeronave, sem grandes custos. A formação nos simuladores de voo teve grande sucesso em programas espaciais, [1].

Os primeiros simuladores eram sistemas constituídos por máquinas de fraca potência movidas ao longo de um terreno e controladas por um leme de direcção. Mais tarde, deram lugar a máquinas de maior potência, nos quais já existia o controlo de *Pitch*. Um sistema idêntico ao descrito, conhecido por Sistema de Pinguim, foi utilizado durante a Primeira Guerra Mundial no qual o piloto poderia ter a sensação de controlo de uma aeronave, [1].

Em 1910, surgiram outros dispositivos para testar protótipos de aeronaves, utilizando as aeronaves sustentadas por balões ou pórticos ferroviários. Nessa altura, o Professor Sanders constrói um dispositivo a partir de componentes de um avião montado numa junta universal, de forma a utilizar o vento como propulsor. Um exemplo de um destes dispositivos é apresentado na figura 2.1, o qual era constituído por duas metades de um barril, montados de forma a representar os controlos de *Pitch* e *Roll* de um avião.

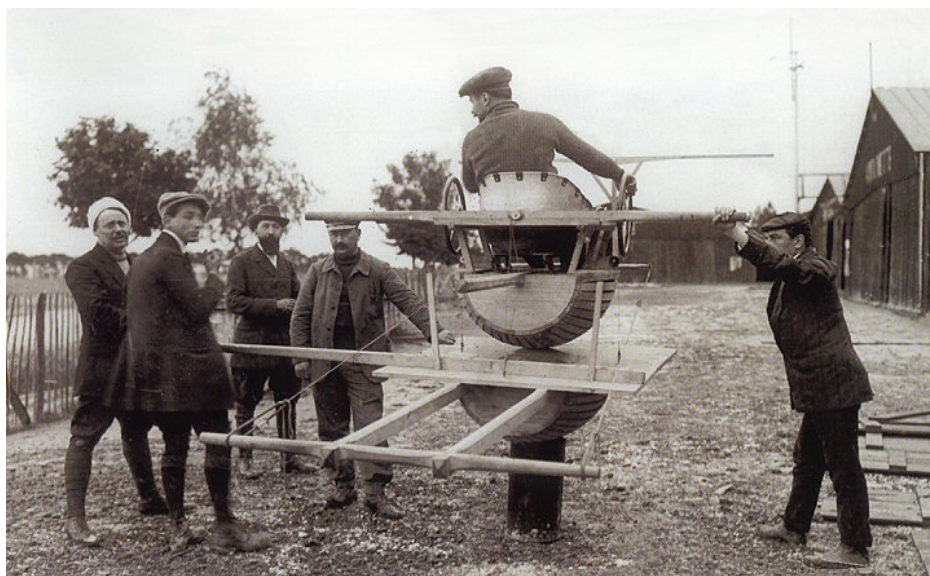


Figura 2.1 – Simulador de Antoniette em 1910, [1]

Durante a Primeira Guerra Mundial, muitos aparelhos foram inventados para auxiliar na selecção dos potenciais aviadores. Em 1915, surgiu um aparelho para medir o tempo de reacção do piloto, com possibilidade de gravação, onde a inclinação era produzida manualmente pelo examinador. Logo de seguida os operadores foram substituídos por sistemas actuadores mecânicos ou eléctricos ligados aos comandos do simulador, permitindo ao examinador colocar perturbações na altitude, [2].

Em 1917, Lender e Heidelberg construíram um sistema actuado a ar comprimido de forma a proporcionar movimentos de *Pitch*, *Roll* e *Yaw*, que, pela primeira vez, apresentava uma boa resposta para a velocidade simulada, incluindo também o ruído do motor e um sistema visual rudimentar. No entanto, em 1920, ficou reconhecida a necessidade de se formarem bons pilotos apenas com orientação por instrumentação, utilizando sistemas fixos, [1].

Na figura 2.2, apresenta-se um dos maiores sucessos, o simulador de Edwin Link. Patenteado em 1930, foi declarado, na época, como o sistema mais eficiente na formação aeronáutica, tendo sido ajustado para proporcionar ao piloto as sensações idênticas às verificadas na realidade e no qual foi, posteriormente, acoplada instrumentação que o tornou no simulador mais completo da época, [2].

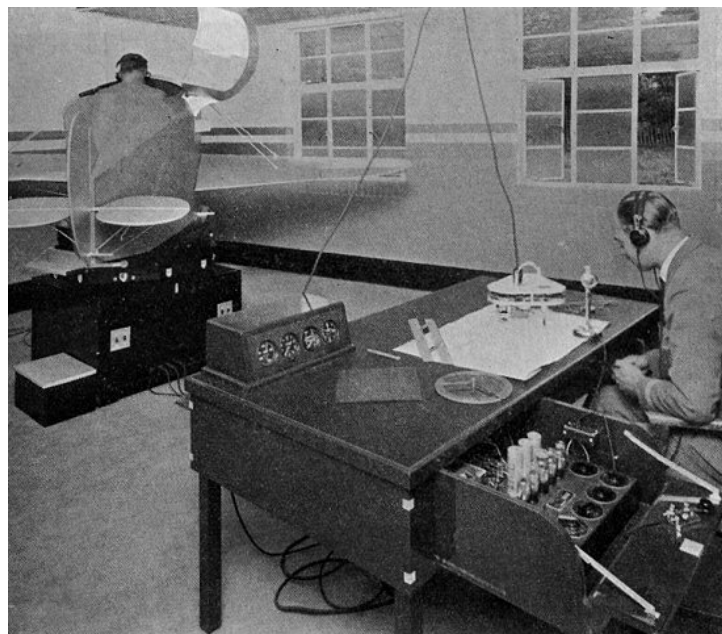


Figura 2.2 – Simulador de Edwin Link, [3]

Este simulador foi produzido em várias versões e vendido para diversos países, tais como, Inglaterra, Japão, França, Alemanha e na URSS. Em 1937, a American Airlines tornou-se a primeira companhia aérea mundial a comprar o simulador Link, para treinar e formar os seus pilotos, [3].

Com o desenvolvimento dos sistemas eléctricos e electrónicos, foi possível adicionar ao simulador de Link um sistema de comunicação e navegação por rádio, desenvolvido por R.C. Dehmel, na companhia Bell Telephone Laboratories.

A.E. Travis desenvolveu o simulador “Aerostructor”, constituído com base em sistemas operacionais eléctricos, visuais e de instrumentação. O sistema visual foi baseado numa repetição de filme tipo cinema, reproduzindo assim os movimentos de *Heave*, *Pitch* e *Roll*. Este simulador nunca foi produzido comercialmente, embora os Estados Unidos da América o tenham utilizado na Marinha, com algumas modificações, para treinar os pilotos a disparar contra aviões inimigos, [1].

No início da Segunda Guerra Mundial, os Estados Unidos da América e Aliados sentiram a necessidade de adquirir os simuladores de Link, com a finalidade de equipar as escolas de aviação para a formação de um grande número de pilotos, [3]. Na figura 2.3, apresenta-se uma imagem da companhia Link Aviation Devices.



Figura 2.3 – Construção dos simuladores “ANT-18 Basic Instruments Trainer”, [4]

Em 1939, os Britânicos solicitaram a Link a concepção de um simulador, de forma a melhorarem as suas capacidades de navegação astronómica. Este foi finalizado em 1941 e alguns dos modelos foram instalados nos Estado Unidos da América, [1].

Durante a Segunda Guerra Mundial, surgiram várias versões de simuladores, concebidos para aplicações muito específicas ligadas à aeronáutica. Mas, os principais avanços tecnológicos nesta matéria foram o desenvolvimento da electrónica e dos computadores analógicos, que permitiram resolver as equações de movimento de voo e

simular a resposta às forças aerodinâmicas, em vez da duplicação empírica dos seus efeitos, [2].

Após a Segunda Guerra Mundial, Curtiss e Wright com a Pan American Airways, constroem um simulador de um Boeing 377 Stratocruiser, que se tornou no primeiro simulador completo a pertencer a uma companhia aérea. Apresenta-se na figura 2.4 o simulador Boeing 377 de Curtiss-Wright, [1].

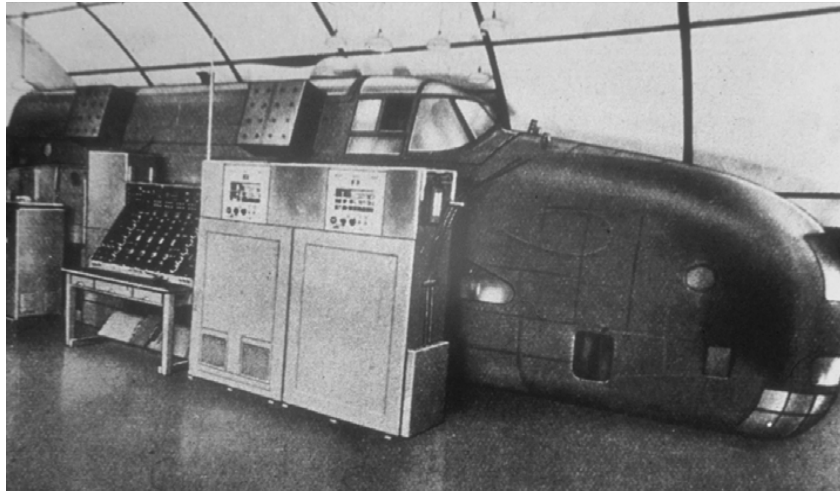


Figura 2.4 – Simulador Boeing 377 de Curtiss-Wright, [3]

A British Overseas Airway Corporation (BOAC), em 1947, decide adquirir o simulador do Boeing 377 Stratocruiser e o conhecimento da companhia Redifon em simuladores, solicitando a esta a construção de um simulador idêntico ao da Pan American. Facto que deu origem a um contrato entre a Redifon e Curtiss-Wright, no início de 1950.

Em 1950, a Pan American e a QANTAS foram as primeiras companhias aéreas a colocar em serviço os simuladores da aeronave B707, que eram, nessa época, os simuladores mais sofisticados do mundo. Porém, estes simuladores não tinham uma grande precisão, o que levou à utilização de um grande número de sistemas computacionais analógicos para se obter um melhor desempenho. Apesar dos esforços realizados, a fiabilidade nos grandes simuladores analógicos não conseguiram atingir uma vez que exigia grande manutenção diária. Com o desenvolvimento dos computadores digitais de segunda geração foi possível satisfazer os requisitos de capacidade de variáveis de entrada/saída, velocidade de processamento e de custos associados na aplicação nos simuladores de voo, [1].

Em 1958, a Redifon recebeu um contrato da BOAC para a produção de um simulador Comet IV. Inicialmente, este sistema foi construído com um grau de liberdade para realizar o movimento de *Pitch*. Só mais tarde, e de forma progressiva, chegaram aos seis graus de

liberdade considerados essenciais para fornecer as acelerações necessárias à simulação de aviões como o B747. Para aumentar as amplitudes máximas dos movimentos, surgiram projectos inovadores, recorrendo à construção de grandes estruturas metálicas e a sistemas hidráulicos como actuadores, causando muitos problemas a nível de manutenção, [1]. Na figura 2.5 apresenta-se o simulador Comet IV.



Figura 2.5 – Simulador Comet IV da Redifon, [5]

Em 1960, a Sylvania Corporation construiu um dispositivo digital de formação de voo desenvolvido pela Universidade da Pensilvânia num programa de investigação que teve início em 1950, [1]. Por outro lado Link, desenvolve o seu próprio modelo de simulação em tempo real, baseando-se no computador digital “Mark 1”. Este sistema utilizava três processadores digitais em paralelo, obtendo assim, novamente, sucesso perante as companhias aéreas e no meio militar dos Estados Unidos da América, [2].

Uma grande parte da investigação sobre simulação de movimentos foi realizada pela NASA (National Aeronautics and Space Administration) durante a preparação para o desembarque do Módulo Lunar, representando um desafio invulgar. Dessa investigação, a NASA elaborou um conjunto de equações de movimento que foram utilizadas, posteriormente, pela maioria dos fabricantes de simuladores, [1].

Desde o início da utilização dos simuladores de voo, considerava-se desejável a reprodução de imagens com o intuito de tornar as simulações mais realistas. Mesmo no simulador de Link, foi acoplado um ecrã panorâmico, onde eram projectadas imagens para dar a ilusão de pilotar uma aeronave, [2].

Na década de 50, foi utilizada uma técnica de projecção através de um ponto de luz (*point-light source projection*) e de técnicas de utilização de sombras (*shadowgraph*). Um desses métodos foi produzido pela Giravions Dorand França e por Shorts de Belfast. Nos Estados Unidos da América foi também utilizada essa técnica, mas não teve evolução por apresentar bastantes deficiências. Entretanto, Link desenvolve um sistema de visualização óptica com película de filme, conhecido na altura por sistema VAMP. Apesar de apresentar boa qualidade de imagem, os utilizadores não tinham liberdade de voo, podendo voar apenas dentro da área da película de filme, [1]. Na figura 2.6, apresenta-se um sistema de visualização acoplado a um simulador.

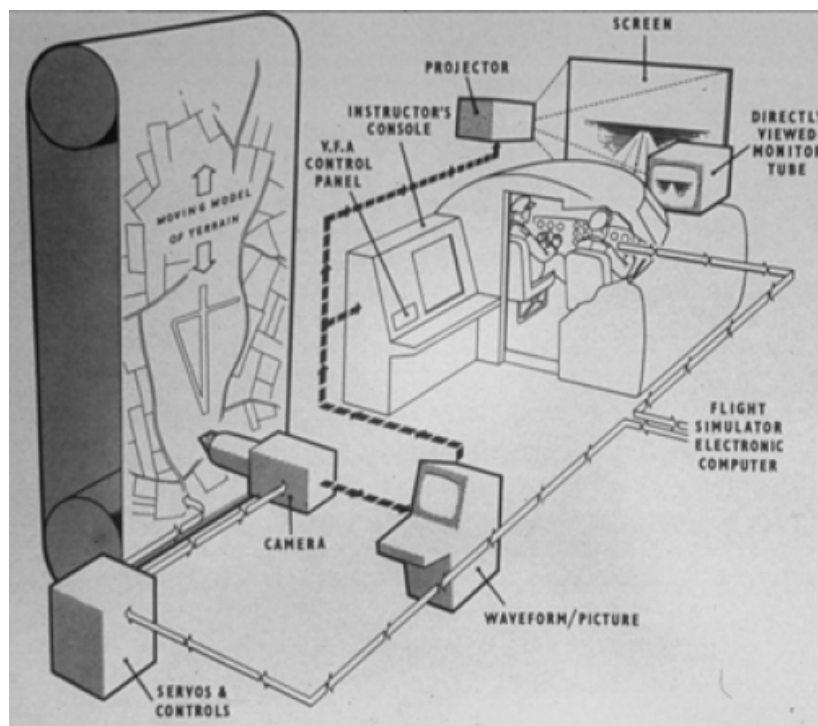


Figura 2.6 – Sistema de visualização acoplado a um simulador, [1]

Com o desenvolvimento da televisão foi possível criar um circuito fechado, onde uma câmara se movimentava à frente de uma grande tela e as imagens captadas eram projectadas à frente do *cockpit* do simulador permitindo movimentos de *Roll*, *Pitch* e *Yaw*. Estes sistemas eram monocromáticos e foram desenvolvidos por Curtiss, Wright e Link na Inglaterra pela Link Division of General Precision (Antiga Air Trainer Link), [1]

Em 1962, surgiram sistemas a cores, introduzidos pela empresa Redifon, nos quais o antigo sistema de correias foi substituído por um sistema rígido onde a câmara andava sobre um terreno. Assim, poderiam recriar países e cidades em miniatura com iluminação para simulações nocturnas, mais realistas. Este sistema foi utilizado até 1970, [1].

Os primeiros sistemas de visualização recorrendo a sistemas computacionais foram produzidos pela General Electric (EUA) para programas espaciais. A primeira versão deste sistema reproduzia apenas um terreno plano e mais tarde objectos a três dimensões.

Os sistemas de visualização para os simuladores foram-se desenvolvendo em paralelo com os sistemas computacionais, com preocupação de aumentar o campo de visão do piloto. Por exemplo, a QANTAS instalou seis ecrãs para a reprodução de imagem, como se fossem as janelas da aeronave, [1].

Em 1970, surge a necessidade de algumas companhias aéreas criarem a associação IAFSTA (International Airline Flight Simulator Technical Association), com o objectivo de discutir os problemas relacionados com os seus simuladores e partilhar experiências entre eles. Mais tarde, esta associação recebeu um convite para se juntar a IATA (Internacional Air Transport Association) para formar o comité técnico FSTSC (Flight Simulator Technical Sub-Committee) no âmbito dos simuladores de voo, cuja primeira reunião foi realizada a 1973 em Denver, EUA. Depois da associação entre a IAFSTA e a IATA, verificou-se que os simuladores de voo atingiram credibilidade para os pilotos e para as entidades reguladoras. Normas internacionais foram criadas para os simuladores de voo e para a indústria de construção de simuladores, [1]. A figura 2.7 apresenta o simulador de voo actual da companhia Aérea QANTAS.



Figura 2.7 – Simulador de voo actual da companhia Aérea QANTAS, [6]

## 2.2 – Realidade Virtual

A realidade virtual não é um conceito recente e a sua aplicação nos simuladores de voo teve sempre grande importância. No entanto, os custos envolvidos eram de tal forma elevados que essa tecnologia era praticamente proibida durante muitos anos. Com o desenvolvimento das tecnologias computacionais (*hardware e software*), a realidade virtual começou a ser viável e a sua aplicação nos simuladores de voo a ter preços acessíveis, dando aos utilizadores desta tecnologia a sensação de estar noutra lugar, [7].

A realidade virtual é considerada uma ciência que engloba o conhecimento de várias áreas, nomeadamente informática, electrónica, robótica e “médica”. Permite sistemas que possam conter características imersivas e interactivas para simular ambientes reais, estimulando os sentidos do utilizador, especialmente visuais, auditivos, tácteis, motores e olfactivos, através de dispositivos próprios, [8].

No fim da Segunda Guerra Mundial, a realidade virtual começou a ser desenvolvida pela força aérea dos Estados Unidos da América. Paralelamente à indústria militar, a indústria do entretenimento também contribuiu para o desenvolvimento de sistemas de realidade virtual, sobretudo na área cinematográfica. Nessa altura, Norton L. Heilig desenvolve um sistema de entretenimento designado de Sensorama (patenteado a 1962), que era um sistema que exibia imagens a três dimensões combinando som estéreo, vibrações mecânicas, aromas e vento através de ventiladores, com o intuito de proporcionar ao utilizador experiências sensoriais diferentes. Na figura 2.8, apresenta-se o Sensorama de Norton L. Heilig.



Figura 2.8 – Sensorama de Norton L. Heilig, [7]

Ainda durante o final da Segunda Guerra Mundial, Philco desenvolve também um sistema de duas câmaras remotas e um protótipo de capacete de realidade virtual com monitores que permitiam aos utilizadores sentirem-se imersos noutra ambiente.

Em 1965, Ivan Sutherland apresentou à comunidade científica a ideia de desenhar objectos sobre o próprio ecrã do computador utilizando uma caneta óptica, começando nessa altura os desenvolvimentos dos sistemas gráficos computacionais. Sutherland foi um dos propulsores da indústria CAD (Computer-aided design) e desenvolveu o primeiro vídeo capacete, actualmente conhecido por HMD (Head Mount Display), com funcionalidade total de gráficos, no projecto The Ultimate Display. Este sistema, desenvolvido na Universidade de Utah, dava ao utilizador a possibilidade de observar os diversos lados de um cubo, através do movimento da cabeça, utilizando uma estrutura flutuante e fio de arame. Na figura 2.9, apresenta-se o projecto de capacete de realidade virtual desenvolvido por Ivan Sutherland.

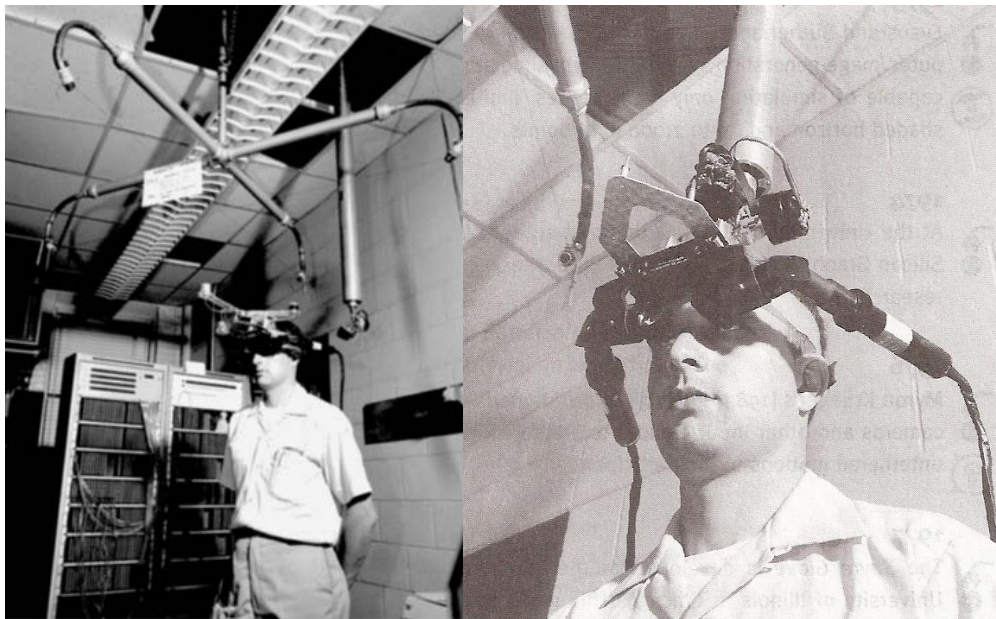


Figura 2.9 – Capacete de realidade virtual de Ivan Sutherland, [9]

Por outro lado, Myron Krueger, na universidade de Wisconsin, combina sistemas de vídeo com sistemas computacionais alcançando a realidade virtual e, em 1975, desenvolveu o Videoplance, que consistia na captura de imagens de alguns utilizadores e projectava essas imagens numa grande tela. Assim, os utilizadores podiam interagir uns com os outros, bem como, com objectos recriados e projectados nessa mesma tela, [7].

Em 1982, Thomas Fueness demonstrou, para a Força Aérea Americana, o VCASS (Visually Coupled Airborne System Simulator) que utilizava computadores e capacetes de vídeo interligados para representar imagens a três dimensões do *cockpit* de um avião.

Assim, os utilizadores podiam aprender com maior realidade, conferido pelos seis graus de liberdade, pela grande qualidade de imagem e pelo processamento rápido de imagens complexas, naquele que era conhecido na época como o *Super Cockpit*, [10]. Na figura 2.10, apresenta-se o projecto VCASS de Thomas Fueness.

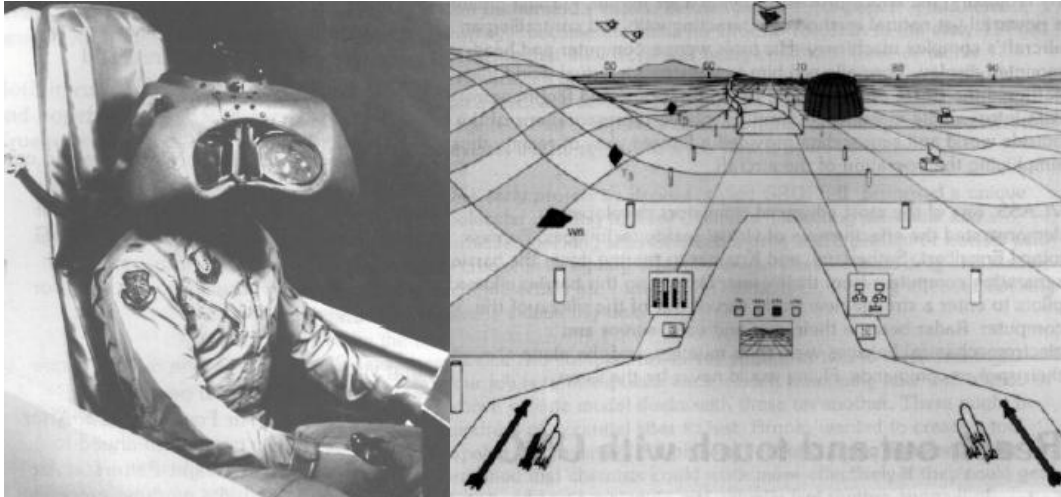


Figura 2.10 – Projecto VCASS de Thomas Fueness, [10]

Com o aparecimento dos monitores LCD (Liquid Crystal Display), Michael McGreevy trabalhou num projecto designado de VIVED (Virtual Visual Environment Display), em 1984 na NASA, no qual era possível gerar imagens estereoscópicas a um preço mais acessível, quando comparado com o VCASS. Scoot Fisher envolveu-se neste projecto em 1985, com o objectivo de incluir luvas para realidade virtual (*DataGlove*), reconhecimento de voz, sistema de som tridimensional e sensores de *feedback* táctil. Essa luva, desenvolvida em 1985 por Thomas Zimmerman, possuía a capacidade de detectar os movimentos e a posição dos dedos da mão, sendo mais tarde comprada pela NASA para o projecto VIVED, [7].

Em 1986, a NASA já tinha um sistema com ambiente virtual onde os utilizadores podiam, por um lado, dar comandos de voz, ouvir vozes sintetizadas e dispor de som tridimensional, e por outro, podiam manipular directamente objectos virtuais através do movimento das mãos. Este projecto possibilitou a comercialização de um conjunto de tecnologias a preços acessíveis, permitindo a empresas de *software* e a investigadores desenvolverem os seus próprios projectos de realidade virtual no mundo inteiro. Desde aí, muitos produtos e serviços foram desenvolvidos e comercializados, tornando os sistemas de realidade virtual numa prática comum em aplicações profissionais, militares e lúdicas, [7].

Os dispositivos mais comuns, quando se pensa em realidade virtual, são os óculos de realidade virtual (simples e leves) ou HMD (características superiores, maior complexidade

e envolvimento), sendo estes os mais populares por serem os dispositivos de saída de dados que mais isolam o utilizador do mundo real. Na figura 2.11 apresentam-se os óculos de realidade virtual Pisight da Sensics.



