

Escola Politécnica da USP

Trabalho de Formatura **Engenharia Mecânica**

Dinâmica de Vôo, Desempenho da Aeronave e **Simulador Tridimensional**

Formando: Mauricio Medaets

Orientador: Prof. Agenor de Toledo Fleury

FICHA CATALOGRÁFICA

Medaets, Mauricio

Dinâmica de vôo, desempenho da aeronave e simulador tridimensional: dedução, resolução de regime permanente e integração das equações tridimensionais de vôo / M. Medaets. – São Paulo, 2009.

159 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

**1. Aeronaves 2. Vôo (Engenharia de aeronaves)
3. Simulação 4. Dinâmica - I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II. t.**

Conteúdo

1. Introdução	11
2. Abordagem Bidimensional	13
2.1. Equações do movimento	13
2.2. Modelos Aerodinâmicos	15
2.3. Equilíbrio de Momentos	17
2.4. Resolução do sistema.....	19
2.5. Exemplo.....	21
2.6. Para reflexão do leitor:.....	24
3. Corrida em pista	25
4. Equacionamento Tridimensional.....	29
4.1. Discussão inicial.....	29
4.2. TMB.....	31
4.3. TMA.....	34
4.4. Esforços Aerodinâmicos	40
4.5. Convenções	43
4.5.1. Índices Subscritos.....	43
4.5.2. Forças e Momentos de Superfícies.....	43
4.5.3. Forças e Momentos de Propulsão	44
4.5.4. Sinais	44
4.6. Sistema Completo de Esforços.....	45
5. Análise e Simplificações dos Esforços.....	47
5.2. Asa	48
5.3. Profundor	49
5.4. Fuselagem.....	50
5.5. Leme	51
5.6. Propulsão.....	52
5.7. Resumo	54
6. Equilíbrio Tridimensional de Momentos	55
7. Análise de Sensibilidade para o Raio Instantâneo de Curva.....	59
8. Resolução Semi-Analítica.....	63
9. Aplicação do Método	67
9.1. Velocidade	67
9.2. Ângulo de Hélice	68
9.3. Raio de Curva	69
9.4. Atuação dos Comandos	71
10. Considerações sobre a Primeira Parte	73

11. Segunda Etapa	75
12. Bases Numéricas.....	77
12.1. As cinco rotações finitas e seu equivalente numérico.....	77
12.2. Vetor Rotação.....	80
12.3. Derivação dos Versores.....	81
12.4. Resgate das Rotações Finitas.....	82
13. TMA Completo	83
13.1. Nova Base.....	83
13.2. Prova Real.....	86
14. O Programa.....	87
14.1. Estrutura	88
14.2. Funções.....	89
14.2.1. Main.....	89
14.2.2. Parâmetros e Constantes.....	90
14.2.3. Regimes de Voo.....	91
14.2.4. Estabilidade Não-Linear.....	92
14.3. Problemas.....	93
14.3.1. Ruído Numérico.....	94
15. Gram-Schmidt e Runge-Kutta	99
16. Gimble-lock.....	101
17. Aprimoramentos do Modelo	103
17.1. Auto-trim.....	103
17.2. Stall da Asa	103
17.3. Dutch-Roll	104
17.4. Momentos da Propulsão	106
18. Real-time e Joystick	107
18.1. Implementação	107
18.2. Exemplos.....	107
19. Conclusão	109
20. Bibliografia	111
ANEXO I - Desenvolvimento em Maple.....	113
ANEXO II - Planta.....	121
ANEXO III - Código em MATLAB.....	123
ANEXO IV - CD.....	159