Figura 2.11 – Óculos de realidade virtual Pisight da Sensics, [11]

Nos últimos anos, a maior parte dos óculos de realidade virtual incluem sensores de movimento da cabeça do utilizador, tendo alguns a possibilidade de rastrear a posição dos olhos tornando-o, além de dispositivo de saída (imagens), também num dispositivo de entrada (posição da cabeça e olhos), enviando os dados para o computador, processando-os e interagindo com o ambiente gráfico.

Actualmente, tem existido bastante investigação em realidade virtual aumentada, isto é, visualização directa do mundo real com a inserção simultânea de elementos virtuais criados computacionalmente. Na figura 2.12, apresentam-se dispositivos para realidade virtual aumentada e, na figura 2.13, apresenta-se, como exemplo, o resultado de uma imagem com realidade virtual aumentada.



Figura 2.12 – Dispositivos para realidade virtual aumentada da Vuzix, [12]



Figura 2.13 – Exemplo de imagem com realidade virtual aumentada, [13]

## 3 – Projecto do Sistema

### 3.1 – Introdução

A construção de um simulador implica necessariamente o estudo prévio de todos os seus componentes. Neste sentido, pretende-se, neste capítulo, apresentar o estudo sintetizado do modelo cinemático da plataforma de Gough-Stewart, o estudo do modelo matemático da aeronave e da *interface* com o utilizador.

Para simular movimentos recorrendo à uma plataforma de Gough-Stewart é necessário analisar as suas características, de forma a verificar a compatibilidade entre o simulador e os movimentos que se pretendem simular. Dado que este tipo de plataforma possui seis graus de liberdade, existem várias hipóteses na escolha do modelo a simular. Neste estudo pretende-se simular a aeronave Cessna-172, cujo modelo matemático foi retirado de um exemplo apresentado na livraria do *software MatLab/Simulink*<sup>®</sup>, [18], na *toolbox AeroSim*<sup>®</sup>, [17].

Para um correcto funcionamento é necessário estudar todos os elementos constituintes do modelo matemático, realizando-se por vezes adaptações, com vista a melhorar as sensações transmitidas ao utilizador, e, na plataforma de Gough-Stewart, há que avaliar a sua resposta às solicitações impostas pela saída do modelo matemático da aeronave, visando melhorar o seu desempenho e aproximando-o da aeronave real.

Os movimentos da plataforma de Gough-Stewart, em conjunto com a visualização do ambiente virtual, estabelecem a saída do sistema. O ambiente virtual é realizado pelo *software FlightGear*<sup>®</sup>, [19], um videojogo gratuito e de código aberto, desenvolvido com o apoio de profissionais voluntários de todo o mundo, que tem sido utilizado em entretenimento e na investigação a nível académico.

### 3.2 – Plataforma de Gough-Stewart

Uma plataforma de Gough-Stewart consiste num manipulador paralelo com características bem definidas. Para criar o movimento e a sustentação do simulador é utilizado um manipulador paralelo com seis graus de liberdade, uma vez que permite três movimentos de translação e três movimentos de rotação num espaço tridimensional. Os movimentos de translação segundo o eixo “x” denominam-se *Surge*, representado por “px”, segundo o eixo “y” denominam-se *Sway*, representado por “py”, segundo o eixo “z”

denominam-se *Heave*, representado por “pz”. Os movimentos de rotação em torno de “x” denominam-se *Roll*, representado por “ $\alpha$ ”, em torno de “y” denominam-se *Pitch*, representado por “ $\beta$ ” e em torno de “z” denominam-se *Yaw*, representado por “ $\gamma$ ”. Estas estruturas são constituídas por um corpo rígido superior, onde se realizam os movimentos de rotação e translação mencionados e um corpo rígido inferior que consiste na base da plataforma. O movimento do corpo rígido superior é realizado à custa de juntas prismáticas designadas por pernas. Para satisfazer a união entre o corpo rígido superior e as pernas e entre pernas e o corpo rígido inferior são utilizadas juntas universais.

Nas plataformas de Gough-Stewart, existem várias formas de ligação entre o corpo rígido superior e o corpo rígido inferior. São utilizadas seis pernas que podem assumir uma configuração do tipo 3-3, 3-6 ou 6-6, em que o primeiro algarismo representa o número de vértices do corpo rígido superior e o segundo algarismo representa o número de vértices do corpo rígido inferior. Para facilitar a obtenção do modelo matemático da plataforma de Gough-Stewart é desejável que a estrutura apresente características simétricas no corpo rígido superior e no corpo rígido inferior, sendo estas normalmente triangulares ou hexagonais. Na figura 3.1, ilustram-se as configurações das plataformas de Gough-Stewart com seis pernas.

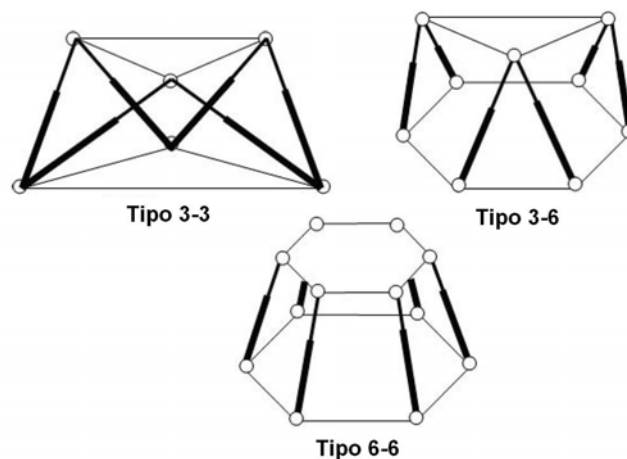


Figura 3.1 – Configurações das Plataformas de Gough-Stewart, [14]

A configuração da plataforma de Gough-Stewart utilizada neste estudo é do tipo 6-6. De referir que, de entre as configurações possíveis, o tipo 6-6 é o preferencial para este tipo de aplicações, segundo [14].

A posição e orientação do topo dependem do comprimento das pernas, onde o estudo da cinemática traduz matematicamente a relação entre o corpo rígido superior e o corpo rígido inferior. O estudo da cinemática consiste na descrição do movimento de um manipulador sem contabilizar as forças e momentos de inércia que causam esse movimento.

O modelo da cinemática directa permite obter a posição e orientação do corpo rígido superior em função das variáveis das juntas (prismáticas e universais), ou seja, assume como entrada do modelo a distância das pernas, obtendo na saída a posição e orientação do corpo rígido superior em relação a um ponto de referência. Por outro lado, o modelo da cinemática inversa permite determinar as variáveis das juntas que correspondem a uma determinada posição e orientação do corpo rígido superior, ou seja, assume como entrada a posição e orientação do corpo rígido superior em relação a um ponto de referência, obtendo na saída a distância das pernas.

Os movimentos de translação e de rotação da aeronave simulada são referenciados à terra, e reproduzidos na interface gráfica do videojogo *FlightGear*<sup>®</sup>, enquanto na plataforma de Gough-Stewart apenas se utilizam os movimentos de rotação referenciados ao centro da base de sustentação desta. Uma vez que se pretende comandar a plataforma de Gough-Stewart, considerando como entrada os movimentos de rotação, recorre-se ao estudo da cinemática inversa.

Considera-se que, para o estudo da cinemática inversa, as juntas universais são ideais, por terem os centros de rotação bem definidos e inalteráveis. De igual modo, considera-se que as juntas prismáticas têm movimentos perfeitos, que todo o sistema não apresenta folgas mecânicas e que as posições dos centros das juntas universais são conhecidas.

Na figura 3.2, ilustra-se a plataforma de Gough-Stewart com a indicação das distâncias que são conhecidas e utilizadas para o cálculo da cinemática inversa.

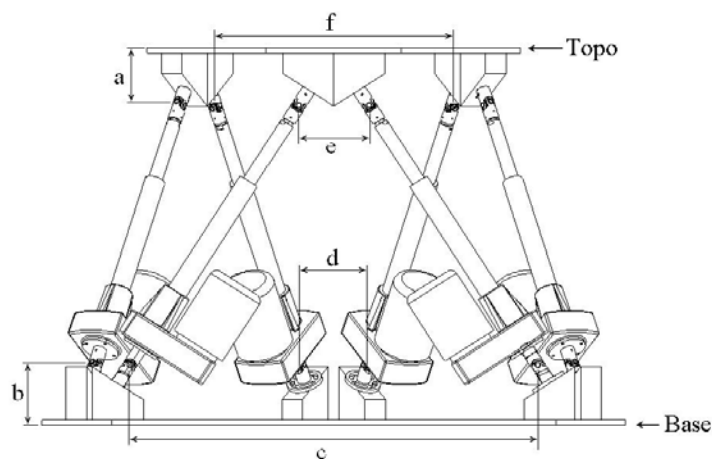


Figura 3.2 – Plataforma de Gough-Stewart, [15]

Assim, para o cálculo da cinemática inversa segundo [16] e [15], considera-se um sistema de eixos de referência ( $X, Y, Z$ ), colocado no centro geométrico da base da plataforma, e um sistema de eixos móvel ( $p_x, p_y, p_z$ ), colocado no centro geométrico do topo. Na figura 3.3, apresenta-se um esboço da localização dos sistemas de eixos

mencionados e caracterizam-se os vértices da plataforma de Gough-Stewart ao nível das juntas universais.

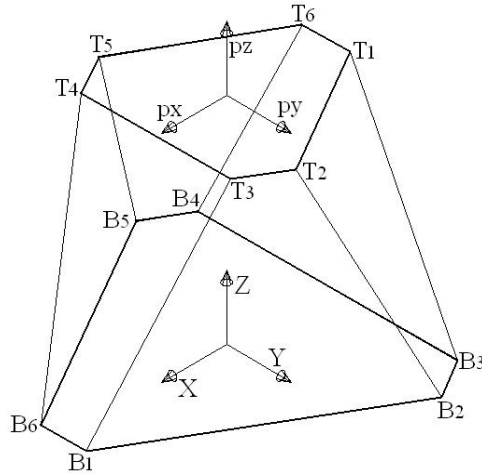


Figura 3.3 – Caracterização dos sistemas de eixos e dos vértices ao nível das juntas universais da Plataforma de Gough-Stewart, [15]

Tendo em conta a posição dos vértices do corpo rígido inferior em relação ao sistema de eixos de referência, a posição dos vértices do corpo rígido superior em relação ao sistema de eixo móvel e conhecendo a transformação homogénea que relaciona os dois sistemas de eixos, calcula-se a distância das seis juntas prismáticas. Apresenta-se como resultado as equações gerais da cinemática inversa ao nível das juntas universais, que corresponde à diferença vectorial entre os vértices de cada perna em relação ao centro geométrico do corpo rígido inferior, considerando o respectivo módulo do vector para obter a distância de cada junta prismática, indicado nas equações (3.1), [16].

$$\left\{ \begin{array}{l} L_1 = \sqrt{(X_{T3} - X_{B1})^2 + (Y_{T3} - Y_{B1})^2 + (Z_{T3} - Z_{B1})^2} \\ L_2 = \sqrt{(X_{T2} - X_{B2})^2 + (Y_{T2} - Y_{B2})^2 + (Z_{T2} - Z_{B2})^2} \\ L_3 = \sqrt{(X_{T1} - X_{B3})^2 + (Y_{T1} - Y_{B3})^2 + (Z_{T1} - Z_{B3})^2} \\ L_4 = \sqrt{(X_{T6} - X_{B4})^2 + (Y_{T6} - Y_{B4})^2 + (Z_{T6} - Z_{B4})^2} \\ L_5 = \sqrt{(X_{T5} - X_{B5})^2 + (Y_{T5} - Y_{B5})^2 + (Z_{T5} - Z_{B5})^2} \\ L_6 = \sqrt{(X_{T4} - X_{B6})^2 + (Y_{T4} - Y_{B6})^2 + (Z_{T4} - Z_{B6})^2} \end{array} \right. \quad (3.1)$$

Segundo [15], conhecendo as equações (3.1) da cinemática inversa ao nível das juntas universais é necessário adaptar estas à plataforma Gough-Stewart, considerando como entrada um novo sistema de eixos (x, y, z) no centro geométrico do topo, uma vez que corresponde à localização do utilizador do simulador. Tendo em conta as características geométricas da plataforma de Gough-Stewart e as equações (3.2) realiza-se essa adaptação.

$$\begin{cases} pz = z - a \cdot \cos \alpha \cdot \cos \beta - b \\ px = x + a \cdot \cos \alpha \\ py = y + a \cdot \sin \alpha \end{cases} \quad (3.2)$$

Recorrendo às equações (3.1) e (3.2) apresenta-se o diagrama de blocos em *MatLab/simulink*<sup>®</sup> da cinemática inversa da plataforma de Gough-Stewart, representado na figura 3.4.

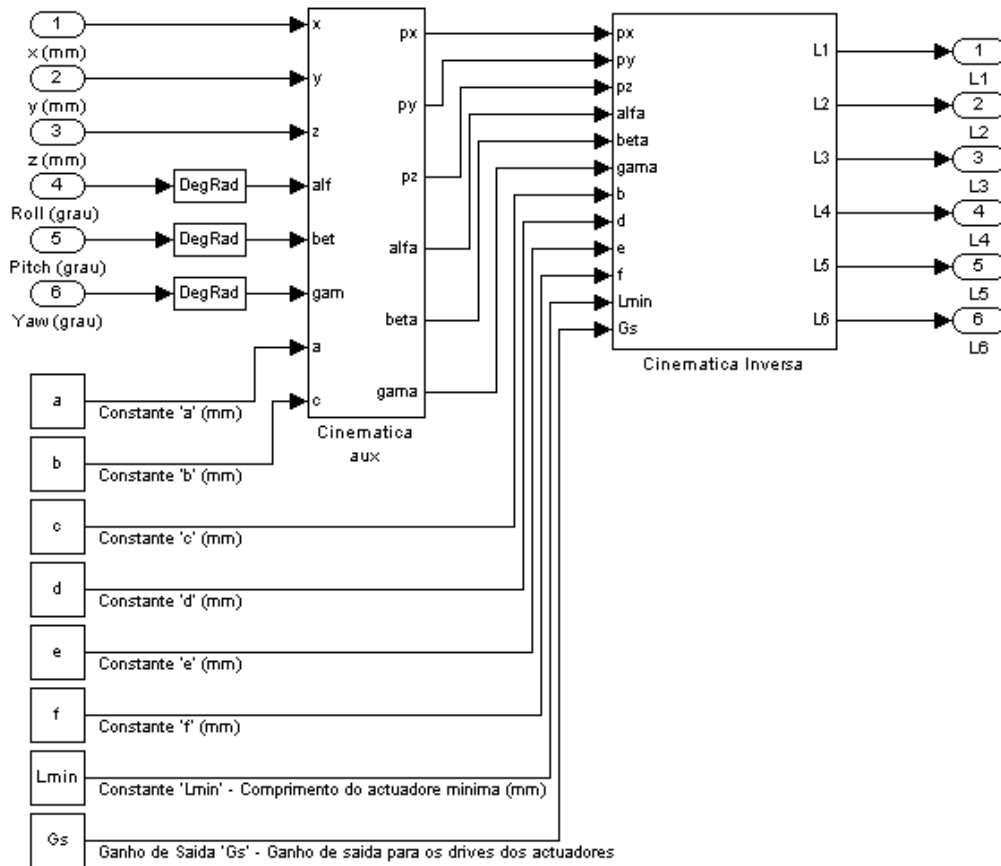


Figura 3.4 – Diagrama de blocos da cinemática inversa da Plataforma Gough-Stewart

De forma a simplificar o modelo do sistema em *MatLab/simulink*<sup>®</sup>, foi criado um subsistema do diagrama de blocos da cinemática inversa da plataforma de Gough-Stewart, apresentado na figura 3.5.

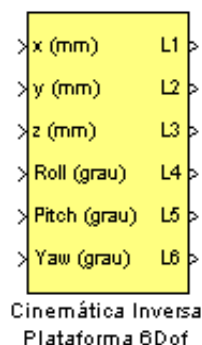


Figura 3.5 – Bloco da cinemática inversa da Plataforma Gough-Stewart

O diagrama de blocos desenvolvido contempla a possibilidade de parametrização das dimensões físicas da plataforma de Gough-Stewart, conhecendo as dimensões indicadas na figura 3.2, permitindo assim a possibilidade desta aplicação ser utilizada em outras plataformas de Gough-Stewart.

Para as juntas prismáticas das plataformas de Gough-Stewart podem ser utilizados vários dispositivos de accionamento. Neste caso, são utilizados actuadores electromecânicos, constituídos por máquinas eléctricas assíncronas rotativas com adaptação mecânica de movimento rotativo para movimento linear.

Introduziu-se um ganho ajustável de saída para que este diagrama de blocos do modelo da cinemática inversa possa ser utilizado noutros tipos de accionamentos. Foi considerado como parametrização o comprimento mínimo dos actuadores electromecânicos, para que a referência de posição dos conversores de potência fosse zero no comprimento mínimo.

Nos capítulos seguintes, a plataforma electromecânica de Gough-Stewart será designada apenas por plataforma electromecânica por questão de simplificação de linguagem.

### 3.3 – Modelo Matemático da Aeronave

Um modelo matemático consiste numa representação aproximada de um sistema físico real, descrito por meio de equações matemáticas. Para se construir um simulador de uma aeronave é necessário conhecer o modelo matemático dessa aeronave.

A *toolbox Aerosim*<sup>®</sup> para o *MatLab/simulink*<sup>®</sup> consiste numa biblioteca de utilitários que permite o rápido desenvolvimento de modelos não lineares dinâmicos de aeronaves com seis graus de liberdade. Existem alguns blocos predefinidos de modelos de aeronaves que permitem o rápido desenvolvimento e a demonstração das funcionalidades desta *toolbox*, [17]. Dentro dos modelos disponíveis escolheu-se o modelo da aeronave Cessna-172 para aplicar no simulador. Este modelo, que foi construído tendo em consideração elementos da base de dados do *software FlightGear*<sup>®</sup>, na versão v0.98, utiliza, como propulsor, um motor de combustão interna acoplado a uma hélice de *pitch* fixo e dois tanques de combustíveis distribuídos pelas asas. Na figura 3.6, apresenta-se o bloco do modelo da aeronave Cessna-172 disponível na livraria da *toolbox Aerosim*<sup>®</sup>, [17]

A utilização do bloco do modelo Cessna-172 requer uma parametrização de forma a definir o modelo da aeronave e as condições iniciais para a simulação. Apresenta-se, na tabela 3.1, os parâmetros de configuração do bloco do modelo Cessna-172.

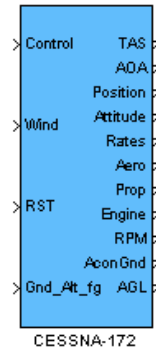


Figura 3.6 – Bloco do modelo da aeronave Cessna-172

Tabela 3.1 – Parametrização do Bloco do modelo Cessna-172, [17]

Designação	Parâmetros
Aeronave a simular	'c172p'
Directoria do software <i>FlightGear</i> <sup>®</sup> v0.98	'c:\Programas\flightgear098\data'
Velocidade inicial em coordenadas geográficas	$[V_N \ V_E \ V_D]$ (rad/s)
Ângulos iniciais de rotação da aeronave em relação ao solo	$[e_0 \ e_x \ e_y \ e_z]$ (rad)
Posição inicial em coordenadas geográficas	[lat Lon h] (rad rad m)
Velocidade inicial do motor de combustão interna	$vm_i$ (rad/s)
Nível de combustível inicial	$nc_i$ [0 100] (%)
Frequência de amostragem para a simulação em <i>MatLab/simulink</i> <sup>®</sup> .	$F_S$ (s)

Na tabela 3.2, apresentam-se as variáveis de entrada do bloco do modelo Cessna-172, onde foi contemplado uma entrada auxiliar com a informação da altura do solo em relação ao nível médio do mar (*Gnd\_alt\_fg*) nas coordenadas geométricas actuais da aeronave durante a simulação fornecida pelo *software FlightGear*<sup>®</sup>.

Tabela 3.2 – Variáveis de entrada do bloco do modelo Cessna-172

Designação das Variáveis		Entrada
<i>Control</i> (Controlo)	Flap	0 ou 1
	Elevador (Pitch)	[-1 1]
	Aileron (Roll)	[-1 1]
	Rudder (Yaw)	[-1 1]
	<i>Thottle</i> (acelerador)	[-1 1]
	<i>Mixture</i> (ar/combustível)	[-1 1]
	<i>Ignition</i> (ignição)	0 ou 1
<i>Wind</i> (Velocidade do vento em coordenadas geográficas)		$[W_N \ W_E \ W_D]$ (rad/s)
<i>RST</i> (Limpar registos de memória)		0 ou 1
<i>Gnd_Alt_fg</i> (Altura do solo em relação ao nível médio do mar)		<i>Gnd_Alt_fg</i> (m)

Na tabela 3.3, apresentam-se as variáveis de saída do bloco do modelo Cessna-172. Na simulação, na plataforma de Gough-Stewart, são utilizados os ângulos de Euler da aeronave em relação ao solo. Para a visualização através do videojogo *FlightGear*<sup>®</sup> são utilizados os mesmos ângulos de Euler e a posição, em coordenadas geográficas.

Tabela 3.3 – Variáveis de saída do bloco do modelo Cessna-172

Designação das Variáveis	Saída
TAS (Velocidade da aeronave)	tas (m/s)
AOA (Ângulo de ataque)	aoa (rad)
Position (Posição em coordenadas geográficas)	[lat Lon h] (rad rad m)
Attitude (Ângulos de Euler da aeronave em relação ao solo)	attitude (rad)
Rates (Velocidade angular em relação ao solo)	rates (rad/m)
Aero (Coeficientes aerodinâmicos)	-
Prop (Coeficientes da hélice)	-
Eng (Coeficientes do motor de combustão)	-
RPM (Velocidade de rotação do motor de combustão)	vm (rpm)
AconGrd (Contacto da aeronave com o solo)	0 ou 1
AGL (Altura em relação ao solo)	agl (m)

Internamente, o bloco do modelo Cessna-172 é constituído pelo modelo aerodinâmico, modelo do sistema de propulsão, modelo inercial, modelo atmosférico, modelo da terra e pelas equações do movimento da aeronave. Na figura 3.7, apresenta-se um diagrama referente à estrutura interna do bloco do modelo Cessna-172.

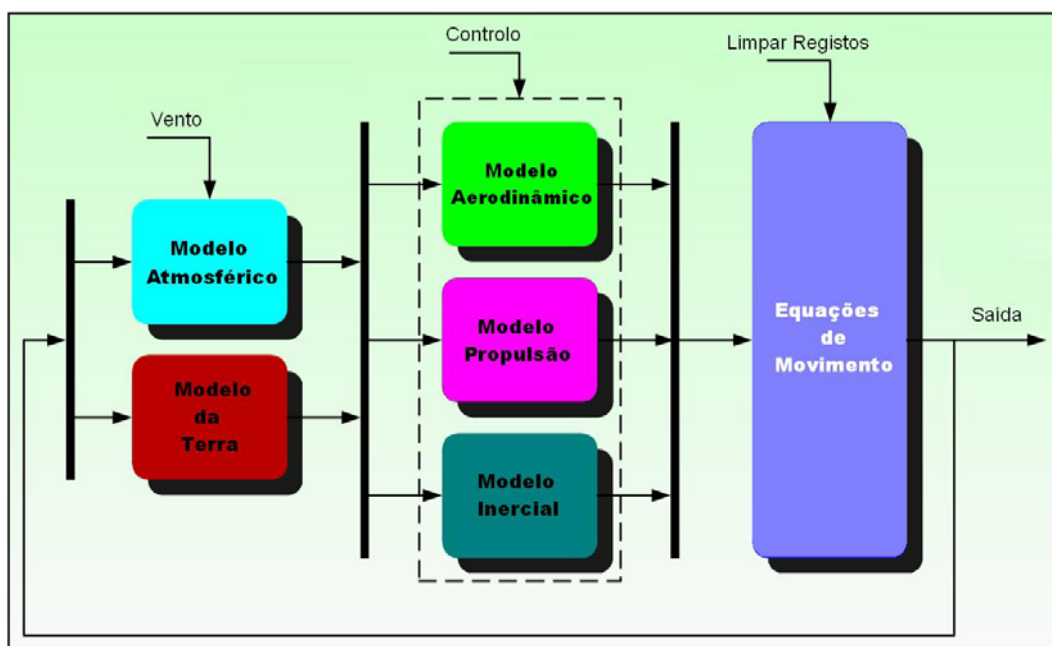


Figura 3.7 – Diagrama representativo da estrutura interna simplificada do bloco do Cessna-172, adaptado de [17]

### 3.3.1 – Descrição do Modelo Aerodinâmico

O modelo aerodinâmico permite estudar as forças aplicadas à aeronave durante o seu deslocamento. Consideram-se a força de sustentação, as forças de tracção e as forças de resistência ao movimento. Dependendo das características da aeronave, estas forças têm consequências diferentes, uma vez que dependem das condições atmosféricas, da localização da aeronave em cada instante de voo e das ordens de comando do piloto.

O modelo aerodinâmico é constituído pelos seguintes blocos: Aerodynamic Force, Aerodynamic Moment, Wind-axes Velocities, Dynamic Pressure, Lift Coefficient, Drag Coefficient, Side Coefficient, Pitch Coefficient, Roll Coefficient e Yaw Coefficient. Uma vez que se utiliza o modelo disponível na livraria MatLab/simulink®, optou-se por manter inalteráveis os nomes dos blocos e das variáveis, por corresponderem a termos técnicos utilizados em aeronáutica. Apresenta-se, na figura 3.8, o diagrama interno do bloco do modelo aerodinâmico.

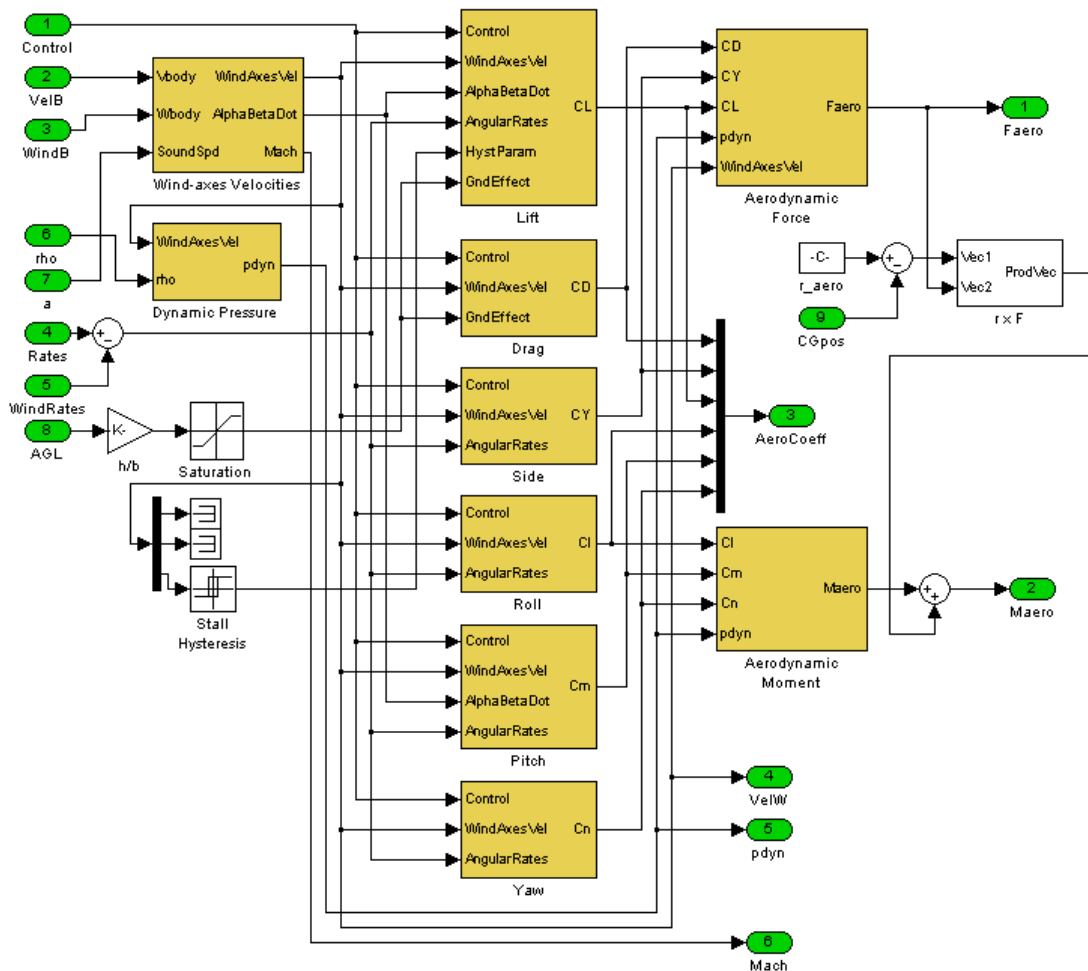


Figura 3.8 – Diagrama interno do bloco do modelo aerodinâmico

O Bloco *Aerodynamic Force* tem como entradas:

$C_D$  (CD) - Coeficiente de resistência

$C_Y$  (CY) - Coeficiente de força lateral

$C_L$  (CL) - Coeficiente de sustentação

$[V_a, \alpha, \beta]$  (WindAxesVel) -  $V_a$  (m/s) Velocidade do ar,  $\alpha$  (rad) Ângulo de ataque e  $\beta$  (rad) Ângulo de derrapagem

$q_d$  (pdyn) - Pressão dinâmica,  $q_d = \frac{1}{2} \rho V_a^2$ , com  $\rho$  - Densidade do ar

Rege-se pelas seguintes expressões:

$$\begin{cases} D = q_d \cdot S \cdot C_D \\ Y = q_d \cdot S \cdot C_Y \\ L = q_d \cdot S \cdot C_L \end{cases} \quad (3.3)$$

Em que:

$S$  - Área da Asa

Onde a saída D, Y e L, representa o vector *Aerodynamic Force* (Faero).

O Bloco *Aerodynamic Moment* tem como entradas:

$C_l$  (Cl) - Coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa

$C_m$  (Cm) - Coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa

$C_n$  (Cn) - Coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa

$q_d$  (pdyn)

Rege-se pelas seguintes expressões:

$$\begin{cases} L_a = q_d \cdot S \cdot b \cdot C_l \\ M = q_d \cdot S \cdot c \cdot C_m \\ N = q_d \cdot S \cdot b \cdot C_n \end{cases} \quad (3.4)$$

Em que:

$c$  - Corda média da aerodinâmica da asa

$b$  - Envergadura da asa

Onde a saída  $L_a$ ,  $M$  e  $N$ , representa o vector *Aerodynamic Moment* (Maero).

O Bloco *Wind-axes Velocities* tem como entradas:

$V_{body}$  (Vbody) - Velocidade do centro de massa em relação ao solo (3x1)

$W_{body}$  (Wbody) - Velocidade do vento em relação ao centro de massa (3x1)

$a$  (SoundSpd) - Velocidade do som à altitude actual

Rege-se pelas seguintes expressões:

$$\begin{cases} [u, v, w] = \vec{V}_{body} - \vec{W}_{body} \\ V_a = \sqrt{u^2 + v^2 + w^2} \end{cases} \Rightarrow \begin{cases} \alpha = \text{atan}\left(\frac{w}{u}\right) \\ \beta = \text{asin}\left(\frac{v}{V_a}\right) \\ M = \frac{V_a}{a} \end{cases} \quad (3.5)$$

Onde o vector de saída  $[V_a, \alpha, \beta]$  (WindAxesVel). A relação entre velocidade ar e velocidade do som à altitude actual é dada por  $M$  (Mach). Existe outra saída  $[\dot{\alpha}, \dot{\beta}]$  (AlphaBetaDot) que é dada pela derivada do Ângulo de ataque e pela derivada Ângulo de derrapagem, ambas em ordem ao tempo.

O Bloco *Dynamic Pressure* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $\rho$  (rho)

Rege-se pela seguinte expressão:

$$q_d = \frac{1}{2} \rho V_a^2 \quad (3.6)$$

Obtém-se a saída  $q_d$  (pdyn).

O Bloco *Lift Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\dot{\alpha}, \dot{\beta}]$  (AlphaBetaDot),  $M$  (Mach)

$[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control) -  $\delta_f$  Flap,  $\delta_e$  Elevator,  $\delta_a$  Aileron e  $\delta_r$  Rudder

$[p, q, r]$  (AngularRates) - Velocidade angular em relação ao solo

Rege-se pela seguinte expressão:

$$C_L = C_{L0} + C_L^\alpha \cdot \alpha + C_L^{\delta_f} \cdot \delta_f + C_L^{\delta_e} \cdot \delta_e + \frac{c}{2 \cdot V_a} (C_L^{\dot{\alpha}} \cdot \dot{\alpha} + C_L^q \cdot q) + C_L^M \cdot M \quad (3.7)$$

Em que:

$C_{L0}$  - Coeficiente de sustentação para zero de ângulo de ataque

$C_L^\alpha$  - Derivada do coeficiente de sustentação em ordem a  $\alpha$

$C_L^{\delta_f}$  - Derivada do coeficiente de sustentação em ordem a  $\delta_f$

$C_L^{\delta_e}$  - Derivada do coeficiente de sustentação em ordem a  $\delta_e$

$C_L^{\dot{\alpha}}$  - Derivada do coeficiente de sustentação em ordem a  $\dot{\alpha}$

$C_L^q$  - Derivada do Coeficiente de sustentação em ordem a  $q$

$C_L^M$  - Derivada do Coeficiente de sustentação em ordem a  $M$

Onde a saída  $C_L$ , representa o coeficiente de sustentação (CL).

O Bloco *Drag Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control),  $C_L$  (CL) e  $M$  (Mach)

Este bloco rege-se pela seguinte expressão:

$$C_D = C_{D0} + \frac{(C_L - C_{L0})^2}{\pi e AR} + C_D^{\delta_f} \cdot \delta_f + C_D^{\delta_e} \cdot \delta_e + C_D^{\delta_a} \cdot \delta_a + C_D^{\delta_r} \cdot \delta_r + C_L^M \cdot M \quad (3.8)$$

Em que:

$C_{L0}$  - Coeficiente de sustentação para o mínimo de coeficiente de resistência

$C_{D0}$  - Mínimo coeficiente de resistência da aeronave

$C_D^{\delta_f}$  - Derivada do coeficiente de resistência em ordem a  $\delta_f$

$C_D^{\delta_e}$  - Derivada do coeficiente de resistência em ordem a  $\delta_e$

$C_D^a$  - Derivada do coeficiente de resistência em ordem a  $a$

$C_D^r$  - Derivada do coeficiente de resistência em ordem a  $r$

$C_D^M$  - Derivada do coeficiente de resistência em ordem a  $M$

Onde a saída  $C_D$ , representa o coeficiente de resistência (CD).

O Bloco *Side Force Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control) e  $[p, q, r]$  (AngularRates)

Este bloco rege-se pela seguinte expressão:

$$C_Y = C_D^\beta \cdot \delta_Y + C_Y^{\delta_a} \cdot \delta_a + C_Y^{\delta_r} \cdot \delta_r + \frac{b}{2V_a} (C_Y^p \cdot p + C_Y^r \cdot r) \quad (3.9)$$

Em que:

$C_Y^\beta$  - Derivada do coeficiente de força lateral em ordem a  $\beta$

$C_Y^{\delta_a}$  - Derivada do coeficiente de força lateral em ordem a  $\delta_a$

$C_Y^{\delta_r}$  - Derivada do coeficiente de força lateral em ordem a  $\delta_r$

$C_Y^p$  - Derivada do coeficiente de força lateral em ordem a  $p$

$C_Y^r$  - Derivada do coeficiente de força lateral em ordem a  $r$

Onde a saída  $C_Y$ , representa o coeficiente de força lateral (CY).

O Bloco *Pitch Moment Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control),  $[\dot{\alpha}, \dot{\beta}]$  (AlphaBetaDot),  $[p, q, r]$  (AngularRates) e  $M$  (Mach)

Este bloco rege-se pela seguinte expressão:

$$C_m = C_{m0} + C_m^\alpha \cdot \alpha + C_m^{\delta_f} \cdot \delta_f + C_m^{\delta_e} \cdot \delta_e + \frac{c}{2V_a} (C_m^{\dot{\alpha}} \cdot \dot{\alpha} + C_m^q \cdot q) + C_m^M \cdot M \quad (3.10)$$

Em que:

$C_{m0}$  - Coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa para zero de ângulo de ataque

$C_m^\alpha$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $\alpha$

$C_m^{\delta_f}$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $\delta_f$

$C_m^{\delta_e}$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $\delta_e$

$C_m^{\dot{\alpha}}$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $\dot{\alpha}$

$C_m^q$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $q$

$C_m^M$  - Derivada do coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa em ordem a  $M$

Onde a saída  $C_m$ , representa o coeficiente do Momento de *Pitch* em relação ao centro aerodinâmico da asa ( $C_m$ ).

O Bloco *Roll Moment Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control) e  $[p, q, r]$  (AngularRates)

Este bloco rege-se pela seguinte expressão:

$$C_l = C_l^\beta \cdot \beta + C_l^{\delta_a} \cdot \delta_a + C_l^{\delta_r} \cdot \delta_r + \frac{b}{2V_a} (C_l^p \cdot p + C_l^r \cdot r) \quad (3.11)$$

Em que:

$C_l^\beta$  - Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\beta$

$C_l^{\delta_a}$  - Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\delta_a$

$C_l^{\delta_r}$  - Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $\delta_r$

$C_l^p$  - Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $p$

$C_l^r$  - Derivada do coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa em ordem a  $r$

Onde a saída  $C_l$ , representa o coeficiente do Momento de *Roll* em relação ao centro aerodinâmico da asa (Cl).

O Bloco *Yaw Moment Coefficient* tem como entradas:

$[V_a, \alpha, \beta]$  (WindAxesVel),  $[\delta_f, \delta_e, \delta_a, \delta_r]$  (Control) e  $[p, q, r]$  (AngularRates)

Este bloco rege-se pela seguinte expressão:

$$C_n = C_n^\beta \cdot \beta + C_n^{\delta_a} \cdot \delta_a + C_n^{\delta_r} \cdot \delta_r + \frac{b}{2V_a} (C_n^p \cdot p + C_n^r \cdot r) \quad (3.12)$$

Em que:

$C_n^\beta$  - Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\beta$

$C_n^{\delta_a}$  - Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\delta_a$

$C_n^{\delta_r}$  - Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $\delta_r$

$C_n^p$  - Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $p$

$C_n^r$  - Derivada do coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa em ordem a  $r$

Onde a saída  $C_n$ , representa o coeficiente do Momento de *Yaw* em relação ao centro aerodinâmico da asa (Cn).

### 3.3.2 – Descrição do Modelo de Propulsão

A resposta do modelo de propulsão depende do motor utilizado na aeronave, onde os parâmetros deste modelo são relativos ao motor em funcionamento ao nível do médio do mar. No entanto, o sistema de equações constituintes do modelo de propulsão tem em consideração a altitude da aeronave.

Internamente, o modelo de propulsão é constituído pelos seguintes blocos: *FG Piston Engine* e *FG Fixed Pitch Propeller*. Na figura 3.9, apresenta-se o diagrama interno do bloco do modelo de propulsão.

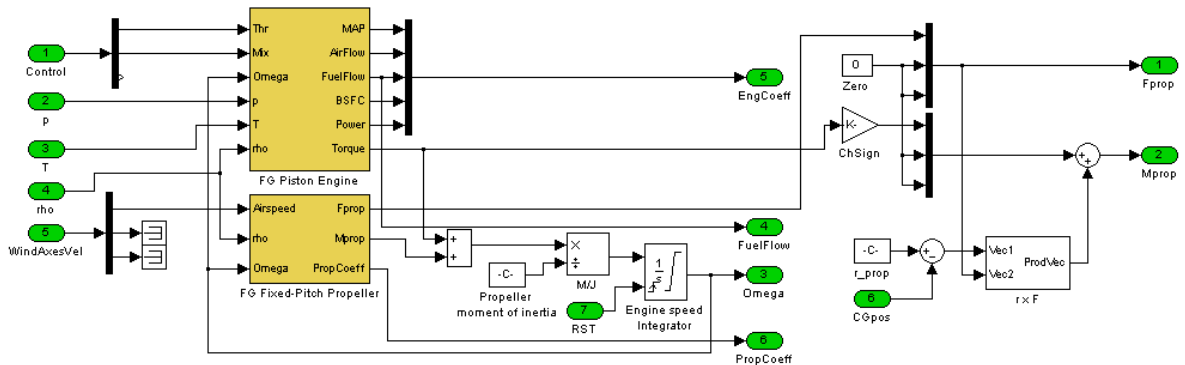


Figura 3.9 – Diagrama interno do bloco do modelo de propulsão

O Bloco *FG Fixed Pitch Propeller* tem como entradas:

$V_a$  (AirSpeed),  $\rho$  (rho) e  $\Omega$  (Omega) - Velocidade angular da rotação da hélice (rad/s)

Este bloco rege-se pelas seguintes expressões:

$$\begin{cases} J = \frac{\pi V_a}{\Omega R} \\ F_p = \frac{4}{\pi^2} \rho R^4 \Omega^2 C_T \\ M_p = \frac{4}{\pi^2} \rho R^5 \Omega^2 C_p \end{cases} \quad (3.13)$$

Em que:

$C_T$  - Coeficiente de impulso

$C_p$  - Coeficiente de potência

$R$  - Raio da hélice

Onde a saída é composta pelo vector coeficiente da hélice  $J$ ,  $C_T$  e  $C_p$  (ProCoeff), Força de impulso da hélice  $F_p$  (Fprop) e Binario de hélice  $M_p$  (Mprop).

O Bloco *FG Piston Engine* consiste numa correspondência entre dados, utilizando, para o efeito, funções do *MatLab/simulink*<sup>®</sup> do tipo *lookup table*, onde as entradas são:

(Thr) - *Throttle*

(Mix) - Relação ar/combustível

(Omega) - Velocidade de rotação do veio do motor

(p) - Pressão atmosférica na altitude actual

(T) - Temperatura atmosférica na altitude actual

Obtêm-se as saídas correspondentes:

(MAP) - Pressão do ar para a posição actual de *Throttle*

(Airflow) - Fluxo de ar

(Fuelflow) - Fluxo de combustível

(BSFC) - Consumo de combustível na travagem

(Power) - Potência instantânea do motor

(Torque) - Binário produzido no veio do motor

### 3.3.3 – Descrição do Modelo Atmosférico e do Modelo Inercial

O modelo atmosférico consiste em sistemas de equações que descrevem fenómenos físicos num dado local a partir de variáveis, tais como a velocidade do vento e a altitude em relação ao nível médio do mar, resultando nas variáveis de temperatura, pressão atmosférica, densidade do ar e velocidade de propagação do som desse local.

Internamente, o modelo atmosférico é constituído pelos seguintes blocos: *Standard Atmosphere*, *Background Wind*, *Turbulence* e *Wind Shear*. Na figura 3.10, apresenta-se o diagrama interno do bloco do modelo atmosférico.

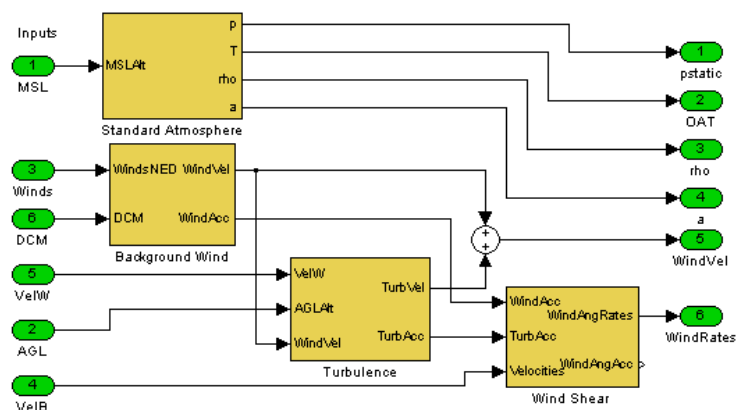


Figura 3.10 – Diagrama interno do bloco do modelo atmosférico

O Bloco *Standard Atmosphere* consiste numa correspondência entre dados, utilizando, para o efeito, funções do *MatLab/simulink*<sup>®</sup> do tipo *lookup table* onde as entradas são:

(MSLAlt) – Altitude actual em relação ao nível médio do mar

Obtêm-se as saídas correspondentes:

(p), (T), (rho) e (SoundSpd)

O Bloco *Background Wind* consiste numa correspondência entre dados, onde as entradas são:

(WindsNED) – Componente da velocidade em relação ao referencial inercial (*North-East-Down*)

(DCM) - *Direction Cosine Matrix Body*

Obtêm-se as saídas correspondentes:

(WindVel) – Velocidade do vento

(WindAcc) – Aceleração do vento

O Bloco *Turbulence* consiste numa correspondência entre dados, onde as entradas são:

$[V_a, \alpha, \beta]$  (VelW), (AGLAlt) = (MSLAlt) e (WindVel)

Obtêm-se as saídas correspondentes:

(TurbVel) - Vector velocidade da turbulência

(TurbAcc) - Vector aceleração da turbulência

O Bloco *Wind Shear* tem como entradas:

$\dot{W}_{Wind}$  (WindAcc),  $\dot{v}_{Wind}$  (TurbAcc) e  $u_{aircraft}$  (velocities) – Velocidade da aeronave

Este bloco rege-se pelas seguintes expressões:

$$\begin{cases} q_{Wind} = \frac{1}{u_{aircraft}} \cdot \frac{d_{w_{Wind}}}{dt} \\ r_{Wind} = \frac{1}{u_{aircraft}} \cdot \frac{d_{v_{Wind}}}{dt} \end{cases} \quad (3.14)$$

Onde a saída é composta por:  $q_{Wind}$  (WindAngRates) - Vector aceleração do vento e  $r_{Wind}$  (WindAngAcc) - Vector aceleração angular causado pelo vento.

No modelo inercial, os momentos de inércia são calculados com base na interpolação linear do centro de massa da aeronave com combustível e o centro de massa da aeronave sem combustível. A quantidade de combustível nos tanques é calculada com base na

integração do consumo de combustível. Na figura 3.11, apresenta-se o diagrama interno do bloco do modelo inercial.

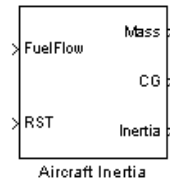


Figura 3.11 – Diagrama interno do bloco do modelo inercial

Onde as entradas são:

(FuelFlow) e (RST) - *Reset* ao integrado

E as saídas:

(Mass) - Massa actual da aeronave

(CG) - Centro de massa em relação ao referencial da aeronave  $[X_a, Y_a, Z_a]$

(Inertia) - Momento de inércia referenciado ao centro de massa actual  $[J_x, J_y, J_z, J_{xz}]$

### 3.3.4 – Descrição do Modelo da Terra

O bloco do modelo da Terra, apresentado na figura 3.12, tem como objectivo calcular a altura da aeronave em relação ao nível médio do mar e obter a força da gravidade associada a uma determinada localização.

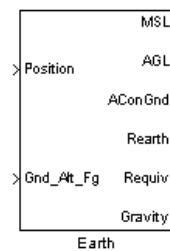


Figura 3.12 – Bloco do modelo da Terra

Sabendo-se que a superfície da terra é irregular, torna-se difícil criar um modelo matemático que defina o nível médio do mar em todas as zonas terrestres [20]. Então é designado de WGS-84, uma Elipsóide como referência da superfície terrestre baseada em expressões matemáticas apresentadas em (3.15).

$$\left\{ \begin{array}{l} R_{meridian} = \frac{r_e(1-\varepsilon^2)}{(1-\varepsilon^2 \sin^2 \phi)^{\frac{3}{2}}} \\ R_{normal} = \frac{r_e}{(1-\varepsilon^2 \sin^2 \phi)^{\frac{1}{2}}} \\ R_{equiv} = \sqrt{R_{meridian} R_{normal}} \\ g = gwGS_0 \frac{1 + gwGS_1 \sin^2 \phi}{(1-\varepsilon^2 \sin^2 \phi)^{\frac{1}{2}}} \end{array} \right. \quad (3.15)$$

Sendo a entrada a posição da aeronave (Position) e considerando os seguintes coeficientes de WGS-84:

$$r_e - 6378137m$$

$$\varepsilon - 0,081819908426$$

$$gwGS_0 - 9,7803267714m/s^2$$

$$gwGS_1 - 0,00193185138639m/s^2 \text{ e}$$

$\phi$  - Latitude da aeronave

Obtém-se na saída:

$R_{meridian}$  (Rmeridian) - Raio do meridiano (m)

$R_{normal}$  (Rnormal) - Raio normal (m)

$R_{equiv}$  (Re) - Raio equivalente (m)

$g$  (Gravity) – Aceleração da gravidade para o local actual ( $m/s^2$ )

Por outro lado, considera-se ainda uma superfície equipotencial do campo gravítico em toda a superfície da terra, denominada Geóide, que representa a superfície que melhor se aproxima do nível médio do mar em toda a superfície terrestre, inclusive podendo essa superfície ser estendida aos continentes. O Geóide é designado por EGM-96.

Ou seja, para o modelo da terra, utiliza-se internamente o bloco WGS-84 e o bloco EGM-96 existente na livraria da *toolbox Aerosim*<sup>®</sup>. No *software MatLab/simulink*<sup>®</sup>, o bloco EGM-96 utiliza valores tabelados para a superfície do nível médio do mar em relação à elipsóide, baseados em dados fornecidos pela NASA, com uma resolução de um grau em todas as direcções, [32] e [17]. Na figura 3.13, apresenta-se a superfície terrestre, com a representação do Geóide e da Elipsóide.

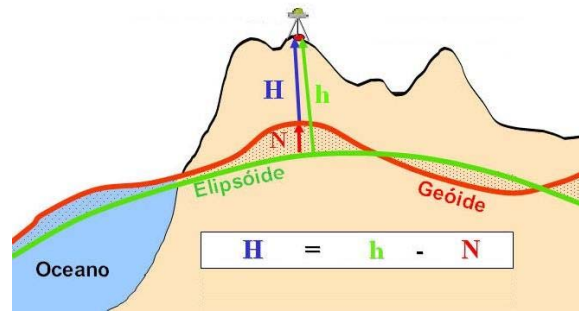


Figura 3.13 – Superfície terrestre, com representação do Geóide e da Elipsóide, [21]

Para o cálculo da altura da aeronave em relação ao solo (AGL), existe um bloco que calcula a diferença entre a altura da aeronave e a altura do solo, ambos referenciados ao nível do médio do mar. Na figura 3.14, apresenta-se o diagrama interno do bloco do modelo da terra.

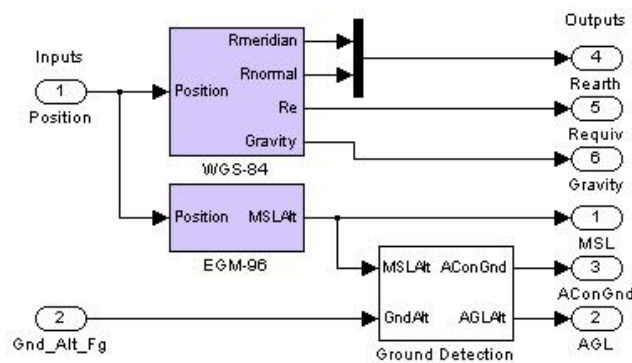


Figura 3.14 – Diagrama interno do bloco do modelo da terra

Na tabela 3.4 apresentam-se as variáveis de entrada do bloco do modelo da terra.

Tabela 3.4 – Variáveis de entrada do bloco do modelo terra

Designação das Variáveis	Entrada
Position (Posição da aeronave em coordenada geográficas)	[lat Lon h] (rad rad m)
Gnd_Alt_Fg (Altura ao solo em relação ao nível do mar)	Gnd_Alt_fg (m)

Na tabela 3.5, apresentam-se as variáveis de saída do bloco do modelo da terra.

Tabela 3.5 – Variáveis de saída do bloco do modelo terra

Designação das Variáveis	Saída
Rearth (Raio do meridiano)	rearth (m)
Requiv (Raio normal)	requiv (m)
Gravity (Gravidade local)	gravity (m/s <sup>2</sup> )
MSL (Altura em relação ao nível médio do mar)	msl (m)
AConGnd (Indicador de Aeronave no solo)	0 ou 1
AGL (Altura em relação ao solo)	agl(m)

### 3.3.5 – Descrição do Modelo das Equações do Movimento

O bloco das equações do movimento descreve a dinâmica da aeronave (que permite a realização da simulação), sendo constituído essencialmente por equações diferenciais. Na figura 3.15, apresenta-se o bloco das equações do movimento referente ao modelo Cessna-172.

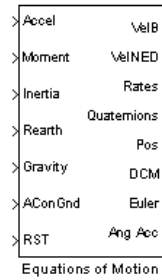


Figura 3.15 – Bloco das equações do movimento referente ao modelo Cessna-172

Internamente este bloco é constituído pelos blocos, *Velocity*, *Position*, *Attitude Quaternions* e *Angular Rate*. Na figura 3.16, apresenta-se o diagrama interno do bloco das equações de movimento.