Relação de Figuras

Figura 2.1 - Diagrama 2D de Forças.....	13
Figura 2.2 - Distribuição de Pressão	16
Figura 2.3 - Relação de CL: asa/profundor	22
Figura 2.4 - Comportamento geral dos coeficientes	22
Figura 2.5 - Mapa de vôo 2D	23
Figura 3.1 - Corrida em pista	25
Figura 4.1 - 6 graus de liberdade	29
Figura 4.2 - Vôo helicoidal	30
Figura 4.3 - Vista superior.....	33
Figura 4.4 - Derrapagem	35
Figura 4.5 - Ângulos de trajetória.....	38
Figura 4.6 - Superfícies de comando	39
Figura 4.7 - Perfil de velocidades induzido por curva	40
Figura 4.8 - Linhas de sustentação	41
Figura 5.1 - Coeficientes: CL, CD e CM.....	47
Figura 5.2 - Ailerons	49
Figura 5.3 - Resumo: Trajetória helicoidal.....	54
Figura 6.2 - Vista superior: Ailerons e Profundor	56
Figura 6.3 - Vista lateral: Leme.....	56
Figura 7.2 - Erro da aproximação.....	61
Figura 8.2 - Comandos.....	63
Figura 9.1 - Velocidade.....	67
Figura 9.2 - Ângulo de hélice.....	68
Figura 9.3 - Raio de curva	69
Figura 9.4 - Erro cometido.....	70
Figura 9.5 - Atuação do profundor	71
Figura 9.6 - Atuação do Leme.....	72
Figura 9.7 - Atuação dos ailerons.....	72
Figura 14.1 - Hierarquia das funções	88
Figura 14.2 - Main	89
Figura 14.3 - Parâmetros e Constantes.....	90
Figura 14.4 - Regimes de Vôo.....	91
Figura 14.5 - Estabilidade e Controle	92
Figura 14.6 - Ruído numérico	94
Figura 14.7 - Divergência ode113 – 100s.....	95
Figura 14.8 - Resultado 3D – 100s.....	95
Figura 14.9 - Divergência ode113 – 200s.....	96
Figura 14.10 - Resultado 3D – 200s.....	96
Figura 14.11 - Divergência ode45 – 200s.....	97
Figura 14.12 - Resultado 3D – 200s.....	97
Figura 15.1 - Estabilidade numérica.....	100
Figura 18.1 - Simulador	108

1. Introdução

Este trabalho visa o desenvolvimento e a simulação de um modelo de voo de aeronaves, enfatizando a dedução completa e não-linear das equações do movimento, a discussão dos modelos aerodinâmicos e a proposição de um método de resolução para o regime permanente e integração numérica das equações.

Partindo de uma abordagem bidimensional do regime de voo, irá se sugerir uma expansão tridimensional, com sistemas de coordenadas convenientemente escolhidos, em função dos quais estudos de manobras, trajetórias, desempenho e estabilidade serão realizados.

Uma metodologia de projeto de aeronave irá se propor ao longo do trabalho, em paralelo com a modelagem matemática extensa, de modo a se estabelecer uma sólida ferramenta de análise de desempenho em voo, que quantifique o comportamento dinâmico de forma mais aprofundada que a atualmente presente na literatura de graduação.

A bibliografia normalmente utilizada em cursos de graduação traz grande enfoque na obtenção de coeficientes aerodinâmicos mas padece de um excesso de simplicidade na resolução das equações dinâmicas.

Regimes de voo compensado devem servir como alicerce para análises de estabilidade e desempenho e, portanto, serão enfoque dos capítulos iniciais trabalho. Uma referência essencial para tais estudos dinâmicos é a boa definição da condição de equilíbrio em torno da qual se pretende operar.

Uma vez que a modelagem aerodinâmicas das forças e momentos sobre a aeronave ocorre em espaços de potências inteiras da velocidade, sistemas polinomiais podem ser analiticamente obtidos e, com algumas restrições discutidas a frente, resolvidos. Uma relação biunívoca entre o conjunto de comandos e a condição final de regime será buscada na resolução da mecânica de voo.

O histórico do aluno em projetos acadêmicos relacionados a construção e projeto de aeronaves radio controladas, como por exemplo a competição Aerodesign organizada pela SAE, permitirá a utilização de dados de aeromodelos reais para a experimentação e refinamento do modelo.

As aeronaves de maior interesse comercial são aeronaves comumente com aft tail e propulsão no nariz ou acoplada às asas. Obviamente, as características aerodinâmicas de

qualquer variação de configuração são específicas devido a efeitos de interferência. Por outro lado, as equações dinâmicas do movimento aeronave são fortemente similares.

Dessa forma, a formulação mecânica é genérica e parte do pressuposto de que os principais coeficientes aerodinâmicos são conhecidos. Este trabalho não dará enfoque a métodos de obtenção de coeficientes, como simulações do escoamento, teoria de escoamento potencial, experimentos em túnel de vento, mas se dedicará a aplicação de tais dados e discutirá suas implicações na dinâmica de voo, evitando ao máximo aproximações. Posteriormente será avaliada a necessidade da introdução de acoplamentos fluído-mecânicos.