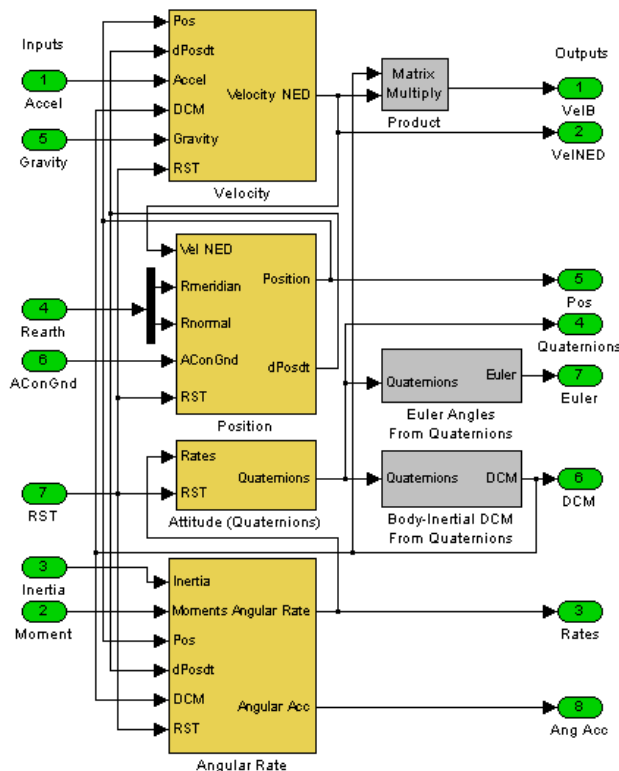


Figura 3.16 – Diagrama interno do bloco das equações de movimento

O bloco *Velocity* tem por função integrar as acelerações da aeronave e obter as velocidades inerciais. As condições iniciais para as integrações são definidas na parametrização do bloco Cessna-172. De salientar que sempre que se realiza uma limpeza

de registos, estes voltam aos seus valores iniciais. As acelerações consideradas devem incluir todas as forças aplicadas na aeronave, incluindo os sistemas de propulsão.

O Bloco *Velocity* tem como entradas:

$[Lat, Lon, Alt]$ (Pos) - Vector de posição da aeronave

$[\dot{Lat}, \dot{Lon}, \dot{Alt}]$ (dPosdt) - Vector da derivada da posição da aeronave

$[a_x, a_y, a_z]$ (Accel) – Vector da aceleração

$R_{p2n}$  (DCM) ,  $g$  (Gravity) e (RST)

Rege-se pelas seguintes expressões:

$$\begin{cases} \dot{V}_N = -(\dot{Lon} + 2\omega_{ie}) \cdot \sin(Lat) \cdot V_E + \dot{Lat} \cdot V_D + f_N \\ \dot{V}_E = -(\dot{Lon} + 2\omega_{ie}) \cdot \sin(Lat) \cdot V_N + (\dot{Lon} + 2\omega_{ie}) \cdot \cos(Lat) \cdot V_D + f_E \\ \dot{V}_D = -\dot{Lat} \cdot V_N - (\dot{Lon} + 2\omega_{ie}) \cdot \cos(Lat) \cdot V_E + f_D + g \end{cases} \quad (3.16)$$

$$\begin{cases} f_N = R_{p2n} \cdot a_x \\ f_E = R_{p2n} \cdot a_y \\ f_D = R_{p2n} \cdot a_z \end{cases} \quad (3.17)$$

Onde a saída  $\dot{V}_N$  ,  $\dot{V}_E$  e  $\dot{V}_D$  representa o vector da derivada da velocidade em relação ao Geóide e  $V_N$  ,  $V_E$  e  $V_D$  representa o vector velocidade em relação ao Geóide (VelocityNED).

No bloco *Position*, obtém-se a posição actual da aeronave através da integração das velocidades inerciais e das variáveis provenientes do bloco modelo da terra. A posição de saída é um vector de estado da aeronave que, por sua vez, é utilizado no sistema WSG-84 e EGM-96 do bloco da terra. A saída da derivada de posição é utilizada pelo bloco de velocidade e pelo bloco *Angular Rates*.

O Bloco *Position* tem como entradas:

$[V_N, V_E, V_D]$ (VeINED),  $R_{meridian}$  (Rmeridian),  $R_{normal}$  (Rnormal), (RST) e

*AConGnd* (AConGnd) - Indicador de Aeronave no solo

Rege-se pelas seguintes expressões:

$$\begin{cases} \dot{Lat} = \frac{V_{North}}{R_{meridian} + Alt} \\ \dot{Lon} = \frac{V_{East}}{(R_{normal} + Alt) \cdot \cos(Lat)} \\ \dot{Alt} = \begin{cases} -V_{Down}, AConGnd = 0 \\ 0, AConGnd = 1 \end{cases} \end{cases} \quad (3.18)$$

Onde a saída  $\dot{Lat}$ ,  $\dot{Lon}$  e  $\dot{Alt}$  representa o vector derivada da posição da aeronave (dPosdt) e  $Lat$ ,  $Lon$  e  $Alt$  representa a posição da aeronave (Position).

O bloco *Attitude Quaternions* integra a velocidade angular e obtêm-se os ângulos de Euler da aeronave em relação ao solo, onde tem com entradas:

(RST) e  $[p, q, r]$ (Rates)

Rege-se pelas seguintes expressões:

$$\begin{bmatrix} \dot{e}_0 \\ \dot{e}_x \\ \dot{e}_y \\ \dot{e}_z \end{bmatrix} = \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \cdot \begin{bmatrix} e_0 \\ e_x \\ e_y \\ e_z \end{bmatrix} \quad (3.19)$$

Onde a saída  $\dot{e}_0$ ,  $\dot{e}_x$ ,  $\dot{e}_y$  e  $\dot{e}_z$  representa Vector of Quaternions (Quaternions).

O bloco Angular Rates integra os momentos e inércias de um corpo rígido (aeronave) definido por seis graus de liberdade e obtêm a velocidade instantânea angular em relação ao solo (Angular Rates). Em que as entradas são:

$[L, M, N]$ (Moments) - Vector dos momentos em relação ao centro de massa

$[J_x, J_y, J_z, J_{xz}]$  (Inertia),  $[Lat, Lon, Alt]$  (Pos),  $[\dot{Lat}, \dot{Lon}, \dot{Alt}]$  (dPosdt),  $R_{p2n}$  (DCM) e (RST)

Rege-se pelas seguintes expressões:

$$\begin{cases} \dot{p} = (c_1 \cdot r + c_2 \cdot p) \cdot q + c_3 \cdot L + c_4 \cdot N - (\omega_{in}^p)_x \\ \dot{q} = c_5 \cdot p \cdot r - c_6 \cdot (p^2 - r^2) + c_7 \cdot M - (\omega_{in}^p)_y \\ \dot{r} = (c_8 \cdot p - c_2 \cdot r) \cdot q + c_4 \cdot L + c_9 \cdot N - (\omega_{in}^p)_z \end{cases} \quad (3.20)$$

Em que  $c_1$  a  $c_9$  representam os coeficientes de inércia calculados com base nas expressões (3.16), conhecendo o vector do momento de inércia em relação ao centro de massa  $[J_x, J_y, J_z, J_{xz}]$  e  $\omega_{in}^p$  representa a velocidade angular de rotação da Terra.

$$\begin{aligned}
 c_1 &= \frac{(J_y - J_z) \cdot J_z - J_{xz}^2}{J_x \cdot J_z - J_{xz}^2} & c_4 &= \frac{J_{xz}}{J_x \cdot J_z - J_{xz}^2} & c_7 &= \frac{1}{J_y} \\
 c_2 &= \frac{(J_x - J_y - J_z) \cdot J_{xz}}{J_x \cdot J_z - J_{xz}^2} & c_5 &= \frac{J_z - J_x}{J_y} & c_8 &= \frac{J_x \cdot (J_x - J_y) + J_{xz}^2}{J_x \cdot J_z - J_{xz}^2} \\
 c_3 &= \frac{J_z}{J_x \cdot J_z - J_{xz}^2} & c_6 &= \frac{J_{xz}}{J_y} & c_9 &= \frac{J_x}{J_x \cdot J_z - J_{xz}^2}
 \end{aligned} \tag{3.21}$$

$$\omega_{in}^p = R_{n2p} \cdot \begin{bmatrix} (\dot{Lon} + \omega_{ie}) \cos(Lat) \\ -Lat \\ -(\dot{Lon} + \omega_{ie}) \sin(Lat) \end{bmatrix} \tag{3.22}$$

Onde a saída  $\dot{p}$ ,  $\dot{q}$  e  $\dot{r}$  representa o vector de aceleração angular (AngularAcc) e  $p$ ,  $q$  e  $r$  representa velocidade angular (AngularRate).

Existem dois blocos internos que são utilizados nas equações de movimento para realizar as transformações matemáticas de *Quaternions* para ângulos de Euler e para DCM.

Na tabela 3.6, apresentam-se as variáveis de entrada do bloco equações do movimento.

Tabela 3.6 – Variáveis de entrada do bloco equações de movimento

Designação das Variáveis	Entrada
Accel (Aceleração)	accel [ $a_x$ $a_y$ $a_z$ ] (m/s <sup>2</sup> )
Moment (Momentos de inércia)	moment [L M N] (kg/N)
Inertia (Inércia em relação ao centro de massa)	inertia [ $J_x$ $J_y$ $J_z$ $J_{xz}$ ] (kg/N)
Rearth (Raio do meridiano)	rearth (m)
Gravity (Gravidade local)	gravity (m/s <sup>2</sup> )
AConGnd (Indicador de aeronave no solo)	0 ou 1
RST (Limpar registos)	0 ou 1

Na tabela 3.7, apresentam-se as variáveis de saída do bloco equações do movimento.

Tabela 3.7 – Variáveis de saída do bloco equações do movimento

Designação das Variáveis	Saída
VelB (Velocidade)	[ $u$ $v$ $w$ ]
VelNED (Velocidade)	[ $V_N$ $V_E$ $V_D$ ]
Rates (Velocidade angular em relação ao solo)	[ $p$ $q$ $r$ ] (rad/m)
Quaternions	[ $e_0$ $e_x$ $e_y$ $e_z$ ]
Pos	[Lat Lon Alt] (rad rad m)
DCM	-
Euler	[ $\Phi$ $\theta$ $\Psi$ ]
Ang Acc	[ $p$ $q$ $r$ ]

No modelo da aeronave Cessna-172 está contemplado um simples controlo de estabilidade (Damper) para reduzir as oscilações durante o voo simulado, aplicado aos movimentos de Pitch e Roll, [17]. Desta forma, reduzem-se as perturbações ao longo o voo tornando a condução mais fácil e confortável. Verifica-se a existência de uma diferença entre a altura da aeronave gerada pelo modelo matemático e a altura visualizada no *software FlightGear*<sup>®</sup>. Assim, com o intuito de corrigir esse *off-set* foi necessário adicionar duas constantes a essa mesma altura. Uma constante de 53m para correcção geral da altura e uma constante de 1,6m para correcção da altura relativa ao centro geométrico da aeronave em relação ao solo, determinadas experimentalmente.

Na figura 3.17, apresenta o diagrama de blocos do modelo da aeronave Cessna-172 com controlo de estabilidade.

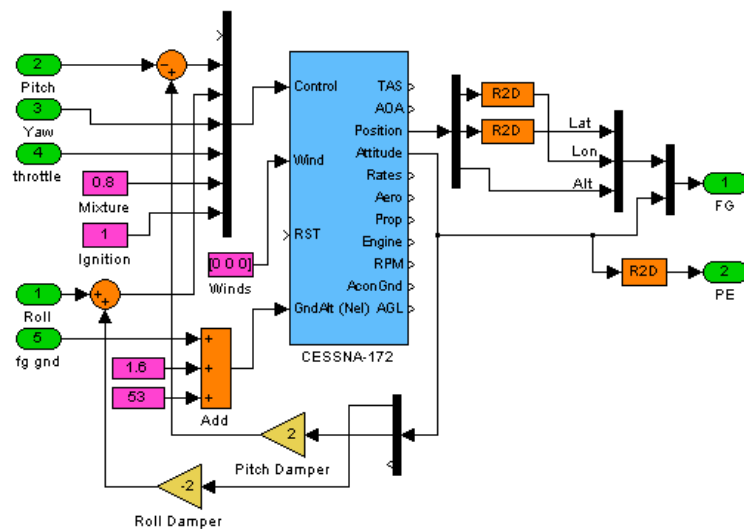


Figura 3.17 – Diagrama de blocos do modelo da aeronave Cessna-172 com controlo de estabilidade

De forma a simplificar o diagrama de blocos do modelo da aeronave Cessna-172 com controlo de estabilidade, foi construído um subsistema em *MatLab/simulink*<sup>®</sup>. Na figura 3.18, apresenta-se o bloco Modelo Cessna-172.



Figura 3.18 – Bloco Modelo Cessna-172

O bloco do modelo Cessna-172 foi dimensionado para conter o mínimo de entradas e saídas necessárias para simplificar a sua utilização no simulador. As entradas *Roll*, *Pitch*, *Yaw* e *Throttle* são provenientes do comando do *joystick* e a entrada *Fg\_Gnd* é proveniente

do *software FlightGear*<sup>®</sup>. Foram concebidas duas saídas FG e PE, nas quais a primeira envia a informação mínima necessária para a animação no *software FlightGear*<sup>®</sup> e a segunda envia os dados para o accionamento da plataforma electromecânica.

### 3.4 – *Software de Simulação FlightGear*<sup>®</sup>

O *software FlightGear*<sup>®</sup>, por ser um sistema de código aberto, torna-se numa estrutura flexível, possibilitando sempre alterações por parte de qualquer utilizador que detenha conhecimentos nesse domínio, por vista a melhorar a simulação, através da inclusão de, por exemplo, novos modelos de aeronaves e novos cenários.

A simulação aerodinâmica é uma das componentes com maior relevância nos simuladores em geral, uma vez que deve reproduzir a mesma reacção da aeronave real durante um voo, pois, o seu desempenho e qualidade são fundamentais para a satisfação dos utilizadores. Os modelos dinâmicos das aeronaves são projectados de forma a reproduzir sensações semelhantes às das aeronaves reais. Quando existem erros nos modelos dinâmicos, a envolvência do utilizador com o ambiente virtual deixa de existir, [22].

É comum, na simulação do ambiente em redor do piloto, existirem modelos gráficos animados tridimensionais, representativos do interior do *cockpit*, incluindo interruptores de painel, instrumentos virtuais, relação visual entre o comando da aeronave e o comando físico através do *joystick* e pedais. Também, no exterior, existe animação gráfica, nomeadamente nos lemes de direcção e hélice do motor de propulsão.

Na figura 3.19, apresenta-se uma imagem representativa da vista do piloto numa aeronave Cessna-172 ao aproximar-se da pista de aterragem no aeroporto internacional de São Francisco, no *software FlightGear*<sup>®</sup>.

Este tipo de *software* pode ser muito útil para a aprendizagem de condução de aeronaves. No entanto, há que ter em conta que um simulador não é por si só uma interacção gráfica. É necessário reunir características que garantam ao piloto uma sensação de estar num ambiente idêntico ao de uma aeronave real. Neste sentido, há que ter em consideração alguns aspectos, tais como, os comandos da aeronave simulada serem o mais idênticos aos da aeronave real, ter o mesmo campo de visão, ter a mesma disposição dos instrumentos de painel, reproduzir as mesmas falhas dos instrumentos reais e todas as manobras realizadas devem ter a mesma resposta quer no simulador quer na aeronave real, [22].



Figura 3.19 – Visão do piloto numa aeronave Cessna-172 ao aproximar-se da pista de aterragem do aeroporto internacional de São Francisco no *software FlightGear*<sup>®</sup>, [22]

Neste contexto, e porque a aprendizagem de um futuro piloto de aviação passa pelo treino num simulador, se esse simulador transmitir uma sensação de facilitismo na condução de uma aeronave, esse futuro piloto poderá ter uma falsa sensação de segurança na condução. Por esse motivo é fundamental que os simuladores tenham comportamentos o mais aproximados possíveis aos das aeronaves reais em todos os aspectos, nomeadamente nas simulações de avarias mecânicas e de avarias na instrumentação, [22].

O *software FlightGear*<sup>®</sup> permite várias opções de iniciação. Existe a possibilidade de executar o videojogo em modo normal, ou em modo remoto onde é possível escolher determinadas opções e arranque.

É programado o ficheiro “runfg.bat”, indicado no anexo A, próprio para a iniciação do *software*, o qual inclui as opções pretendidas de arranque, dentro de uma possibilidade de configurações possíveis. Em modo remoto uma das opções consiste na criação de uma porta de comunicação com o *software MatLab/simulink*<sup>®</sup> para a escrita da posição da aeronave em coordenadas geográficas e a escrita dos respectivos ângulos de Euler da aeronave em relação ao solo. Outra opção relevante consiste na criação de outra porta de comunicação exclusiva para a leitura da altura do solo em relação ao nível médio do mar, tendo por referência a posição da aeronave no ambiente virtual. Neste modo remoto de execução do *FlightGear*<sup>®</sup>, este fica unicamente a funcionar como uma saída visual de animação gráfica, em que necessita remotamente das variáveis descritas acima para conferir o efeito de movimento ao ambiente virtual.

O modelo da aeronave Cessna-172 em *MatLab/simulink*<sup>®</sup> assume como constante a altura do solo em relação ao nível médio do mar, funcionando correctamente para voos onde a essa altura seja constante, [17]. No entanto, no ambiente gráfico onde se propõe realizar a simulação isso não acontece. Por esse motivo, torna-se necessário durante o voo obter em cada instante a altura do solo em relação ao nível médio do mar, através da leitura da porta de comunicação do *software FlightGear*<sup>®</sup>.

As variáveis a enviar para o *software FlightGear*<sup>®</sup> traduzem-se num vector resultante do modelo da aeronave Cessna-172 em *MatLab/simulink*<sup>®</sup>, que é constituído pelos movimentos de translação Longitude, Latitude e Altitude e pelos movimentos de rotação *Roll*, *Pitch* e *Yaw*. Apresenta-se, na figura 3.20, o bloco do *MatLab/simulink*<sup>®</sup> para a escrita de variáveis no *software FlightGear*<sup>®</sup>. Neste Bloco configura-se o endereço da rede *Ethernet IP (internet protocol)*, a porta de comunicação e o tempo de actualização.

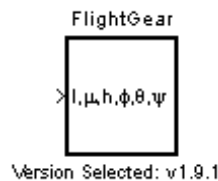


Figura 3.20 – Bloco do *MatLab/simulink*<sup>®</sup> para a escrita de variáveis no *software FlightGear*<sup>®</sup>

Para realizar a leitura da altura do solo em relação ao nível médio do mar através da porta de comunicação concebida no *software FlightGear*<sup>®</sup> foi necessário desenvolver um programa auxiliar “fg\_simulink”, adaptado de [23], uma vez que o *software MatLab/simulink*<sup>®</sup> não apresentara nenhuma solução de leitura das variáveis do videojogo directamente. Após a inicialização do programa, são executadas duas rotinas de entrada onde é feita a abertura das portas de comunicação para leitura da variável (*gnd\_elev\_m*) do *software FlightGear*<sup>®</sup> e para a escrita no *software MatLab/simulink*<sup>®</sup>. Até existir ordem de paragem, o programa “fg\_simulink” realiza uma rotina cíclica em que actualiza a variável (*gnd\_elev\_m*) e envia para o programa geral do *MatLab/simulink*<sup>®</sup>. A restante programação tem o intuito de conceber um painel de comando e configuração de forma a facilitar a utilização do mesmo. Na figura 3.21, apresenta-se o painel do programa “fg\_simulink” de leitura da altura do solo em relação ao nível médio do mar.

O programa “fg\_simulink” é desenvolvido em *MatLab/simulink*<sup>®</sup> no formato *m-file*, em que a comunicação é suportada por rede *Ethernet* no protocolo *User Datagram Protocol (UDP)*. Com o intuito de libertar o processamento do programa geral, o programa “fg\_simulink.” é compilado de forma a operar de forma independente, sem necessitar de utilizar recursos do *MatLab/simulink*<sup>®</sup>, possibilitando inclusive a execução do programa

noutro computador. No anexo B, disponibiliza-se a listagem de programação do programa auxiliar “fg\_simulink.exe”.



Figura 3.21 – Painel do programa “fg\_simulink.exe” de leitura da altura do solo em relação ao nível médio do mar

Para receber os dados da variável (gnd\_elev\_m) no *software MatLab/simulink*<sup>®</sup> é necessário utilizar o bloco de comunicação *UDP Receive Binary*. Apresenta-se, na figura 3.22, o bloco comunicação UDP em *MatLab/simulink*<sup>®</sup> de leitura da variável de altura do solo em relação ao nível médio do mar.

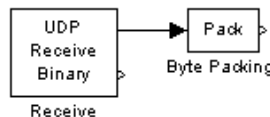


Figura 3.22 – Bloco de comunicação UDP em *MatLab/simulink*<sup>®</sup> de leitura da variável de altura do solo em relação ao nível médio do mar.

O bloco de comunicação *UDP Receive Binary* tem a função de receber dados numa determinada porta de comunicação e num endereço de IP. Este bloco é configurado de acordo com o tamanho da palavra em *bytes*, que neste caso são 8 *bytes* de resolução para a variável de altura do solo em relação ao nível médio do mar. O bloco *Pack* transforma os dados de *bytes* para variável decimal (Gng\_Alt) para ser utilizada no modelo da aeronave.

### 3.5 – Realidade Virtual

O desenvolvimento tecnológico dos microprocessadores faz com que os computadores sejam cada vez mais rápidos, permitido alcançar novos horizontes a preços acessíveis. Estes sistemas computacionais, quando ligados a uma boa *interface* gráfica, permitem ao utilizador observar ambientes idênticos aos reais e até experimentar sensações que por vezes não poderiam ser acessíveis na vida real. No universo dos gráficos computacionais não existem fronteiras nem restrições, podendo ser criados ambientes representativos do mundo real ou de um mundo imaginário, sendo estes manipulados pelo utilizador como bem entender. A realidade virtual surge naturalmente com o desenvolvimento dos sistema

computacionais, apresentando-se como elo de ligação para controlo e monitorização destes sistemas, apresentando vantagens quando utilizado nas simulações de situações reais em diversas áreas de conhecimento, [8].

### 3.5.1 – Conceitos Sobre Visualização

Nos simuladores de voo, os óculos de realidade virtual são dispositivos de *interface* humana com o intuito de levar o utilizador a abstrair-se do mundo real e ingressar num mundo virtual através da exibição de imagens. Os dispositivos de *interface* de realidade virtual têm características de interacção e imersão aos estímulos do ser humano, principalmente visuais, auditivos e tácteis.

Os sistemas de realidade virtual podem ser divididos em dispositivos de entrada e de saída. Os dispositivos de entrada são os responsáveis pelo envio da informação para os sistemas computacionais proveniente do utilizador através da *interface* humana como, por exemplo, o *joystick* e os pedais. Por outro lado, os dispositivos de saída são responsáveis pelos estímulos dos sentidos do utilizador, como, por exemplo, os óculos de realidade virtual e a plataforma electromecânica.

A amplitude visual do ser humano tem limitações, alcançando aproximadamente um máximo de  $180^\circ$  em ambas as direcções, vertical e horizontal. É designado de campo de visão (FOV - *Field Of View*) a extensão angular que pode ser visualizada pelo ser humano. No campo vertical é limitado inferiormente pelas bochechas e superiormente pelas sobranceiras para cerca de  $150^\circ$ . Enquanto o campo horizontal é limitado em  $60^\circ$  para o lado do nariz e  $90^\circ$  para o lado contrário ao nariz, perfazendo um total de  $150^\circ$  para cada olho. No seu conjunto, os dois olhos podem alcançar, assim,  $180^\circ$  horizontalmente com  $120^\circ$  de sobreposição binocular, quando focados no infinito, [24]. Apresenta-se, na figura 3.23, uma representação do campo de visão do ser humano.

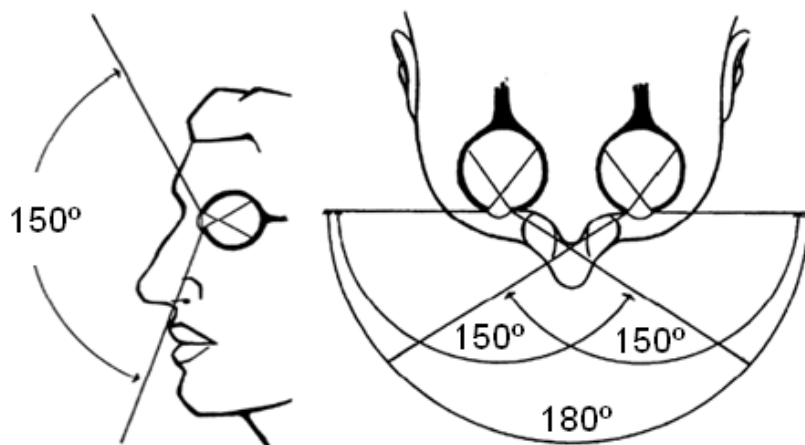


Figura 3.23 – Representação do campo de visão do ser humano, [24]

Por exemplo, um monitor de 21” observado a uma distância de 50cm abrange aproximadamente um campo de visão de 48°, enquanto uns óculos de realidade virtual abrangem um campo de visão de 40° a 60°, tipicamente. Alguns dispositivos com ecrã panorâmico podem chegar a 140° de campo de visão, [24].

Os óculos de realidade virtual podem ser monoscópicos ou estereoscópicos, permitindo este último a visualização de imagens a três dimensões. O conceito dos sistemas estereoscópicos baseia-se no facto do ser humano possuir dois olhos e destes estarem ligeiramente afastados um do outro, em que cada um vê imagens diferentes. Estas imagens são canalizadas para o cérebro humano construindo assim a noção de profundidade, distância, posição e tamanho dos objectos ou imagens visualizados. A distância entre as imagens visualizadas pelo olho esquerdo e pelo olho direito é chamada de paralaxe. Enquanto a distância horizontal entre pontos iguais da imagem da esquerda e da imagem da direita, quando sobrepostas na retina, é considerada como disparidade. A paralaxe que existe entre as duas imagens do par estereoscópico produzem disparidade nos olhos, o que permite ao cérebro compreender as imagens a três dimensões [8].