Ao final do trabalho, o simulador resultante poderá ser programado com dados aerodinâmicos de qualquer aeronave que possua configuração similar ao padrão utilizado e fornecerá comportamentos dinâmicos que serão confrontados com a realidade.

2. Abordagem Bidimensional

2.1. Equações do movimento

Como ponto de partida toma-se a análise plana do problema, num sistema cartesiano inercial com velocidade relativa ao solo nula. De tal forma, o diagrama de corpo livre em uma aeronave de dois fólios segue ilustrado abaixo. As coordenadas dos centros aerodinâmicos de das superfícies sustentadoras em relação ao centro de massa da aeronave é dado de maneira polar por γ_i e l_i .

Pela convenção adotada, “w” subscrito faz alusão à asa, “ht” à Horizontal Tail ou profundor. O ângulo θ indica o arco entre a horizontal e o eixo da aeronave que possui simultaneamente a resultante da propulsão e o centro de massa. Logo, este modelo requer a pertinência do CG à linha de ação da força tratora.

O ângulo ϕ indica a direção aparente do escoamento induzida pela movimentação do CG e pelo vento. As velocidades induzidas pela rotação de corpo rígido θ_{ponto} são por hora desconsideradas e irão se mostrar desnecessárias para análise realizada nesta seção.

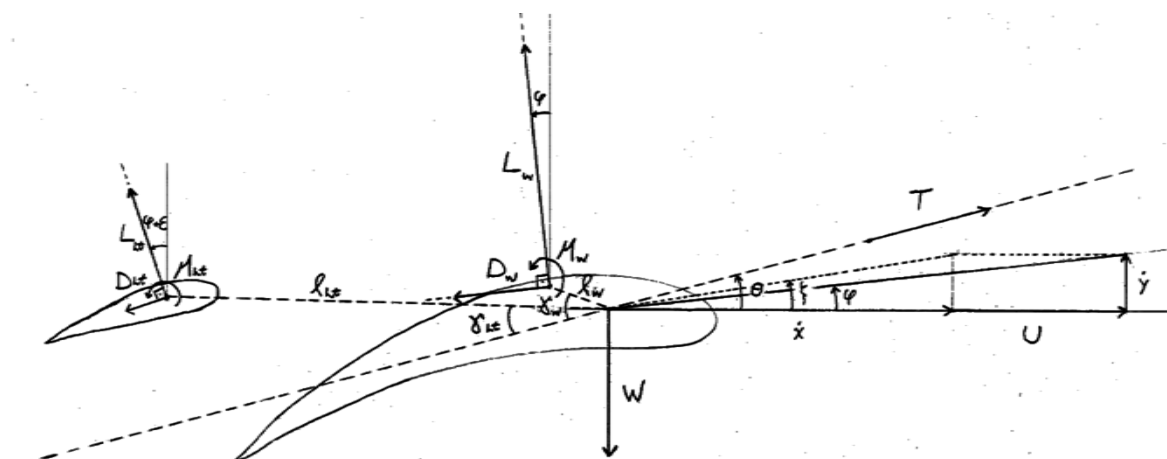


Figura 2.1 - Diagrama 2D de Forças

Aplicando os teoremas de movimento do centro de massa e do momento angular, obtêm-se três equações diferenciais.

$$\begin{aligned}
(2.1) \quad m\ddot{x} &= T \cos \theta - D_w \cos \varphi - D_{ht} \cos(\varepsilon + \varphi) - L_w \sin \varphi - L_{ht} \sin(\varepsilon + \varphi) \\
m\ddot{y} &= T \sin \theta + L_w \cos \varphi + L_{ht} \cos(\varepsilon + \varphi) - D_w \sin \varphi - D_{ht} \sin(\varepsilon + \varphi) - W \\
I\ddot{\theta} &= D_w l_w \sin(\gamma_w - \theta + \varphi) - L_w l_w \cos(\gamma_w - \theta + \varphi) + M_w + \\
&\quad + D_{ht} l_{ht} \sin(\gamma_{ht} - \theta + \varphi + \varepsilon) - L_{ht} l_{ht} \cos(\gamma_{ht} - \theta + \varphi + \varepsilon) + M_{ht}
\end{aligned}$$

Para trajetórias próximas a horizontal, a expansão das funções trigonométricas em series de Euler e o negligenciamento de termos de segunda ordem e superior introduz erros de no máximo 3,6% na função cosseno e 1,2% na função seno desde que os ângulos não excedam o limite de 15 graus. Dessa forma, podem-se tornar as equações lineares na variável independente φ ao introduzirem-se as aproximações abaixo.

$$(2.2) \quad \begin{cases} \tan \varphi = \frac{\dot{y}}{\dot{x} + U} \\ V = \sqrt{(\dot{x} + U)^2 + \dot{y}^2} = \frac{\dot{x} + U}{\cos \varphi} \\ \tan \xi = \frac{\dot{y}}{\dot{x}} = \frac{V \sin \varphi}{V \cos \varphi - U} \end{cases} \xRightarrow{\varphi \rightarrow 0} \begin{cases} V \cong \dot{x} + U \\ \varphi \cong \frac{\dot{y}}{V} \\ \xi \cong \frac{\varphi V}{V - U} \end{cases} \xRightarrow{\frac{d}{dt}} \begin{cases} \ddot{x} \cong \dot{V} - \varphi \dot{\varphi} V \\ \ddot{y} \cong \dot{\varphi} V + \varphi \dot{V} \end{cases}$$

Note-se que ao alinhar a resultante da propulsão com o centro de massa, a equação resultante do TMA apresenta todos seus termos proporcionais ao quadrado da velocidade. Essa característica implica em restringir-se ao caso em que o empuxo gerado pela propulsão não gera momentos sobre a aeronave. Tal condição permite a resolução analítica das equações. Caso o desalinhamento da linha de ação do empuxo com o centro de massa seja desejado, um método alternativo deverá ser aplicado como discutido no capítulo 17.

A consideração da fuselagem deve acarretar na inclusão de novos termos nesta formulação, mas não irá alterar a natureza das equações do movimento. O capítulo 5 tratará desse caso.

Uma vez que o ponto de partida de toda análise é o conjunto de regimes de voo compensado e permanente, buscam-se as soluções do sistema de equações para acelerações nulas, ou seja o conjunto de soluções de um sistema algébrico.

O conjunto de soluções, dada a natureza física do fenômeno, deve associar tríades de velocidade, atitude e taxa de subida ao conjunto de condições de voo. É esperado que escoamentos mais “estressados” na asa, ou seja para maiores coeficientes de sustentação estejam associados a velocidades menores e taxas de subida altas, enquanto menores coeficientes de sustentação levem a velocidades mais altas e taxas de subida inclusive

negativas. Logo, uma parametrização do espaço de soluções sugere-se em função da própria sustentação da asa, ou mais especificamente, do ângulo efetivo de ataque da mesma.

É ainda necessário que se modelem as forças e momentos aerodinâmicos. A discussão a seguir deve ajudar a esclarecer abordagem e fornecer uma base teórica para o leitor não muito familiarizado aos modelos clássicos de esforços aerodinâmicos em asas. Tais grandezas são obtidas pela utilização de coeficientes adimensionais largamente estudados na literatura.

2.2. Modelos Aerodinâmicos

Para um pólo qualquer a distribuição de pressões gera sobre a superfície uma força resultante (decomposta por definição na direção paralela e perpendicular ao escoamento) e um momento. É possível escolher um pólo que garanta a variação mínima do momento resultante com o ângulo de ataque. Esse pólo, chamado de Centro Aerodinâmico, é uma definição altamente conveniente pois permite, na maioria dos casos, que o momento seja considerado constante e apenas as forças de sustentação e arrasto dependam do ângulo de incidência do escoamento. Em geral, pode não existir um pólo em relação ao qual o momento não varie. No entanto, o pólo de variação mínima permite em boa aproximação [1] a linearização do coeficiente de momento com derivadas muito próximas de zero.

Faz-se a hipótese de que a transição do regime laminar para o turbulento não deve ocorrer na faixa de velocidades em que a aeronave opera. Dessa forma, os coeficientes passam a depender em boa aproximação apenas do ângulo efetivo de ataque quando adimensionalizados pela pressão dinâmica, $\frac{1}{2}\rho V^2$, a área da asa, e corda média no caso do momento. Caso contrário, o modelo deve ser aplicado por trechos.

Para o cálculo dos coeficientes adimensionais, a teoria de linhas de sustentação formulada por Prandtl no início do século XX permite a conversão de dados de fólios bidimensionais em coeficientes referentes a asas reais com envergadura finita, afinamento e torção.