Existem quatro tipos de paralaxe utilizados nos sistemas estereoscópicos, nomeadamente: zero, divergente, positiva e negativa. Apresentam-se, na figura 3.24, os diferentes tipos de paralaxe.

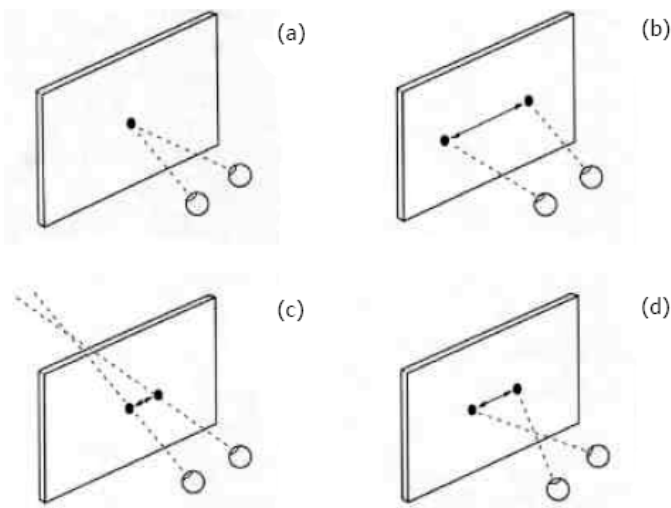


Figura 3.24 – Tipo de paralaxe: a) zero, b) divergente, c) positiva, d) negativa, [8]

Na paralaxe zero, os eixos oculares cruzam-se no plano de observação. Na paralaxe divergente, os eixos oculares nunca se cruzam, sendo esta uma situação que não se considera neste estudo, uma vez que não acontece no mundo real. Na paralaxe positiva, os eixos oculares cruzam-se atrás do plano de observação, dando noção de profundidade às imagens observadas. Por fim, na paralaxe negativa os eixos oculares cruzam-se à frente do

plano de observação, fazendo com que as imagens sobressaíam do plano de observação, [8].

Existem três formas para alcançar o efeito estereoscópico no ser humano, nomeadamente a forma passiva, a forma activa e a forma auto estereoscópica. Nos sistemas estereoscópios passivos são produzidas imagens sobrepostas em que o utilizador necessita de óculos com filtro passivos, podendo estes ser de polarização de luz ou de separação das cores do espectro, conhecido por anaglifo, [25]. Nos sistemas estereoscópios activos, o utilizador também necessita de óculos mas com capacidade de obstrução. Estes óculos, cujas lentes alternam entre o opaco e o transparente funcionam sincronizados com o ecrã onde são apresentadas as imagens. Ou seja, num primeiro momento o ecrã apresenta a imagem da esquerda e os óculos fecham a passagem de luz para o olho direito, permitindo que a imagem seja vista apenas pelo olho esquerdo, e, de forma quase simultânea, um segundo momento, o ecrã projecta a imagem da direita e os óculos fecham a passagem de luz para o olho esquerdo, obtendo-se o efeito estereoscópico, [25]. Os auto estereoscópicos são sistemas em que o utilizador não necessita utilizar óculos, uma vez que o ecrã possui uma película que direcciona os feixes luminosos de cada *pixel* para cada olho respectivo. Assim, o utilizador só vê com o olho direito a imagem destinada ao olho direito e o olho esquerdo só vê a imagem destinada ao olho esquerdo, [25].

### 3.5.2 – Técnicas de Produção de Pares Estereoscópicos

Existem três formas para a geração dos pares estereoscópicos, nomeadamente a projecção *off-axis*, a projecção *on-axis* e a rotação.

De acordo com [8], na projecção *off-axis*, assume-se a existência de dois centros de projecção, sendo a visão esquerda produzida com base no centro de projecção esquerdo e a visão direita produzida com base no centro de projecção direito. As vistas são implementadas com uma transformação de visualização e podem incluir a transformação de projecção em perspectiva.

Na projecção *on-axis*, assume-se a existência de um centro de projecção em conjunto com translações horizontais. Neste caso cada imagem do par estereoscópico é realizada através de translação da imagem para a direita ou para a esquerda, da projecção em perspectiva e em translação da imagem para o sentido contrário da primeira translação.

Na técnica de rotação, para se obterem as imagens no par estereoscópico é aplicada uma rotação vertical do centro de projecção.

Na técnica *off-axis* o campo de visão é superior 40% em relação à técnica *on-axis*. Apresenta-se, na figura 3.25, as projecções *off-axis* e *on-axis*, em que E representa o olho esquerdo e D o olho direito.

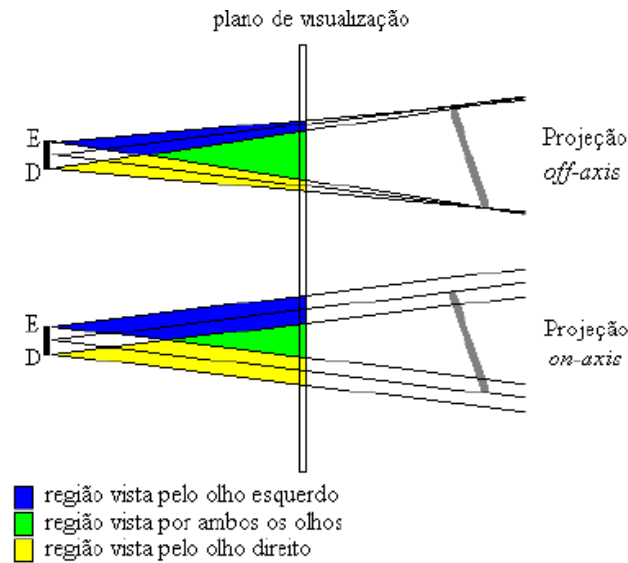


Figura 3.25 – Projecções *off-axis* e *on-axis*, [8]

Ainda de acordo com [8] e considerando a figura 3.26, na projecção *off-axis* implementada juntamente com a projecção em perspectiva, após a transformação de visualização tridimensional para bidimensional localiza-se o plano a ser visualizado paralelamente ao plano  $xy$ , passando pela origem.

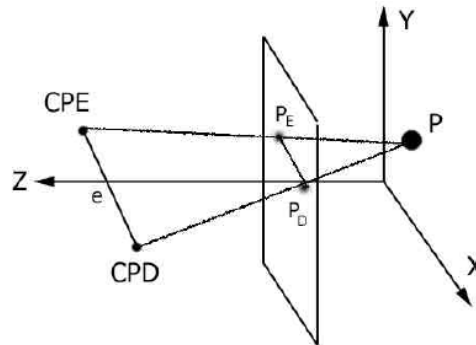


Figura 3.26 – Ponto P e suas projecções ( $P_E$  e  $P_D$ ) relativas aos centros CPE e CPD, [8]

Um centro de projecção padrão para o sistema de visualização monoscópico é então localizado no ponto  $(0,0,-d)$ , onde  $d$  é a distância do centro de projecção ao plano de projecção.

Sendo assim, a projecção de um ponto  $(x, y, z)$  passa a ter as coordenadas  $(x_p, y_p)$  no plano de projecção, onde:

$$x_p = \frac{xd}{d+z} \quad \text{e} \quad y_p = \frac{yd}{d+z} \quad (3.18)$$

Para produção das imagens do par estereoscópico na técnica *off-axis* é necessária a utilização de dois centros de projecção, um centro de projecção para a visão direita (CPD) e outro para a visão da esquerda (CPE) alinhada horizontalmente. O CPD é movido perpendicularmente ao eixo  $z$  para a posição  $(e/2, 0, -d)$  e o CPE para  $(-e/2, 0, d)$ , onde “ $e$ ” representa a separação total entre os centros de projecção.

Concluindo, para um ponto arbitrário  $P = (x, y, z)$  a projecção para a visão esquerda  $PE = (xE, yE)$ , em coordenadas no plano de projecção é dada pelas expressões:

$$xE = \frac{xd - \frac{ze}{2}}{d + z} \quad \text{e} \quad yE = \frac{yd}{d + z} \quad (3.19)$$

E a projecção de P para a visão direita,  $PD = (xD, yD)$ , tem os valores de coordenadas no plano de projecção é dada pelas expressões:

$$xD = \frac{xd + \frac{ze}{2}}{d + z} \quad \text{e} \quad yD = \frac{yd}{d + z} \quad (3.20)$$

### 3.5.3 – Dispositivos de Visualização

Os óculos de realidade virtual ou ecrã montados sobre a cabeça (HMD - *Head Mounted Display*) são geralmente constituídos por dois pequenos monitores, um para cada olho. Assim, cada um pode exibir a imagem específica para cada olho, alcançando-se desta forma o efeito estereoscópico. Existem alguns sistemas designados de realidade virtual aumentada, que consistem em acrescentar à imagem real visualizada pelo ser humano, informação extra, como por exemplo, distância, velocidade, informação sobre um determinado objecto, entre outros. Neste caso, pode ser utilizado um único ecrã para apenas um dos olhos ou então, um sistema de projecção da informação nas lentes transparentes dos óculos para esse efeito, [24].

Existem vários tipos de óculos de realidade virtual no mercado com o intuito de responder a diferentes aplicações, nomeadamente no âmbito lúdico, científico ou militar. No que respeita às aplicações lúdicas, por vezes não se justificam grandes investimentos, daí que características como a resolução e taxa de actualização serem por vezes limitadas. Nas aplicações científicas e militares, justifica-se o investimento em óculos de realidade virtual com um grau de qualidade e características superiores, contendo resoluções e taxas de actualização superiores. Por exemplo, no caso das aplicações militares, estas podem proporcionar ao soldado informações táticas, mapas, distâncias, etc. Relativamente às

aplicações lúdicas, estas podem ser utilizadas em videojogos ou na visualização de filmes estereoscópicos (3D) ou não estereoscópicos (2D). No que se refere ao âmbito científico, considera-se a sua aplicação a diversas áreas como a medicina e a engenharia, [8].

Os pequenos monitores, ao ficarem junto aos olhos, concedem ao utilizador a sensação de estar a visualizar um monitor de dimensão considerável. Por exemplo, os óculos utilizados neste trabalho, são da marca *eMagin*<sup>®</sup> no modelo z800, de acordo com a figura 3.27. Segundo o fabricante este dispositivo possibilita ao utilizador a sensação de estar perante um ecrã de 105”, à distância de 3m.



Figura 3.27 – Óculos de realidade virtual *eMagin*<sup>®</sup> no modelo z800, [26]

Alguns óculos de realidade virtual (exemplo *eMagin*<sup>®</sup> z800) têm incorporado sensores de movimento a fim de detectarem a posição da cabeça. Assim, estes óculos, quando aplicados a um simulador permitem uma maior interacção com o ambiente gráfico, de acordo com os movimentos da cabeça. No caso do simulador de voo a utilização deste tipo de óculos de realidade virtual dá a sensação de se estar dentro do *cockpit* da aeronave, pois, quando se roda a cabeça, a imagem visualizada acompanha os mesmos movimentos, na mesma proporção e no mesmo sentido. Por exemplo, quando se olha para um dos lados, é possível visualizar-se toda a parte lateral da aeronave, incluindo a janela do *cockpit*, ou, quando se olha em frente, pode-se visualizar o painel de instrumentos e o pára-brisas da aeronave. Desta forma, obtém-se um nível de imersão maior, alcançando-se assim uma maior interacção com o ambiente gráfico, quando comparado a outros óculos sem sensores de movimento.

Existem características que são fundamentais ter presentes na escolha de uns óculos de realidade virtual, como, por exemplo, o campo de visão, a resolução, a taxa de actualização, o contraste, a latência e a sincronização. Na tabela 3.8, apresenta-se, algumas características de óculos de realidade virtual.

As características que se devem ter em consideração para a utilização em sistema de simulação são a latência, que mede o tempo de reacção do sistema quando solicitada uma ordem na entrada, e a sincronização entre os diversos dispositivos.

Tabela 3.8 – Características de óculos de realidade virtual

Marca	Modelo	FOV	Resolução	Cores	Contraste	Sensores	Áudio
EMD	i-theater	32°d	640x480	24bits	-	n	sim
i-O Display	i-glasse 3D	26°d	800x600	24bits	-	n	sim
eMagin	Z800	40°h	800x600	24bits	200:1	sim	sim
Sensics	sight	60°h	1280x1024	24bits	800:1	sim	sim
Sensics	pisight	179°h	2600x1200	24bits	800:1	opcional	sim

A latência geralmente ocorre devido a atrasos no processamento das imagens e no tempo de comunicação entre os dispositivos. Em latências abaixo de 100ms, não existem grandes consequências na *performance* do simulador, [24]. A taxa de actualização do ecrã deve ser a maior possível, para que não seja perceptível para o utilizador. Assim, considera-se que valores na ordem de 15Hz são suficientes para satisfazer os sentidos do ser humano. No entanto, para movimentos rápidos é preferível que a taxa de actualização seja superior a 60Hz, [24]. O acoplamento destes sistemas à plataformas de simulação, deve-se ter em conta a sincronização, pois, é fundamental que as imagens visualizadas nos óculos de realidade virtual e as sensações provocadas pelos movimentos da plataforma de simulação cheguem, ao mesmo tempo, ao cérebro humano. Caso contrário, pode provocar desconforto e desorientação ao utilizador.

Os óculos de realidade virtual utilizados neste trabalho incluem um sensor de movimento do tipo magnético, no qual a posição medida é do tipo relativa uma vez que é definido pelo utilizador uma referência de zero, através do *software* fornecido pelos óculos de realidade virtual. Os movimentos detectados são de rotação da cabeça do utilizador, nomeadamente *Pitch* e *Roll*.

Os óculos eMagin z800 apresentam algumas limitações, nomeadamente no que diz respeito ao sensor de movimentos da cabeça do utilizador. Quando se calibra a referência de zero do sensor, este deveria fixar essa posição como estado inicial, mas tal não acontece. Este problema deve-se ao facto do sensor ser do tipo magnético, porque todo o sistema da plataforma electromecânica contém muitas peças metálicas e por existir muito ruído electromagnético envolvente, [24]. Apesar do sistema de visualização funcionar com as limitações descritas, foi possível verificar por momentos o seu funcionamento na interacção com o ambiente gráfico dos óculos de realidade virtual.

## 4 – Implementação

### 4.1 – Introdução

Neste capítulo descrevem-se os aspectos de programação em *software MatLab/simulink*<sup>®</sup> para o funcionamento do simulador da aeronave Cessna-172, nomeadamente a interligação entre a interface humana *joystick*, o modelo matemático da aeronave, o modelo da cinemática inversa da plataforma electromecânica, a interface humana óculos de realidade virtual e os conversores de potência da plataforma electromecânica. Na figura 4.1, apresenta-se o diagrama de blocos “geral” do simulador da aeronave Cessna-172.

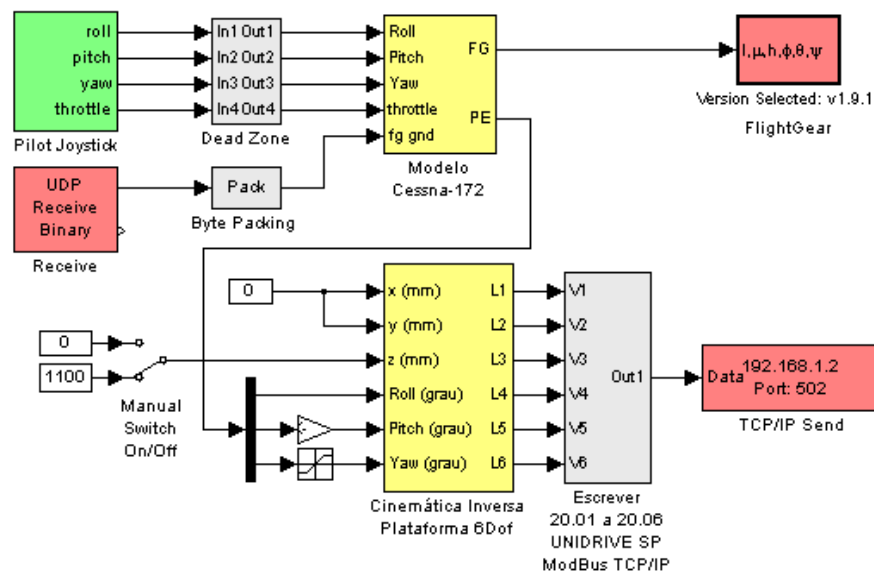


Figura 4.1 – Diagrama de blocos “geral” do simulador da aeronave Cessna-172

A interface humana *joystick* utiliza comunicação do tipo USB (*Universal Serial Bus*) para a ligação ao computador onde se encontra o programa desenvolvido em *MatLab/simulink*<sup>®</sup>. Por outro lado, a ligação entre este último com *software FlightGear*<sup>®</sup> e os conversores de potência da plataforma electromecânica é realizado por rede *Ethernet*. Na figura 4.2, apresenta-se um diagrama de blocos representando a comunicação utilizada entre dispositivos, onde VGA representa *Video Graphics Array*.

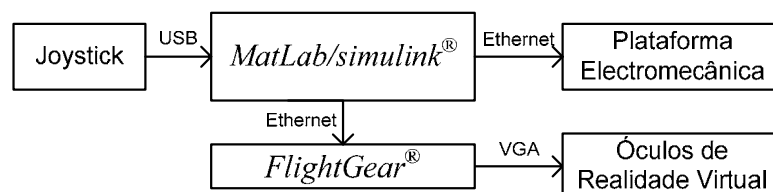


Figura 4.2 – Diagrama de blocos representando a comunicação utilizada entre dispositivos

Apresentam-se ainda os ensaios laboratoriais, a análise de resultados das simulações do modelo matemático da aeronave Cessna-172 e da cinemática inversa da plataforma electromecânica utilizando um sensor inercial colocado no topo da plataforma.

## 4.2 – Equipamento e *Software*

Para o funcionamento do simulador é necessário interligar os diversos dispositivos físicos e canalizar a informação para o computador onde se encontra todo o *software* desenvolvido para o simulador. Para que a transmissão de dados entre dispositivos e computador seja realizada no menor tempo possível, foi desenvolvida programação complementar (ModBus TCP/IP) e incorporada no modelo do sistema em estudo, de forma a minimizar a latência de todo o sistema.

### 4.2.1 – *Joystick*

Os movimentos dos comandos do simulador afectam directamente as entradas do modelo matemático Cessna-172 através da interface humana *Joystick*. Para tal, em computadores que usam por base sistemas operativos da *Microsoft*<sup>®</sup>, utiliza-se o bloco *Pilot Joystick* da livraria *MatLab/simulink*<sup>®</sup>. Neste bloco tem-se a possibilidade de seleccionar o *joystick ID* que se pretende utilizar (para o caso de existir mais que um dispositivo de entrada instalado no computador), o tipo de saída (quatro eixos ou todas as saídas disponíveis) e o tempo de amostragem.

Optou-se por uma saída a quatro eixos pelo facto de estes terem a informação necessária para satisfazer a entrada do bloco do modelo matemático Cessna-172.

O bloco *Pilot Joystick* apresenta nas suas saídas valores de amplitude compreendidos entre -1 e 1, tendo valor zero para as posições centrais da *manche* e dos pedais, enquanto no *Throttle* (acelerador) é de 0 a 1. Todos os comandos efectuados pelo utilizador afectam directamente as quatro saídas de forma linear. A posição da *manche* afecta a saída de *Roll* e *Pitch*, designados no dispositivo físico por eixos X e Y, a posição dos pedais afecta a saída *Yaw*, designados no dispositivo físico de eixo R, e a posição do acelerador afecta a saída de *Throttle*, designado no dispositivo físico de eixo Z. Apresenta-se, na figura 4.3, a interface humana *joystick* de entrada do simulador.

O *Joystick* utilizado apresenta algumas limitações nos movimentos, nomeadamente nos movimentos da *manche*, uma vez que estes dão a sensação de pouca resistência ao movimento provocado pelo utilizador, fazendo com que uma pequena força provoque uma alteração na posição da *manche* que, por sua vez, transmite essa variação de posição ao

modelo matemático e, conseqüentemente, provoca movimento na plataforma electromecânica. Apesar de existir um ajuste físico para aumentar a resistência ao movimento, este demonstrou-se ser insuficiente. É de notar que a força necessária, que o utilizador deveria realizar, para mover qualquer mecanismo de comando deverá ser idêntica ao experimentado na aeronave real.

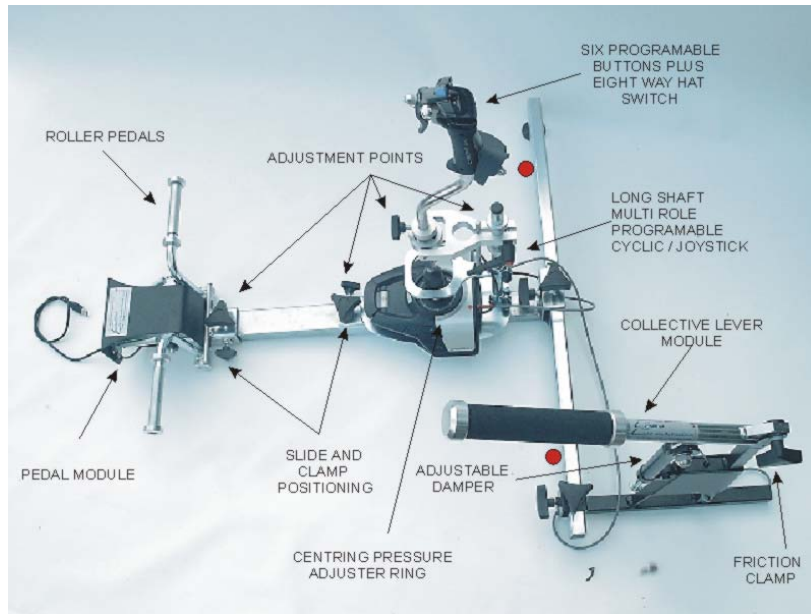


Figura 4.3 – Interface humana: joystick de entrada do simulador, [27]

De forma a obviar esse problema na interface humana *Joystick* foi inserido o bloco *DeadZone* entre o bloco *Pilot Joystick* e o bloco Modelo Cessna-172. Este Bloco tem a função de conceber uma zona neutra garantindo o valor zero na saída, numa determinada zona central da posição da *manche*. Apresenta-se na figura 4.4 o diagrama interno do bloco *DeadZone*.

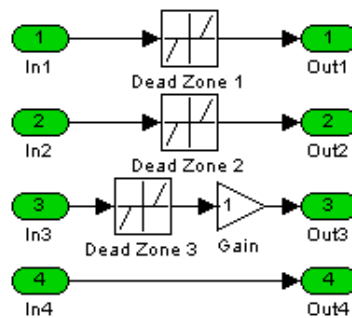


Figura 4.4 – Diagrama interno do bloco *DeadZone*

De forma empírica, e tendo em conta a afinação física do joystick no simulador, foram testados diversos valores para definir essa zona de saída igual a zero para os diferentes movimentos. Na tabela 4.1, apresentam-se os valores que foram considerados satisfatórios para o bloco *DeadZone*.

Tabela 4.1 – Valores utilizados para o bloco DeadZone

Entrada	Movimento	Mínimo	Máximo
In1 - Out1	Roll	-0,3	0,3
In2 - Out2	Pitch	-0,3	0,3
In3 - Out3	Yaw	-0,2	0,2
In4 - Out4	Throttle	-	-

#### 4.2.2 – Modelo Cessna-172

No diagrama de blocos “geral”, o bloco Modelo Cessna-172 é responsável pela reacção de grande parte do sistema, uma vez que é neste que se encontram as expressões matemáticas que traduzem os movimentos de reacção da aeronave em termos de posição e orientação, directamente relacionados com as ordens do comando de entrada de *Roll* (X), *Pitch* (Y), *Yaw* (R) e *Throttle* (Z).

A saída FG representa um vector constituído pela posição em coordenadas geográficas e pelos ângulos de Euler da aeronave, nomeadamente Latitude, Longitude, Altitude, *Roll*, *Pitch* e *Yaw*, perfazendo os seis graus de liberdade necessários à definição dos movimentos da aeronave em ambiente gráfico. Por outro lado, a saída PE representa um vector constituído pelos ângulos de Euler que representam a orientação da aeronave em termos de *Roll*, *Pitch* e *Yaw*. Devido a limitações físicas da plataforma electromecânica, em termos de amplitudes dos movimentos, não foram considerados os movimentos de *Heave*, *Surge* e *Sway* na simulação da Latitude, Longitude e Altitude, uma vez que teoricamente seria desejável que a plataforma permitisse amplitudes infinitas de movimentos. Na tabela 4.2 apresentam-se as características da plataforma electromecânica.

Tabela 4.2 – Características dos movimentos da plataforma electromecânica, [15]

Movimento	Amplitude	Velocidade	Aceleração
Pitch	24°	65 °/s	165 °/s <sup>2</sup>
Roll	20°	70 °/s	175 °/ s <sup>2</sup>
Yaw	28°	100 °/s	100 °/ s <sup>2</sup>
Heave	0,3m	0,50 m/s	1,30 m/ s <sup>2</sup>
Surge	0,26m	0,80 m/s	2,00 m/ s <sup>2</sup>
Sway	0,22m	0,70 m/s	1,80 m/ s <sup>2</sup>

No entanto, pelo facto de se utilizar apenas os ângulos de Euler para os movimentos da plataforma electromecânica, não foi prejudicada a simulação da aeronave, uma vez que a posição geográfica dada pela Latitude, Longitude e Altitude da aeronave são consideradas no ambiente gráfico, transmitindo desta forma ao utilizador a sensação de alteração da localização geográfica.

### 4.2.3 – Óculos de Realidade Virtual

Os óculos de realidade virtual são ligados ao computador onde se encontra o *software FlightGear*<sup>®</sup>. Neste sentido, torna-se necessário instalar os *drivers* e o *software (Z800 3DVisor Software Utility)* que são parte integrante dos óculos de realidade virtual. Com o *software* dos óculos de realidade virtual calibra-se a posição de referência zero para o sensor de movimento da cabeça do utilizador, a sensibilidade do sensor, ajusta-se o contraste, brilho e a taxa de actualização. Também existe a possibilidade de desactivar o sensor de posição da cabeça do utilizador.

A ligação aos óculos de realidade virtual é realizada por uma porta de comunicação do tipo USB, que para além de permitir a ligação ao *software* de configuração acima descrito, tem também a função de alimentar o equipamento. Para o sinal de vídeo, utiliza-se a saída VGA auxiliar do computador.

### 4.2.4 – Plataforma Electromecânica

Para se aplicar o modelo matemático da cinemática inversa torna-se indispensável conhecer as características físicas da plataforma electromecânica e ter em atenção a programação desenvolvida nos conversores de potência, nomeadamente a relação entre as amplitudes de referência de posição e as amplitudes de deslocação das pernas da plataforma electromecânica, de forma a parametrizar o ganho  $G_s$  de saída do bloco. Na tabela 4.3, apresentam-se os parâmetros do bloco cinemática inversa plataforma 6DOF.

Tabela 4.3 – Parâmetros do bloco Cinemática inversa plataforma 6DOF, [15]

Distância entre juntas (mm)	
a	180
b	1200
c	180
d	210
e	220
f	690
Lmin	750
$G_s$	10

Depois de definidas as constantes do bloco Cinemática inversa plataforma 6DOF, ajustam-se os referenciais de saída do modelo matemático Cessna-172 para o referencial definido pelo modelo da cinemática inversa da plataforma electromecânica, com o objectivo de fazer coincidir os sentidos de rotação da plataforma electromecânica com o sentido da saída do modelo matemático Cessna-172 e com as *interfaces* humanas, nomeadamente *joystick* e as imagens geradas nos óculos de realidade virtual. Por outro lado, deve-se, quando necessário, limitar as amplitudes dos movimentos da plataforma

electromecânica, garantindo espaço de manobra e maior amplitude nos movimentos que se consideram de maior importância. Neste caso, foi necessário inverter o sinal do movimento *Pitch*. Optou-se por limitar o movimento *Yaw* e anular os movimentos de *Surge* (x) e *Sway* (y) pelas razões descritas em 4.4.2. O movimento de *Heave* (z) foi utilizado para elevar a plataforma da posição de repouso para a posição inicial de funcionamento, enquanto os movimentos *Pitch* e *Roll* não têm restrições.

Após calculadas as distâncias das variáveis  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$ ,  $L_5$  e  $L_6$  (comprimento das pernas da plataforma electromecânica), procede-se ao tratamento dos dados, com o objectivo de enviar essa informação pela rede *Ethernet* e, assim, disponibilizar os valores correspondentes ao deslocamento das pernas da plataforma electromecânica no conversor de potência *Unidrive SP (Master)*. O protocolo de comunicação utilizado pelo conversor de potência *Unidrive SP* é do tipo ModBus TCP/IP. Este conversor de potência apenas tem a função de distribuir as referências de posição para os seis conversores de potência (*Slaves*), utilizando uma rede de comunicação própria do fabricante, designada CTnet.

A CTnet consiste numa rede de comunicação de elevada velocidade (5Mbits/s) que permite transferência rápida de dados entre os dispositivos nela ligados. Por esse motivo é possível obter-se uma taxa de actualização nos seis conversores de potência, aproximadamente de 2ms, [29].

Os conversores de potência são responsáveis pelo controlo de posição dos actuadores electromecânicos da plataforma electromecânica. Uma vez que não existe nenhum bloco na livraria *MatLab/simulink*<sup>®</sup> que suporte directamente este tipo de comunicação, houve a necessidade de construir o bloco ModBus TCP/IP, com o objectivo de estruturar a mensagem em *bytes* no protocolo ModBus e enviar por TCP/IP utilizando o bloco TCP Send existente na livraria *MatLab/simulink*<sup>®</sup>.

Para a implementação do bloco ModBus TCP/IP foi necessário estruturar o *Protocol Data Unit* (PDU), [30] e [31].

Define-se de PDU a estrutura da mensagem codificada em protocolo ModBus. Sendo esta independente do meio físico de transmissão é constituída pela função e pelos dados. Em ModBus TCP/IP é necessário acrescentar ao PDU um cabeçalho próprio, designado de *ModBus Application Protocol Header* (MBAP Header). O conjunto MBAP e PDU designa-se *Application Data Unit* (ADU). Na tabela 4.4, apresenta-se a estrutura da mensagem em ModBus TCP/IP.

Tabela 4.4 – Estrutura da mensagem em ModBus TCP/IP ADU

<i>ModBus TCP/IP ADU</i>					
<i>ModBus Application Protocol Header (MBAP Header)</i>				<i>Protocol Data Unit (PDU)</i>	
<i>Transaction ID</i>	<i>Protocol ID</i>	<i>Length</i>	<i>ID Slave</i>	<i>Função</i>	<i>Dados</i>
<i>2 Bytes</i>	<i>2 Bytes</i>	<i>2 Bytes</i>	<i>1 Byte</i>	<i>1 Byte</i>	<i>Vários Bytes</i>

No protocolo ModBus, existem vários tipos de funções, nomeadamente as funções definidas pela comunidade ModBus (*Public Functions Code*), as quais são normalizadas e utilizadas pela maior parte dos fabricantes. Existem também as funções definidas pelo próprio utilizador (*User Defined Function Codes*) e as funções específicas de alguns fabricantes (*Reserved Function Codes*), [31]. O conversor de potência do fabricante *Unidrive SP* utilizado neste trabalho, tem à disposição as funções ModBus apresentadas na tabela 4.5.

Tabela 4.5 – Funções ModBus disponíveis no conversor de potência *Unidrive SP*, [28]

Função	Descrição
3	Ler vários registos de 16bits
6	Escrever um registo de 16bits
16	Escrever vários registos de 16bits
23	Ler e escrever vários registos de 16bits

Nos conversores de potência *Unidrive SP*, para além da utilização das funções normalizadas, existem também funções secundárias próprias do fabricante colocadas nos *bytes* iniciais dos dados, com o objectivo de encaminhar a mensagem para uma determinada posição de memória, [28].

Por fim, depois de estruturada a mensagem *ModBus TCP/IP ADU*, utiliza-se o bloco TCP Send, da livreria *MatLab/simulink*<sup>®</sup>, que tem por função encaminhar a mensagem a um destinatário identificado por um endereço de IP e por uma porta de comunicação na rede *Ethernet*. Neste bloco, configura-se o endereço de IP, a Porta de comunicação e o tempo de actualização entre mensagens. Em relação à mensagem *ModBus TCP/IP ADU*, o bloco TCP Send adiciona, no início da mensagem, o endereçamento necessário e, no fim, um código *Checksum* (CRC) de verificação de erros. Na figura 4.6, apresenta-se a estrutura do protocolo TCP/IP utilizado como suporte à rede *Ethernet*.

Tabela 4.6 – Estrutura do protocolo TCP/IP utilizado como suporte à rede *Ethernet*

<i>Ethernet Header</i>	<i>IP Header</i>	<i>TCP Header</i>	<i>ModBus TCP/IP ADU</i>	<i>CRC</i>
<i>14 Bytes</i>	<i>20 Bytes</i>	<i>20 Bytes</i>	<i>Vários Bytes</i>	<i>4 Bytes</i>

Desta forma, considerando as variáveis  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$ ,  $L_5$  e  $L_6$ , na tabela 4.7, apresenta-se a estrutura da mensagem *ModBus TCP/IP ADU* com indicação do valor dos *bytes* para o conversor de potência *Unidrive SP*.

Tabela 4.7 – Estrutura da mensagem *ModBus TCP/IP* ADU com indicação do valor dos *bytes* para o conversor de potência Unidrive SP

ModBus Application Protocol Header (MBAP Header)																	
Transaction ID		Protocol ID		Length		ID Slave											
High	Low	High	Low	High	Low	-											
0	2	0	0	0	19	1											
Protocol Data Unit (PDU)																	
Função		Dados															
16	7	208	0	6	12	H1	L1	H2	L2	H3	L3	H4	L4	H5	L5	H6	L6

Em que os 5 *bytes* iniciais dos Dados são definidos pela seguinte ordem: *High byte*, endereço de início do *ModBus* (7); *Low byte*, endereço de início do *ModBus* (208); *High byte*, Número de registos *ModBus* (0); *Low byte*, Número de registos *ModBus* (6); Número de *bytes* dos dados seguintes *ModBus* (12).

Uma vez que a comunicação é realizada em *bytes* e as variáveis L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, L<sub>4</sub>, L<sub>5</sub> e L<sub>6</sub> podem assumir qualquer valor numérico entre 0 e 2500, há que as transformar e dividir em *bytes*. Sabendo-se que cada *byte* são 8 *bits*, representando no máximo um valor numérico até 256, e que o conversor de potência *Unidrive SP* recebe dados em 16 *bits*, ou seja, uma *Word*, há que converter cada variável para 2 *bytes*. Como exemplo, na tabela 4.7, a variável L<sub>1</sub>, representa-se por H1 o *High Byte* e por L1 *Low Byte* na estrutura da mensagem PDU.

### 4.3 – Simulações do Modelo Matemático Cessna-172

As simulações do modelo matemático Cessna-172 foram realizadas utilizando-se o programa *MatLab/Simulink*<sup>®</sup>, com o intuito de conferir o comportamento do modelo para sinais de entrada, de forma quadrada e sinusoidal, obtidos à custa de um gerador de funções próprio do *software MatLab/Simulink*<sup>®</sup>. Na figura 4.5, apresenta-se o diagrama de blocos utilizado para simulação do modelo matemático Cessna-172.

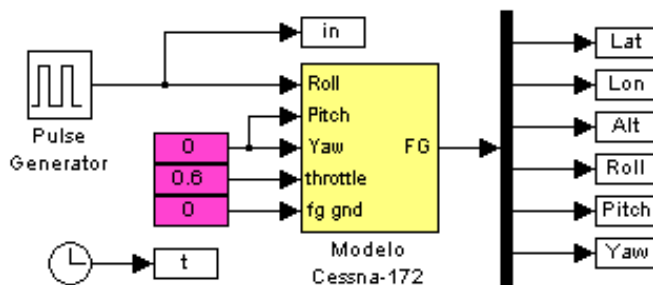


Figura 4.5 – Diagrama de bloco utilizado para simulação do modelo matemático Cessna-172

Como condições iniciais para a realização das simulações consideraram-se os valores apresentados na tabela 4.8 para a parametrização do bloco interno Cessna-172.

Tabela 4.8 – Condições iniciais para a simulação do modelo Cessna-172

Parâmetro	Valor
Velocidade inicial	[50 50 0]
<i>Initial angular rate</i>	[0 0 0]
<i>Initial quaternions</i>	[1 0 0 0]
Posição inicial	[40*pi/180 -10*pi/180 1000]
Rotação inicial do motor	1500*pi/30
Combustível inicial	[50 50]

Para as simulações foram considerados os movimentos individuais de *Roll*, *Pitch*, *Yaw* e *Throttle*, considerando 0m para a altura do solo em relação ao nível médio do mar (fg\_gnd). É de salientar que em todos os movimentos simulados, excepto o de *Throttle*, foi considerado o valor de 0,6, com o intuito de garantir uma propulsão mínima necessária para manter a aeronave em condições estáveis, para a realização das manobras solicitadas pelas outras entradas.

Na figura 4.6, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de 0,5, período de 100s e com razão cíclica a 50% para a entrada “Roll” do Modelo Cessna-172.

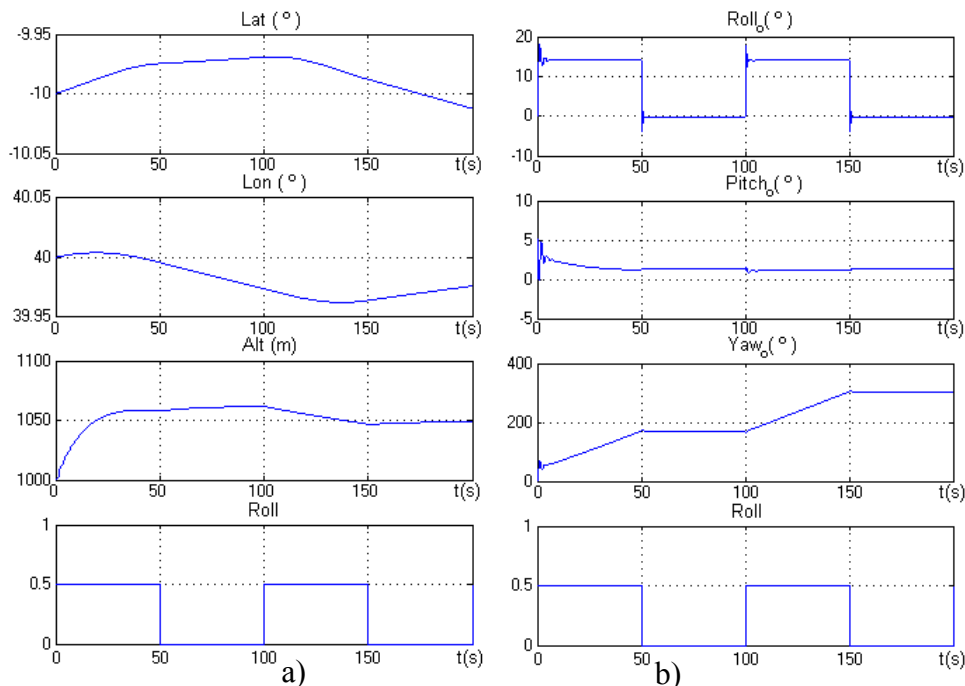


Figura 4.6 – Resposta temporal do Modelo Cessna-172 para um movimento *Roll* do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.7, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de 0,5 e período de 100s para a entrada “Roll” do Modelo Cessna-172.

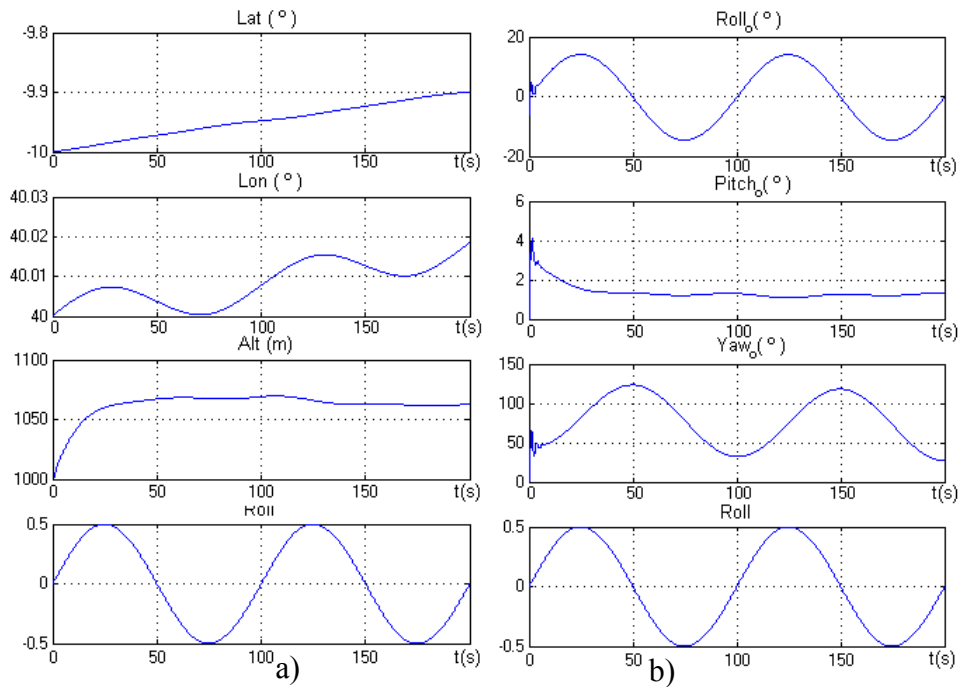


Figura 4.7 – Resposta temporal do Modelo Cessna-172 para um movimento *Roll* do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.8, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de 0,5, período de 100s e com razão cíclica a 50% para a entrada “Pitch” do Modelo Cessna-172.

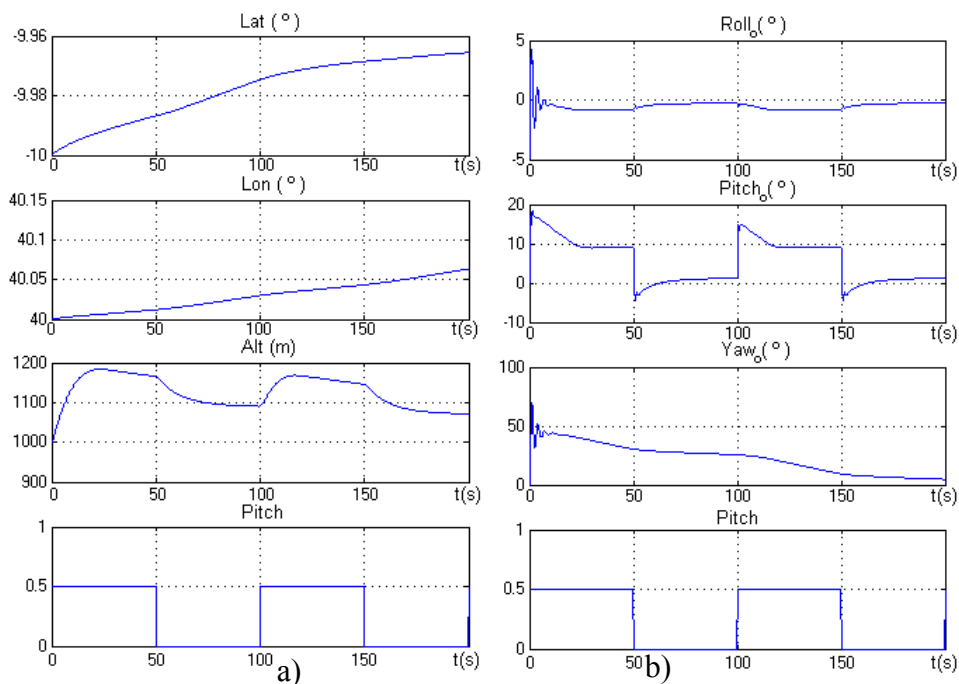


Figura 4.8 – Resposta temporal do Modelo Cessna-172 para um movimento *Pitch* do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.9, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de 0,5 e período de 100s para a entrada “Pitch” do Modelo Cessna-172.

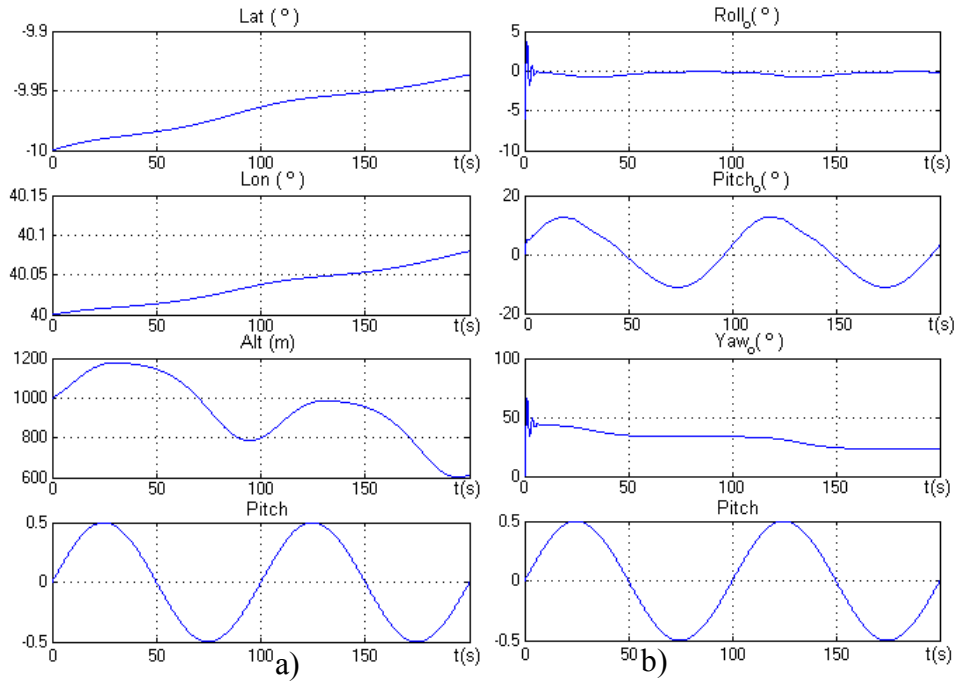


Figura 4.9 – Resposta temporal do Modelo Cessna-172 para um movimento *Pitch* do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.10, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de 0,5, período de 100s e com razão cíclica a 50% para a entrada “Yaw” do Modelo Cessna-172.

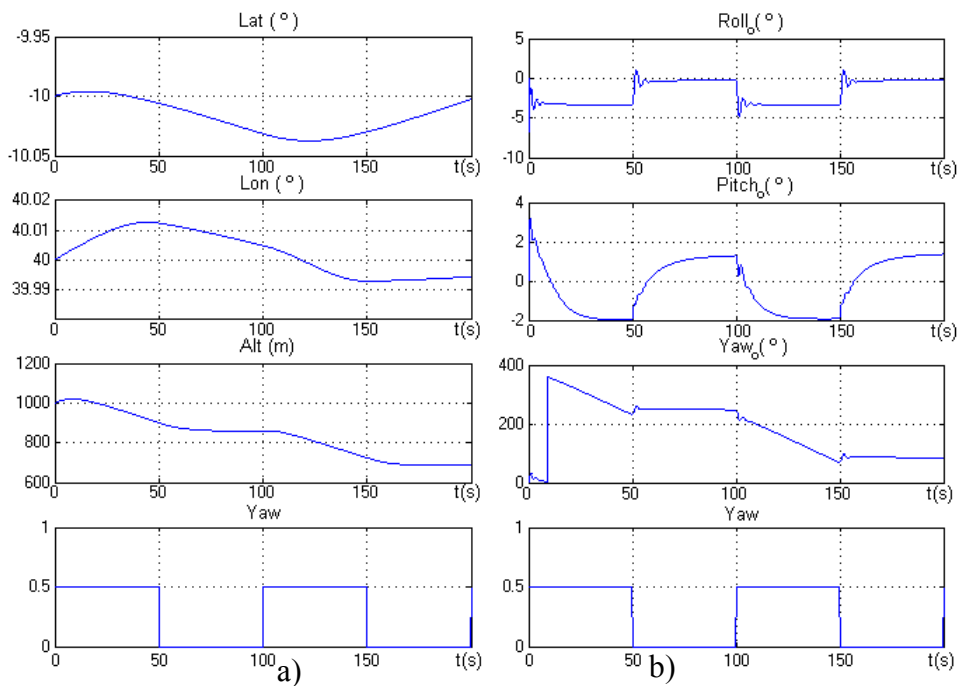


Figura 4.10 – Resposta temporal do Modelo Cessna-172 para um movimento *Yaw* do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.11, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de 0,5 e período de 100s para a entrada “Yaw” do Modelo Cessna-172.

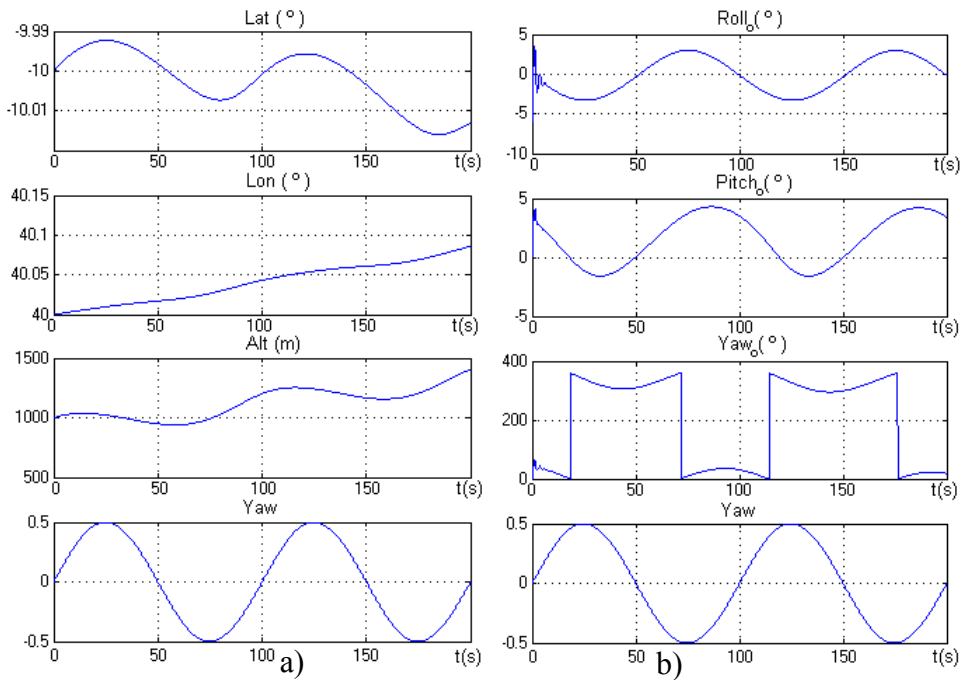


Figura 4.11 – Resposta temporal do Modelo Cessna-172 para um movimento *Yaw* do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.12, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de 1, período de 100s e com razão cíclica a 50% para a entrada “Throttle” do Modelo Cessna-172.

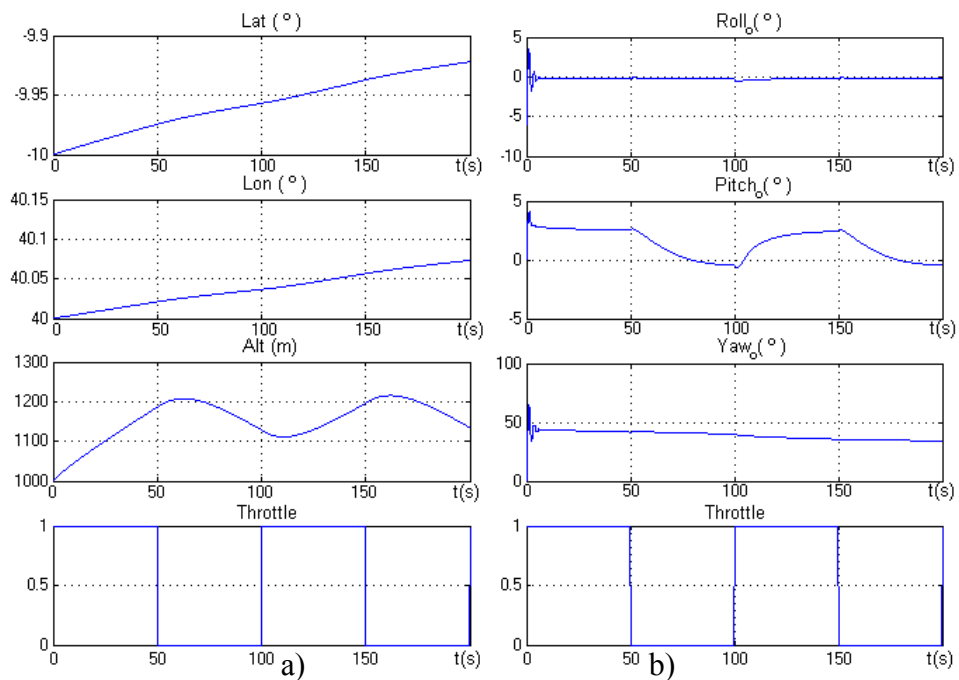


Figura 4.12 – Resposta temporal do Modelo Cessna-172 para um movimento *Throttle* do tipo quadrada. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Na figura 4.13, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de 0,5 e período de 100s para a entrada “Throttle” do Modelo Cessna-172.