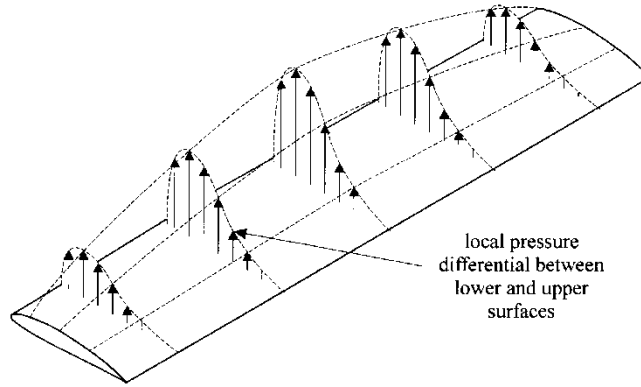


Figura 2.2 - Distribuição de Pressão

Essa ferramenta gera um sistema linear cuja ordem é tão maior quanto for a precisão desejada, e permanece como o principal método utilizado na tridimensionalização de fólhos em asa [3]. Por outro lado, o método possui limitações para a geometria da asa e efeitos de interferência. Para esses casos, existem generalizações da teoria [8] que não serão abordadas pois levam a refinamentos de grau superior ao momentaneamente considerado.

A implicação prática mais relevante na modelagem por linhas de sustentação é o acoplamento da sustentação com o arrasto da superfície em questão. A teoria demonstra que o “custo” de um acréscimo de sustentação por meio de um aumento do ângulo de ataque é uma variação parabólica do arrasto gerado. O coeficiente de arrasto passa a ser definido por um termo constante (arrasto viscosos e de pressão) e outro proporcional ao quadrado do coeficiente de sustentação (arrasto induzido por sustentação).

A asa é vista tão mais eficiente quanto menor a variação do arrasto induzido com a sustentação. Conceitualmente, a asa gerará tão menos arrasto induzido quanto mais alongada for a mesma, ou seja, sua envergadura superar sua corda em mais ordens de grandeza possível. O parâmetro adimensional, $\lambda = b^2/S$, Aspect Ratio quantifica essa esbelteza.

Há um limite máximo demonstrado pela teoria chamado de eficiência elíptica. Para esse caso ideal o coeficiente de arrasto apresenta a seguinte forma, $C_D = C_{D0} + \frac{C_L^2}{\pi\lambda}$. O distanciamento desse caso introduz um fator de correção chamado de coeficiente de Oswald, $e \in]0,1] \subset \mathbb{R}$, no denominador do termo de arrasto induzido que implica em um aumento do mesmo.

Seguindo a teoria brevemente discutida acima, o modelo aerodinâmico de forças de momentos para a asa e o profundor adquire a seguinte forma:

$$(2.3) \quad \begin{cases} L_w = \frac{1}{2} \rho V^2 S_w C_{L_w} \\ D_w = \frac{1}{2} \rho V^2 S_w \left(C_{D0_w} + \frac{C_{L_w}^2}{\pi \lambda_w e_w} \right) \\ M_w = \frac{1}{2} \rho V^2 S_w C_{M_w} c_w \end{cases} \quad \begin{cases} L_{ht} = \frac{1}{2} \rho V^2 S_{ht} C_{L_{ht}} \\ D_{ht} = \frac{1}{2} \rho V^2 S_{ht} \left(C_{D0_{ht}} + \frac{C_{L_{ht}}^2}{\pi \lambda_{ht} e_{ht}} \right) \\ M_{ht} = \frac{1}{2} \rho V^2 S_{ht} C_{M_{ht}} c_{ht} \end{cases}$$

Uma vez que o modelo aerodinâmico obriga a adoção de termos em segunda potência da velocidade, pode-se assumir que o empuxo é representado por uma função de segundo grau da velocidade normal incidente no plano da hélice ou turbina, sem introdução de empecilhos algébricos algum. Uma discussão mais aprofundada da variação da força de empuxo com a velocidade para propulsões a hélice ocorrerá na segunda metade do trabalho.

$$(2.4) \quad T(v) = a_2 v^2 + a_1 v + a_0; \quad v = V \cos(\theta - \varphi)$$

2.3. Equilíbrio de Momentos

Em regime permanente, para um dado ângulo efetivo de ataque, a diferença entre θ e φ deve permanecer constante e é dada pela própria definição do C_L :

$$(2.5) \quad C_{L_w} = C_{L_{\alpha_w}} (\theta - \varphi - \alpha_{L0_w}) \Rightarrow \alpha = \theta - \varphi = \frac{C_{L_w}}{C_{L_{\alpha_w}}} + \alpha_{L0_w}$$

Assim deixa-se α em função apenas do C_L buscado, de coeficientes aerodinâmicos e parâmetros geométricos que permanecem constantes.