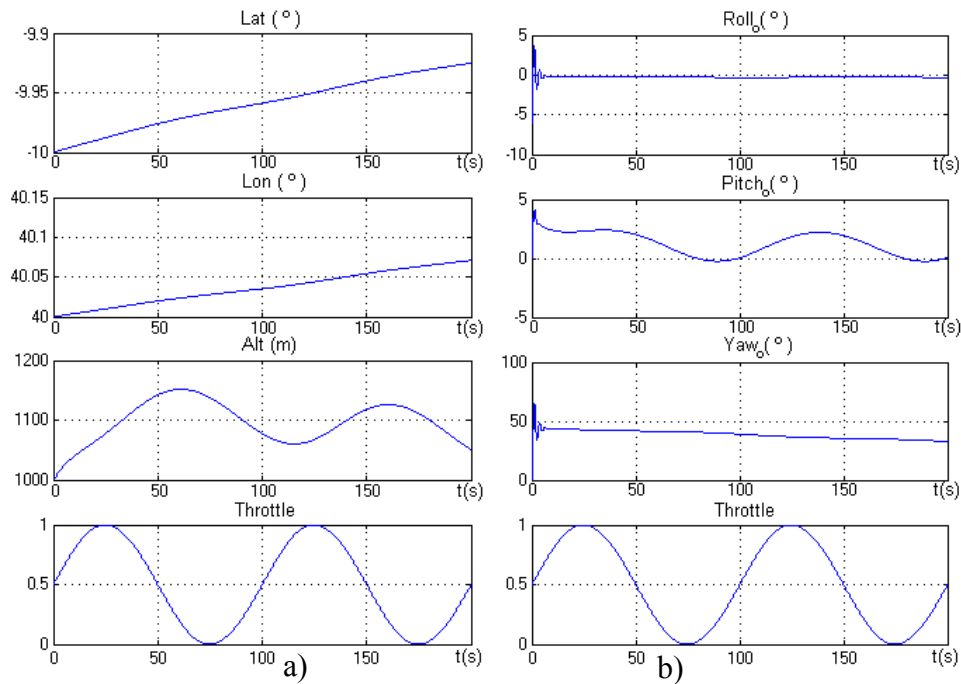


Figura 4.13 – Resposta temporal do Modelo Cessna-172 para um movimento *Throttle* do tipo sinusoidal. Posição Geográfica da aeronave (a) e ângulos de Euler (b) respectivamente

Analisando as figuras 4.6 a 4.13, verifica-se que a resposta do modelo matemático da aeronave Cessna-172 face às entradas propostas corresponde de forma análoga às respostas do *software FlightGear*<sup>®</sup>, em modo videojogo, e do *software Microsoft Flight Simulator*, este último utilizado apenas como *software* de apoio à verificação do comportamento de aeronaves. Como não foi possível comprovar os valores resultantes da simulação com valores eventualmente medidos em uma aeronave real, assume-se como adequada a implementação do modelo em *MatLab/simulink*<sup>®</sup>, tendo em conta os resultados apresentados face aos verificados no *software FlightGear*<sup>®</sup> que neste caso, foi utilizado como referência.

De referir que o modelo matemático da aeronave Cessna-172 implementado em *MatLab/simulink*<sup>®</sup> é o mesmo que se encontra incorporado no *software FlightGear*<sup>®</sup>, em modo videojogo. Por conseguinte, os resultados de simulação apresentam o mesmo comportamento em ambos os *software* utilizados.

#### 4.4 – Ensaios Laboratoriais

Com o intuito de validar o modelo da Cinemática inversa, foram realizados ensaios à plataforma electromecânica. Para tal, realizou-se um diagrama de blocos em *software*

*MatLab/Simulink*® para esse propósito. Na figura 4.14, apresenta-se o diagrama de blocos da cinemática inversa para comando da plataforma electromecânica.

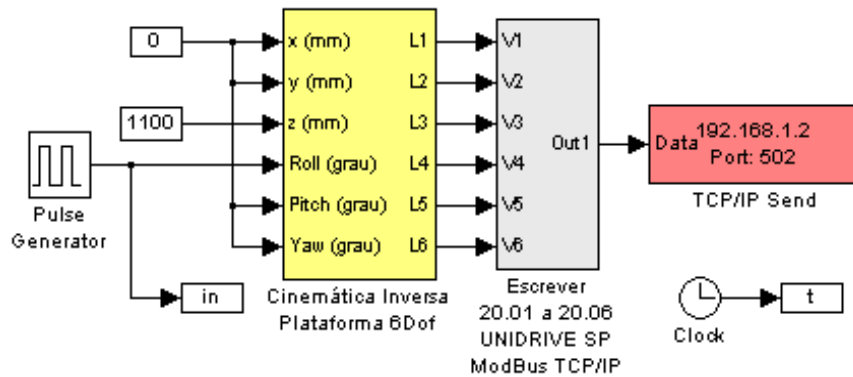


Figura 4.14 – Diagrama de bloco da cinemática inversa para comando plataforma electromecânica

Para a análise do comportamento da plataforma electromecânica foi considerado um gerador de funções, com sinais do tipo quadrado e sinusoidal, aplicado aos movimentos *Roll*, *Pitch* e *Yaw* das respectivas entradas do modelo matemático da Cinemática Inversa. O software *MatLab/simulink*® foi configurado para funcionar em modo *pseudo real time*, isto é, a simulação é realizada sincronizadamente com o relógio do computador, ou seja, por exemplo, 60s de simulação correspondem a 60s em tempo real.

Para confirmar os movimentos da plataforma electromecânica, foi instalado um sensor inercial com o objectivo de medir a orientação desta e, assim, validar os respectivos movimentos. Na figura 4.15, apresenta-se o sensor inercial *Xsens*® Mti.



Figura 4.15 – Sensor inercial *Xsens*® Mti, [33]

O sensor *Xsens*® Mti tem incorporado acelerómetros, giroscópios e magnetômetros nos 3 eixos cartesianos respectivos. Por essa razão considera-se um sensor de medida de inércia. No Anexo C, podem ser consultadas as características estáticas e dinâmicas do sensor *Xsens*® Mti.

Todas as medidas são realizadas em tempo real pelo software *MT Manager* da *Xsens*® e posteriormente analisadas em folha de cálculo no software *Microsoft*® *Office Excel*.

Na figura 4.16, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de  $15^\circ$ , frequência de 0,3 Hz e com razão cíclica a 50% para a entrada “Roll” da Cinemática Inversa da plataforma electromecânica.

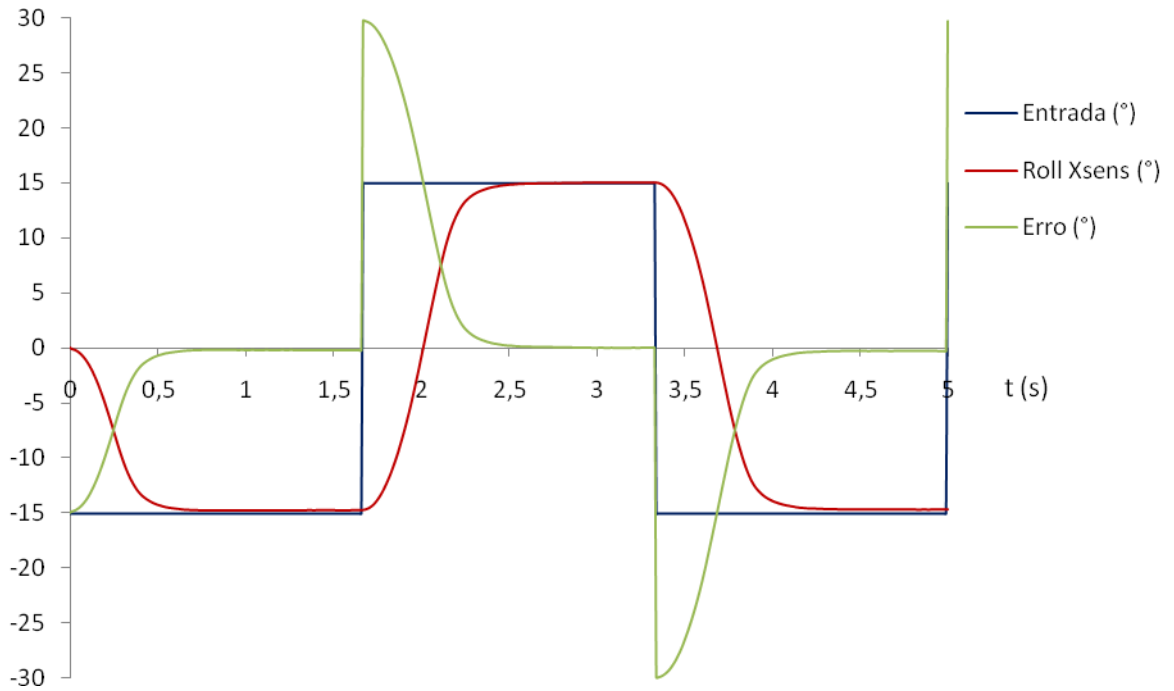


Figura 4.16 – Resposta temporal da plataforma electromecânica para um Roll do tipo quadrada

Na figura 4.17, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de  $15^\circ$  e frequência de 0,3 Hz para a entrada “Roll” da Cinemática Inversa da plataforma electromecânica.

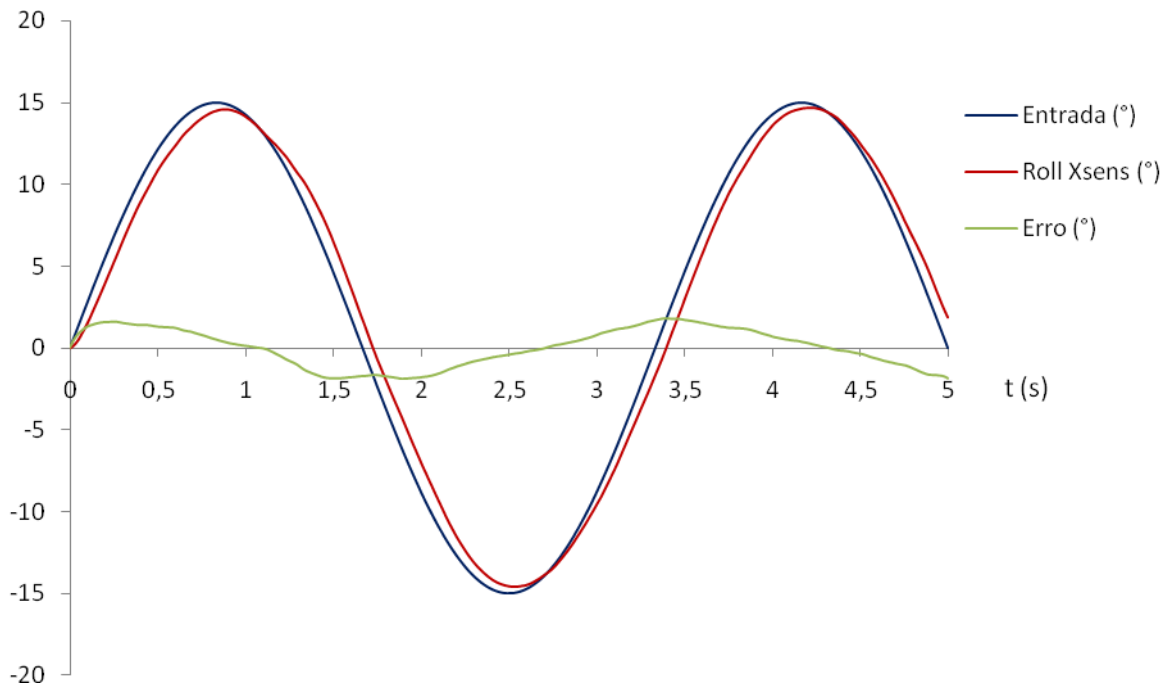


Figura 4.17 – Resposta temporal do Modelo Cessna-172 para um Roll do tipo sinusoidal

Na figura 4.18, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de  $15^\circ$ , frequência de 0,3 Hz e com razão cíclica a 50% para a entrada “Pitch” da Cinemática Inversa da plataforma electromecânica.

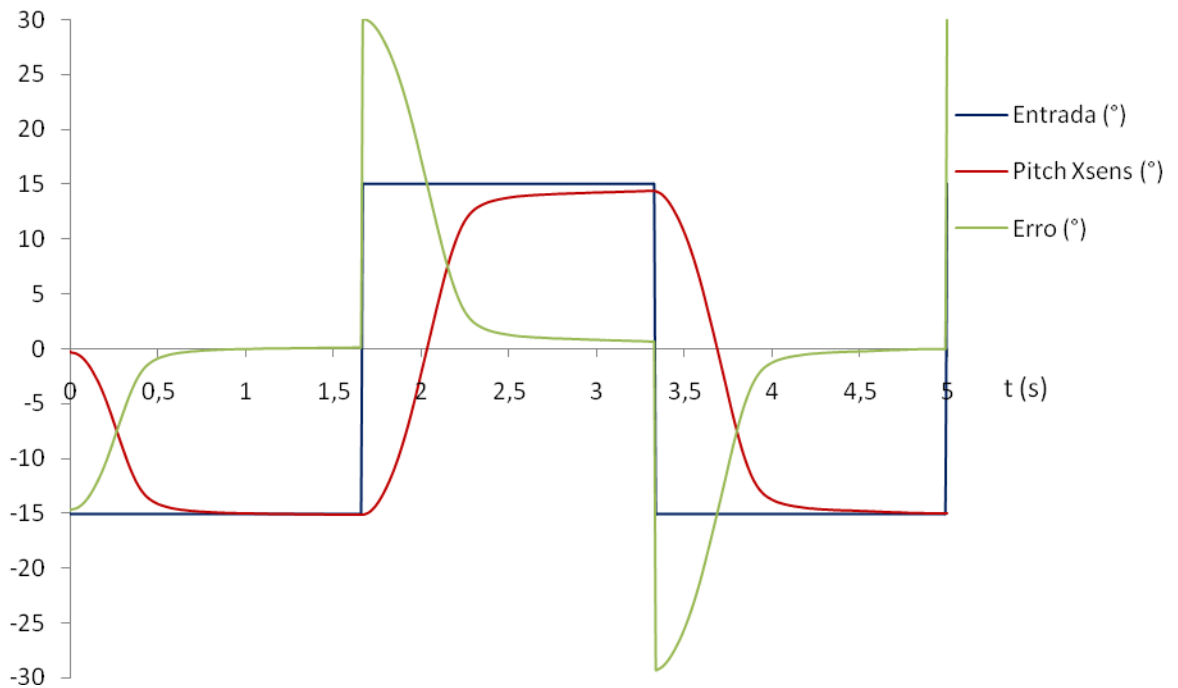


Figura 4.18 – Resposta temporal do Modelo Cessna-172 para um *Pitch* do tipo quadrada

Na figura 4.19, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de  $15^\circ$  e frequência de 0,3 Hz para a entrada “Pitch” da Cinemática Inversa da plataforma electromecânica.

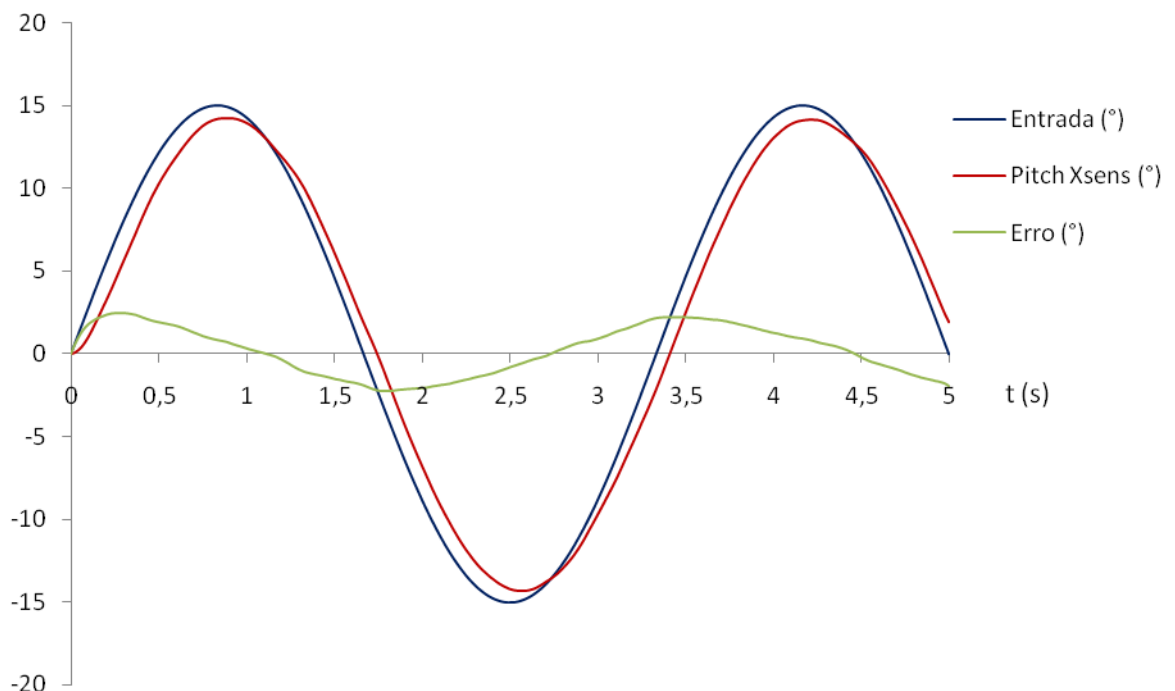


Figura 4.19 – Resposta temporal do Modelo Cessna-172 para um *Pitch* do tipo sinusoidal

Na figura 4.20, apresenta-se a resposta temporal para uma entrada do tipo quadrada com amplitude de  $15^\circ$ , frequência de 0,3 Hz e com razão cíclica a 50% para a entrada “Yaw” da Cinemática Inversa da plataforma electromecânica.

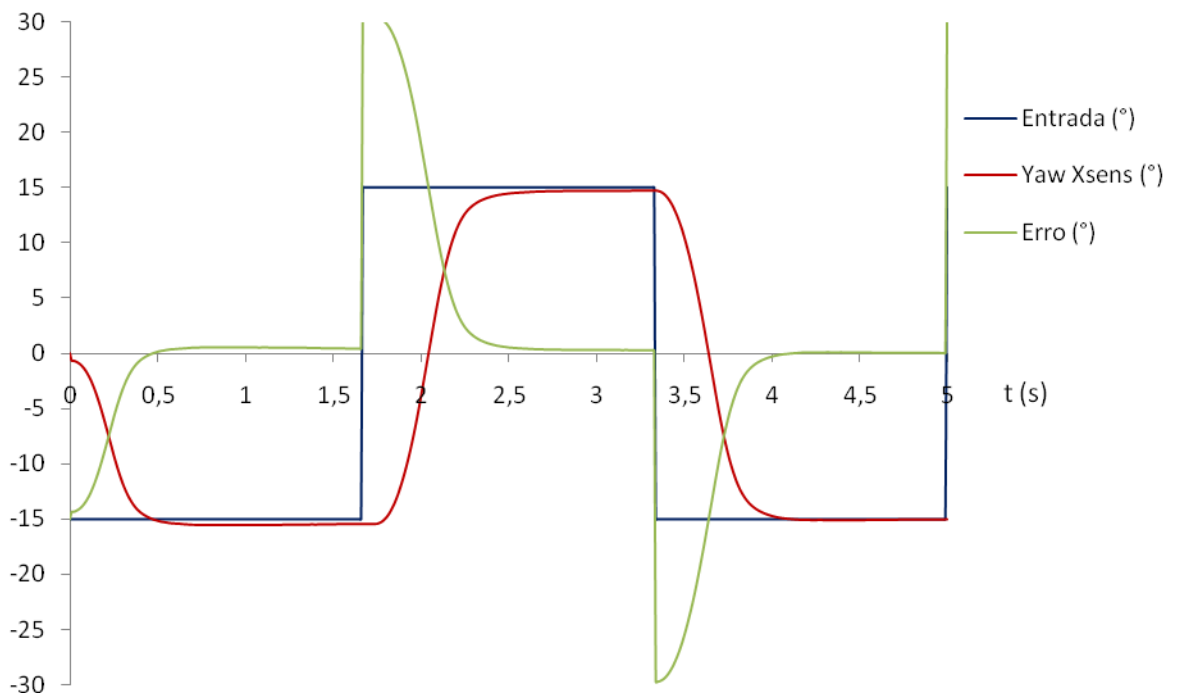


Figura 4.20 – Resposta temporal do Modelo Cessna-172 para um *Yaw* do tipo quadrada

Na figura 4.21, apresenta-se a resposta temporal para uma entrada do tipo sinusoidal com amplitude de  $15^\circ$  e frequência de 0,3 Hz para a entrada “Yaw” da Cinemática Inversa da plataforma electromecânica.

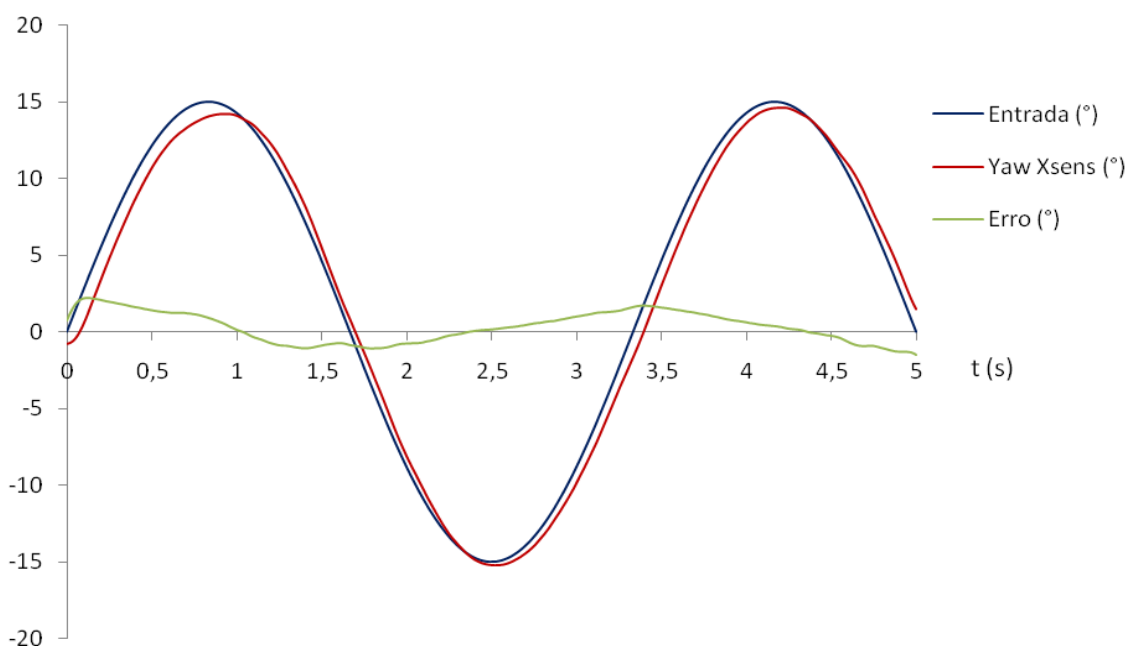


Figura 4.21 – Resposta temporal do Modelo Cessna-172 para um *Yaw* do tipo sinusoidal

As figuras 4.16, 4.18 e 4.20 mostram a evolução temporal para uma referência de posição angular do tipo quadrada para os movimentos de *Roll*, *Pitch* e *Yaw* respectivamente. Os tempos de estabelecimento a 2% da posição inicial de funcionamento da plataforma electromecânica para a referência angular inicial de  $-15^\circ$  é de 0,64s para o *Roll*, 0,66s para o *Pitch* e 0,43s para o *Yaw*. Após alcançar a referência, a plataforma apresenta um erro estático em valor absoluto de  $0,4^\circ (\pm 0,5^\circ)$  para o *Roll*,  $0,1^\circ (\pm 0,5^\circ)$  para o *Pitch* e  $0,5^\circ (\pm 0,5^\circ)$  para o *Yaw*, tendo um erro relativo de 2,0%, 0,7% e 3,4% respectivamente.

As figuras 4.17, 4.19 e 4.21 mostram a evolução temporal para uma referência de posição angular do tipo sinusoidal para os movimentos de *Roll*, *Pitch* e *Yaw* respectivamente. Em que os erros absolutos máximos verificados ao longo dos 5s são de  $2,0^\circ (\pm 0,5^\circ)$  para o *Roll*,  $2,5^\circ (\pm 0,5^\circ)$  para o *Pitch* e  $2,3^\circ (\pm 0,5^\circ)$  para o *Yaw*, tendo como erros relativos 13,4%, 16,7% e 15,4% respectivamente.

Tendo em conta o ambiente envolvente ao sensor *Xsens*<sup>®</sup> Mti, próprio da plataforma electromecânica, nomeadamente peças metálicas, máquinas eléctricas e conversores de potência, os resultados das medidas efectuadas podem apresentar erros de medição acrescidos, para além dos erros característicos do próprio sensor. Apesar de nas figuras 4.16 a 4.20 não estar representado o intervalo de incerteza característico do próprio sensor, na análise efectuada, este foi considerado e indicado entre parênteses.

## 4.5 – Sistema Global

Apesar da complexidade do sistema global do simulador Cessna-172, utilizou-se um único computador para processar todas as aplicações necessárias ao seu funcionamento. No entanto, para se obterem melhores resultados, poder-se-ia distribuir as várias aplicações por diversos computadores. Na figura 4.22, apresenta-se a monitorização do funcionamento do simulador Cessna-172.

Para se colocar o simulador Cessna-172 em condição de operacionalidade, é necessário executar o ficheiro auxiliar “runfg.bat”, programado com as opções próprias de arranque do *software FlightGear*<sup>®</sup>, o ficheiro “fg\_simulink.exe”, que lê a altura do solo em relação ao nível médio do mar, e o ficheiro “geral.mdl”, do *software MatLab/simulink*<sup>®</sup>. De seguida, configuram-se os endereços de IP e as Portas de comunicação dos diversos dispositivos. Concluindo estes procedimentos o sistema encontra-se em condições de arranque.

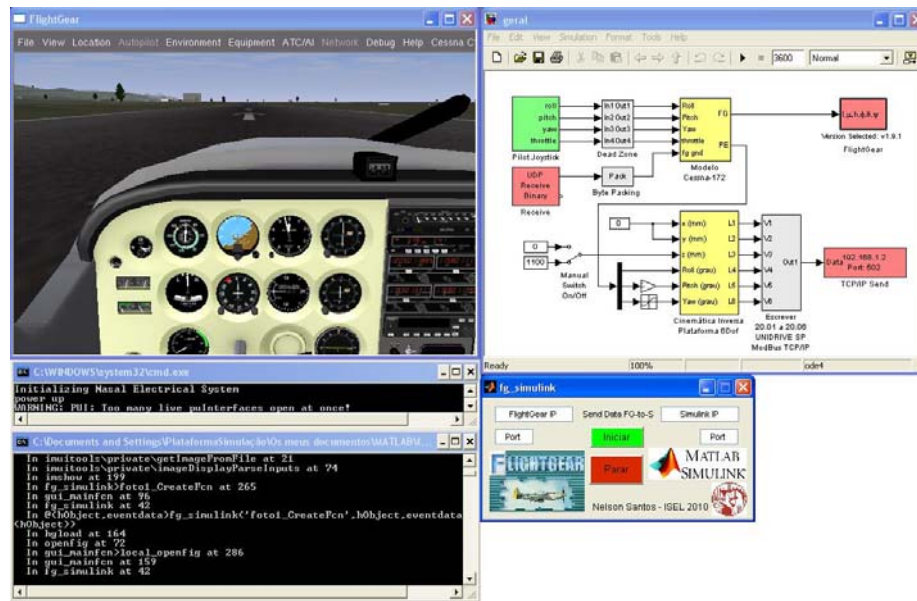


Figura 4.22 – Monitorização do funcionamento do simulador Cessna-172

Por fim, para colocar o simulador Cessna-172 em funcionamento, no programa “geral.mdl” do *MatLab/simulink*<sup>®</sup>, deve-se ajustar o *Switch* para a altura de 1,1m, com o intuito de passar a plataforma electromecânica do estado de repouso para o estado de funcionamento e, finalmente, colocar em modo *Run* o programa, ficando o simulador em operação no modo *Pseudo Real Time*.

Na figura 4.23, apresenta-se uma fotografia da plataforma electromecânica (a) e a imagem respectiva visualizada nos óculos de realidade virtual (b).

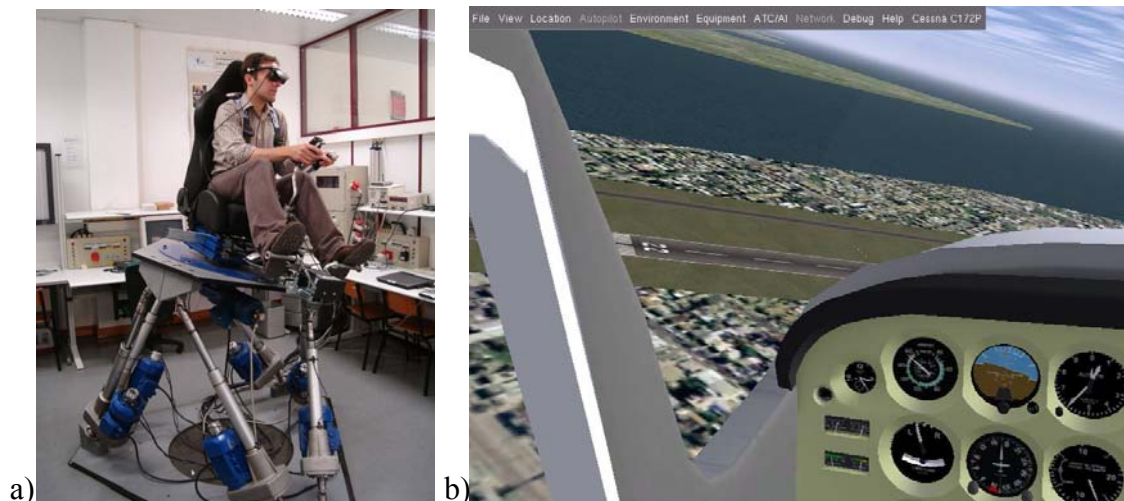


Figura 4.23 – Fotografia da plataforma electromecânica (a) e a imagem respectiva visualizada nos óculos de realidade virtual (b)

De salientar que a complexidade do sistema está para além das aplicações necessárias ao funcionamento do simulador, uma vez que no global este é composto pela plataforma electromecânica, *interfaces* humanas (*joystick* e óculos de realidade virtual), comunicação entre os diversos dispositivos e o *software* inerente ao seu funcionamento.

## 5 – Conclusões

### 5.1 – Considerações Finais

Neste trabalho aplicaram-se um modelo dinâmico de uma aeronave e sistemas de realidade virtual numa plataforma electromecânica de Gough-Stewart. Enquadraram-se estas temáticas em contextos históricos dos simuladores de voo e dos sistemas de realidade virtual.

Estudou-se a cinemática inversa da plataforma electromecânica, com o intuito de se desenvolver um modelo matemático para aplicação no *software MatLab/simulink*<sup>®</sup>, que permitisse a sua aplicação a outras plataformas de Gough-Stewart.

Tendo em conta o equipamento disponível, escolheu-se implementar na plataforma electromecânica o modelo dinâmico da aeronave Cessna-172, disponibilizado na livreria do *software MatLab/simulink*<sup>®</sup>, na *toolbox Aerosim*<sup>®</sup>, procedendo-se de seguida ao seu estudo, nomeadamente os modelos aerodinâmico, de propulsão, atmosférico, inercial, da Terra e das equações de movimento, com o intuito de aplicar o modelo dinâmico na cadeia de comando da plataforma electromecânica. Para o efeito, foram realizadas adaptações no modelo original, com vista a melhorar as sensações transmitidas ao utilizador do simulador, nomeadamente no modelo da Terra, criando-se uma variável de entrada para a altura do solo em relação ao nível médio do mar, no bloco do modelo Cessna-172 e desenvolveu-se ainda programação própria de leitura dessa variável no *software FlightGear*<sup>®</sup>, em tempo real, permitindo assim um maior realismo na simulação de voo, pelo que se concluiu que a plataforma electromecânica apresenta grande flexibilidade na implementação de modelos matemáticos e versatilidade na programação dos mesmos.

Configurou-se o *software FlightGear*<sup>®</sup> com o objectivo de receber os resultados do modelo dinâmico da aeronave e estabelecer os óculos de realidade virtual como *interface* humana de saída gráfica. Estudaram-se princípios de visualização e geração de imagens estereoscópicas, com o intuito de compreender o princípio de funcionamento dos óculos de realidade virtual, donde se concluiu que, se as imagens visualizadas não respeitarem o princípio da estereoscopia, poder-se-á correr o risco de transmitir ao utilizador sensações diferentes das inicialmente propostas para um simulador de voo. Apesar do *software FlightGear*<sup>®</sup> não produzir imagens em três dimensões, permitiu a exploração do ambiente gráfico e a obtenção de uma sensação satisfatória de condução de uma aeronave. Como

*interface* humana de entrada foi utilizado o *joystick* e o sensor de detecção da orientação da cabeça, incluído nos próprios óculos de realidade virtual, permitindo este último, adicionar maior interactividade, face a outros trabalhos desenvolvidos anteriormente na plataforma electromecânica.

Com base no modelo dinâmico da aeronave Cessna-172, realizaram-se simulações numéricas em *software MatLab/simulink*<sup>®</sup>, de forma a verificar o funcionamento correcto do modelo face aos *software* utilizados como referência, nomeadamente *FlightGear*<sup>®</sup> e *Microsoft*<sup>®</sup> *Flight Simulator*, satisfazendo a sua implementação na cadeia de comando da plataforma electromecânica, pelo que se concluiu que este demonstrou ser adequado e funcional na sua aplicação ao simulador voo.

A validação da implementação do modelo cinemático da plataforma electromecânica em *software MatLab/simulink*<sup>®</sup> foi efectuada recorrendo-se à análise da evolução temporal, através da comparação entre o sinal enviado para o bloco da cinemática inversa e o medido através do sensor inercial colocado na plataforma electromecânica. Apesar de ter sido desenvolvida programação com o intuito de permitir a leitura da orientação do sensor inercial em *software MatLab/simulink*<sup>®</sup>, este demonstrou ter uma grande latência face ao *software* próprio do sensor. A alternativa à análise de resultados neste âmbito foi realizada em *software Microsoft*<sup>®</sup> *Office Excel*, apesar do inconveniente de não poder garantir totalmente a sincronização entre os sinais analisados. No entanto, foi possível verificar que a plataforma electromecânica responde às solicitações impostas à entrada do bloco da cinemática inversa, apesar do sensor inercial utilizado apresentar um intervalo de incerteza nas medições de  $\pm 0,5^\circ$  e se considerar ser um sensor inadequado para aplicações em ambientes industriais, pois, pelas suas características, dificultou a obtenção dos resultados experimentais.

Estudou-se o protocolo de comunicação ModBus TCP/IP e desenvolveu-se programação própria para estabelecer a comunicação entre o *software MatLab/simulink*<sup>®</sup> e os conversores de potência, uma vez que não existia nenhuma aplicação na livreria deste que suportasse directamente a comunicação através da rede *Ethernet*, no protocolo ModBus TCP/IP. Apesar de não ser desenvolvida programação para a medição exacta do tempo de latência, concluiu-se que existiu um grande avanço neste campo, comparativamente a trabalhos desenvolvidos anteriormente na plataforma electromecânica, alcançando-se taxas de actualização na ordem de 2ms. É de salientar que foram testados outros tipos de comunicação, nomeadamente sistemas baseados em OPC Servers (Ole for Process Control), que têm como função a adaptação de protocolos, concluindo-se que estes

apresentaram taxas de actualização na ordem de 100ms, pelo que a sua única vantagem face ao implementado é a existência de blocos de comunicação próprios no *software MatLab/simulink*<sup>®</sup>.

Os resultados experimentais no simulador Cessna-172 demonstraram-se satisfatórios, apesar de, os óculos de realidade virtual apresentaram algumas limitações, nomeadamente na referência de zero, na orientação da cabeça do utilizador. Apesar desta limitação foi possível interagir com o ambiente virtual, confirmando-se que os sistemas visuais e interactivos nos simuladores de voo são fundamentais.

## 5.2 – Perspectivas de Desenvolvimento Futuro

Na sequência deste trabalho, como perspectivas de desenvolvimento futuro prevê-se a continuação de estudos de investigação e desenvolvimento experimental, de forma a introduzir melhorias no simulador e possibilitar aplicações em outras aéreas da engenharia.

No âmbito da plataforma electromecânica, prevê-se melhorar o seu comportamento dinâmico, diminuindo-se o peso das peças móveis e substituindo-se os actuadores electromecânicos por motores electromecânicos lineares de características superiores. A implementação de sensores inerciais de características superiores às utilizadas neste trabalho, possibilitará a implementação futura de sistemas de controlo de posição ou trajectória em cadeia fechada, processado em sistemas computacionais próprios de tempo real.

O sistema de realidade virtual poderá ser melhorado adquirindo óculos ou capacete para esse efeito com características superiores às utilizadas, visando a melhoria das características de imersão do utilizador. Por outro lado, e em alternativa, a construção de uma cabine fechada com implementação de instrumentação idêntica à da uma aeronave real seria outra perspectiva de desenvolvimento futuro a considerar. Ainda no âmbito da realidade virtual poderia considerar-se a melhoria do ambiente gráfico, através da utilização de *software* de simulação de voo de qualidade gráfica superior.

A implementação de outros modelos dinâmicos à plataforma, bem como *hardware* característico desses modelos permitiria uma maior versatilidade da plataforma, por exemplo, a inclusão de volante e pedais para a simulação de modelos matemáticos de veículos automóveis.

Por fim, considera-se que este projecto tem muitas aéreas com potencial de desenvolvimento, uma vez que, com a comunicação implementada para a plataforma

electromecânica através do *software MatLab/simulink*<sup>®</sup>, que é uma das ferramentas de cálculo mais utilizadas no meio académico, simplifica-se bastante a aplicação de novos modelos matemáticos à plataforma electromecânica.

## Referências Bibliográficas

- [1] – Ray L. Page, Brief History of Flight Simulation, Qantas Simulation Services. 1996
- [2] – Takashi Yoneyama, Jackson MatSuura, Aplicação dos Simuladores de Voo no Desenvolvimento e Avaliação de Aeronaves e Periféricos, ITA/IEEE, 1995
- [3] – Kevin Moore, Thales Training and Simulation Ltd  
Website: <http://homepage.ntlworld.com/bleep/SimHist3.html> (09/2010)
- [4] – Link Simulation & Trainer  
Website: <http://www.link.com/history.html> (09/2010)
- [5] – Royal Aeronautical Society, History of Flight Simulation  
Website: [http://www.raes-fsg.org.uk/20/Image\\_Gallery](http://www.raes-fsg.org.uk/20/Image_Gallery) (09/2010)
- [6] – QANTAS, Australia, Qantas Flight Training  
Website: <http://www.qantastraining.com/facilities.html> (09/2010)
- [7] – António Netto, Liliane Machado, Maria Oliveira, Realidade Virtual - Definições, Dispositivos e Aplicações, ISSN 1519-8219  
Website: <http://www3.di.ufpb.br/liliane/publi.html> (09/2010)
- [8] – Liliane Machado, Marcelo Zuffo, (PhD) Realidade Virtual no Modelamento e Simulação de Procedimentos invasivos em Oncologia Pediátrica, 2003  
Website: <http://www3.di.ufpb.br/liliane/publi.html> (09/2010)
- [9] – Virtual Kitchenball, University of Illinois  
Website: <https://segue.atlas.uiuc.edu/index.php?&action=site&site=aangell>  
(09/2010)
- [10] – Pimentel, K. & Teixeira, K. Virtual Reality – Thought the new looking glass, New York , McGraw-Hill 1995
- [11] – Sensics – panoramic head-mounted displays  
Website: <http://sensics.com/> (09/2010)
- [12] – Vuzux – View the future  
Website: <http://www.vuzix.com> (09/2010)

- [13] – The New York Times – Online Paper, Technology  
 Website: <http://www.nytimes.com/2009/03/09/technology/09topps.html> (09/2010)
- [14] – Ajay V. Sonar, Kyle D. Burdick, Ryan R. Begin, Eric M. Resch, Elizabeth M. Thompson, Eric Thacher, Janice Searleman, George Fulk and James J. Carroll, Development of a Virtual Reality-based Power Wheel Chair Simulator, IEEE 2005
- [15] – José Quadrado, Nelson Santos, Nuno Gancho, Emulação do Comportamento de um Helicóptero usando uma Plataforma Electromecânica com Sete Graus de Liberdade, ISEL 2007
- [16] – Kai Liu, John M. Fitzgerald and Frank L. Lewis, Kinematic Analysis of Stewart Platform Manipulator; IEEE 1993
- [17] – AeroSim – Aeronautical Simulation Blockset, User’s Guide 1.2  
 Website: <http://www.u-dynamics.com/aerosim> (09/2009)
- [18] – MathWorks, MatLab/simulink  
 Website: <http://www.mathworks.com> (09/2009)
- [19] – FlightGear, Website: <http://www.flightgear.org> (09/2009)
- [20] – Nicolas H. Durland, Defining Mean Sea Level in Military Simulations With DTED, Northrop Grumman Corporation
- [21] – Natural Resources, Canada, Canadian spatial Reference System  
 Website: [http://www.geod.nrcan.gc.ca/tools-outils/gpsh\\_e.php](http://www.geod.nrcan.gc.ca/tools-outils/gpsh_e.php) (09/2010)
- [22] – Alexander R. Perry, PAMurry, San Diego, The FlightGear Flight Simulator, CA
- [23] – Johann Schweikl, FlightGear Route Manager, MatLabCentral, 2009  
 Website: <http://www.mathworks.com/matlabcentral/fileexchange/25047> (09/2009)
- [24] – Tomasz Mazuryk and Michael Gervautz, Virtual Reality, History, Applications, Technology and Future, Vienna University of Technology, Austria
- [25] – Alexandre V. Maschio, A Estereoscopia: Investigação de Processos de Aquisição, Edição e Exibição de Imagens Estereoscópicas em Movimento, 2008  
 Website: <http://www.faac.unesp.br/posgraduacao/design/> (09/2010)

[26] – Emagin,

Website: <http://www.emagin.com/> (09/2010)

[27] – RCSimulation,

Website: <http://www.rcsimulations.com/> (09/2010)

[28] – Control Techniques, User Guide SM-Ethernet

Website: <http://www.controltechniques.com> (09/2010)

[29] – Control Techniques, User Guide Unidrive SP

Website: <http://www.controltechniques.com> (09/2010)

[30] – Introduction to ModBus TCP/IP, Technical Reference, Acromag Incorporated

[31] – Adelino Silva, Mário Felgueiras, Protocolo de Comunicação ModBus ISEP

[32] – The NASA GSFC and NIMA Join Geopotencial Model

Website: <http://cddis.nasa.gov/926/egm96/egm96.html> (09/2010)

[33] – Xsens Mti, Data sheet

Website: <http://www.xsens.com/en/general/mti> (09/2010)

## Anexos

### Anexo A – Listagem do Programa de Inicialização do *FlightGear*<sup>®</sup>

#### Ficheiro “runfg.bat”

C:

```
cd C:\Programas\FlightGear
```

```
SET FG_HOME=C:\Programas\FlightGear
```

```
SET FG_ROOT=C:\Programas\FlightGear\Data
```

```
SET FG_SCENERY=C:\Programas\FlightGear\Data\Scenery
```

```
SET FG_ROOT=C:\Programas\FlightGear\data
```

```
.\bin\win32\fgfs --aircraft=c172p --fdm=network,localhost,5501,5502,5503 --fog-fastest -
-disable-clouds --start-date-lat=2010:01:01:16:00:00 --disable-sound --on-ground --enable-
freeze --airport=LPOR --runway=04 --heading=32 --offset-distance=4.72 --offset-
azimuth=0 --lon=-9.0391 --lat=38.87195 --altitude=9 --props=5505
```

### Anexo B – Listagem do Programa de Comunicação entre *FlightGear*<sup>®</sup> e o *MatLab/Simulink*<sup>®</sup>

#### Ficheiro “fg\_simulink.exe”

```
function varargout = fg_simulink(varargin)
% FG_SIMULINK M-file for fg_simulink.fig
% FG_SIMULINK, by itself, creates a new FG_SIMULINK or raises the
existing
% singleton*.
%
% H = FG_SIMULINK returns the handle to a new FG_SIMULINK or the handle
to
% the existing singleton*.
%
% FG_SIMULINK('CALLBACK', hObject, eventData, handles,...) calls the local
% function named CALLBACK in FG_SIMULINK.M with the given input
arguments.
%
% FG_SIMULINK('Property','Value',...) creates a new FG_SIMULINK or raises
the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before fg_simulink_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to fg_simulink_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
```

```

% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help fg_simulink
% Last Modified by GUIDE v2.5 02-Aug-2010 16:12:59
% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @fg_simulink_OpeningFcn, ...
                  'gui_OutputFcn',  @fg_simulink_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before fg_simulink is made visible.
function fg_simulink_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to fg_simulink (see VARARGIN)

% Choose default command line output for fg_simulink
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes fg_simulink wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = fg_simulink_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function fgip_Callback(hObject, eventdata, handles)
% hObject    handle to fgip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fgip as text
%        str2double(get(hObject,'String')) returns contents of fgip as a
double

% --- Executes during object creation, after setting all properties.
function fgip_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to fgip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function fgport_Callback(hObject, eventdata, handles)
% hObject    handle to fgport (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of fgport as text
%         str2double(get(hObject,'String')) returns contents of fgport as
a double

% --- Executes during object creation, after setting all properties.
function fgport_CreateFcn(hObject, eventdata, handles)
% hObject    handle to fgport (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sip_Callback(hObject, eventdata, handles)
% hObject    handle to sip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sip as text
%         str2double(get(hObject,'String')) returns contents of sip as a
double

% --- Executes during object creation, after setting all properties.
function sip_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sip (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function sport_Callback(hObject, eventdata, handles)
% hObject    handle to sport (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of sport as text
%        str2double(get(hObject,'String')) returns contents of sport as a
double
% --- Executes during object creation, after setting all properties.
function sport_CreateFcn(hObject, eventdata, handles)
% hObject      handle to sport (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject      handle to start (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

global on
global gnd_elev_m
global pitch_deg
global roll_deg
global off

off=0;
on=1;

flightgearIP = get(handles.fgip,'String');
flightgearPORT = str2num(get(handles.fgport,'String'));

simulinkIP = get(handles.sip,'String');
simulinkPORT = str2num(get(handles.sport,'String'));

%iniciar porta do FlightGear
import java.net.Socket
import java.io.*

fg_socket = Socket(flightgearIP, flightgearPORT);
out = PrintWriter(fg_socket.getOutputStream(), true);
in = BufferedReader(InputStreamReader(fg_socket.getInputStream()));

%iniciar porta simulink
udpsimulink = udp(simulinkIP,simulinkPORT);
fopen(udpsimulink);

while on==1

%ler altura ao solo
    out.println(sprintf('get /position/ground-elev-m\r\n'));
    gnd_elev_m_str = char(in.readLine());
    j = findstr(gnd_elev_m_str, '');
    gnd_elev_m = str2double(gnd_elev_m_str(j(1)+1:j(2)-1));

% ler Pitch

```

```

out.println(sprintf('get /orientation/pitch-deg\r\n'));
pitch_deg_str = char(in.readLine());
j = findstr(pitch_deg_str, '');
pitch_deg = str2double(pitch_deg_str(j(1)+1:j(2)-1));

% ler Roll
out.println(sprintf('get /orientation/roll-deg\r\n'));
roll_deg_str = char(in.readLine());
j = findstr(roll_deg_str, '');
roll_deg = str2double(roll_deg_str(j(1)+1:j(2)-1));

%converter para array uint8 para enviar
gndem = typecast(gnd_elev_m, 'uint8');
pitch = typecast(pitch_deg, 'uint8');
roll = typecast(roll_deg, 'uint8');

%enviar para o simulink
fwrite(udpsimulink, [gndem pitch roll]);

pause(0.2);

if off==1
    break
end
end

%fechar porta simulink
fclose(udpsimulink)
delete(udpsimulink)

%fechar porta fg
out.close();
in.close();
fg_socket.close();

% --- Executes on button press in stop.
function stop_Callback(hObject, eventdata, handles)
% hObject    handle to stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global off
off=1;

%FOTOS
function foto1_CreateFcn(hObject, eventdata, handles)
axes (hObject);
imshow ('fg1.jpg');

function foto2_CreateFcn(hObject, eventdata, handles)
axes (hObject);
imshow ('matlab.jpg');

function foto3_CreateFcn(hObject, eventdata, handles)
axes (hObject);
imshow ('isel.png');

```

## Anexo C – Características do Sensor Inercial XSENS MTI

### MTI TECHNICAL SPECIFICATIONS

#### Attitude and Heading

Static accuracy (roll/pitch)	<0.5 deg
Static accuracy (heading) <sup>1</sup>	<1 deg
Dynamic accuracy <sup>2</sup>	2 deg RMS
Angular resolution <sup>3</sup>	0.05 deg
Dynamic range:	
- Pitch	± 90 deg
- Roll/Heading	± 180 deg
Maximum update rate:	
- Onboard processing	256 Hz
- External processing	512 Hz

Specified performance operating range <sup>4</sup>	0...+55 °C
--	------------

#### Interfacing

Digital interface	RS-232, RS-485, RS-422 (max 921k6 bps) and USB (ext. converter)
Operating voltage	4,5 - 30V
Power consumption	350 mW
Interface options I/O	SyncOut, AnalogIn, SyncIn (depends on digital interface)

#### Maximum operational limits

Ambient temperature operating range <sup>4</sup>	-40...+85 °C
--	--------------

### INDIVIDUAL SENSOR SPECIFICATIONS

#### Sensor performance

Dimensions	
Full Scale (standard)	
Linearity	
Bias stability <sup>6</sup>	
Scale Factor stability <sup>5</sup>	
Noise	
Alignment error	
Bandwidth	
Max update rate	

#### Rate of turn

3 axes	
± 300 deg/s	
0.1% of FS	
1 deg/s	
-	
0.05 deg/s/√Hz	
0.1 deg	
40 Hz	
512 Hz	

#### Acceleration

3 axes	
± 50 m/s <sup>2</sup>	
0.2% of FS	
0.02 m/s <sup>2</sup>	
0.03%	
0.002 m/s <sup>2</sup> /√Hz	
0.1 deg	
30 Hz	
512 Hz	

#### Magnetic field

3 axes	
± 750 mGauss	
0.2% of FS	
0.1 mGauss	
0.5%	
0.5 mGauss	
0.1 deg	
10 Hz	
512 Hz	

## ERRATA

**Páginas 15, 20, 23, 34, 53, 57, 58, 71 e 72, nos respectivos Parágrafos 2, 6, 2, 3, 3, 4, 3, 3 e 4, nas respectivas Linhas 6, 10, 4, 2, 4, 3-6, 2, 2 e 3**

Onde se lê “livraria” deve ler-se “biblioteca”.

**Página 21, Tabela 3.2**

Onde se lê “[ $W_N W_E W_D$ ] (rad/s)” deve ler-se “[ $W_N W_E W_D$ ] (m/s)”.

**Página 22, Tabela 3.2**

Onde se lê “rates (rad/m)” deve ler-se “rates (rad/s)”.

**Página 42, Parágrafo 2, Linha 3**

Onde se lê “e arranque” deve-se ler-se “de arranque”.

**Página 51, Parágrafo 2, Linha 3**

Onde se lê “na *performance*” deve ler-se “no desempenho”.

**Página 58 e 59, Tabela 4.5, 4.6 e 4.7**

Deve colocar-se o tamanho de letra para 12

**Página 66, 67 e 68, Figura 4.17, 4.18, 4.19, 4.20 e 4.21**

Onde se lê “do Modelo Cessna-172” deve ler-se “da plataforma electromecânica”.