Utilizando essa relação, pode-se realizar uma substituição na equação dos momentos e deixá-la independente de θ e φ .

$$(2.6) \quad \frac{I \ddot{\theta}}{1/2 \rho V^2} = S_w l_w \left(C_{D0_w} + \frac{C_{L_w}^2}{\pi \lambda_w e_w} \right) \sin \left(\gamma_w - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} \right) - S_w l_w C_{L_w} \cos \left(\gamma_w - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} \right) + S_w C_{M_w} c_w +$$

$$S_{ht} l_{ht} \left(C_{D0_{ht}} + \frac{C_{L_{ht}}^2}{\pi \lambda_{ht} e_{ht}} \right) \sin \left(\gamma_{ht} - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} + \varepsilon \right) - S_{ht} l_{ht} C_{L_{ht}} \cos \left(\gamma_{ht} - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} + \varepsilon \right) + S_{ht} C_{M_{ht}} c_{ht}$$

Impondo o equilíbrio, ou seja $\theta_{2\text{pontos}}$ nulo, uma relação de momentos é obtida:

$$\begin{aligned}
(2.7) \quad 0 = & C_{L_{ht}}^2 \left[\frac{S_{ht} I_{ht}}{\pi \lambda_{ht} e_{ht}} \sin \left(\gamma_{ht} - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} + \varepsilon \right) \right] + C_{L_{ht}} \left[-S_{ht} I_{ht} \cos \left(\gamma_{ht} - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} + \varepsilon \right) \right] + \\
& + \left[S_w I_w \left(C_{D0_w} + \frac{C_{L_w}^2}{\pi \lambda_w e_w} \right) \sin \left(\gamma_w - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} \right) - S_w I_w C_{L_w} \cos \left(\gamma_w - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} \right) + \right. \\
& \left. + S_{ht} I_{ht} C_{D0_{ht}} \sin \left(\gamma_{ht} - \frac{C_{L_w}}{C_{L_{\alpha_w}}} - \alpha_{L0_w} + \varepsilon \right) + S_w C_{M_w} c_w + S_{ht} C_{M_{ht}} c_{ht} \right] \Rightarrow C_{L_{ht}} = f(C_{L_w})
\end{aligned}$$

Tal relação implica em uma função de segundo grau em CLht como parâmetro livre, resultando em CLw que deve ser atingido no equilíbrio. De tal forma, surge a função $C_{L_{ht}} = f(C_{L_w})$, que traduz-se como o CLht necessário para que se obtenha o desejado CLw. Essa formulação é altamente conveniente, pois o controle da elevação da aeronave é justamente, na maioria grande dos casos, dado pela deflexão do aileron do profundor e portanto pode ser ajustada de forma a garantir o requerido CLht.

Deve-se enfatizar que a estabilidade do equilíbrio obtido ainda não foi demonstrada, mas a escolha adequada dos braços li e ângulos γ_i deve implicar na característica restituidora dos momentos e será equacionada apenas posteriormente.

O próximo paço na resolução do sistema é a substituição da função $C_{L_{ht}} = f(C_{L_w})$ nas duas outras equações e o isolamento algébrico das variáveis V e φ . A parametrização pelo CLw acarreta uma outra vantagem: a eliminação da variável θ da equação para a aceleração em y (a projeção da força de propulsão é dada em função de θ). Para isso, recorre-se à substituição:

$$(2.8) \quad \theta = \frac{C_{L_w}}{C_{L_{\alpha_w}}} + \alpha_{L0_w} + \varphi$$

Além dessas simplificações, a linearidade das equações na variável φ garante o direto isolamento analítico da mesma e será extremamente importante no método de resolução proposto a seguir.

2.4. Resolução do sistema

As acelerações do TMB são zeradas e as substituições de θ são realizadas.

$$(2.9) \quad \begin{cases} 0 = T(V \cos(\theta - \varphi)) - D_w - D_{ht} - \varphi L_w - (\varepsilon + \varphi) L_{ht} \\ 0 = \theta T(V \cos(\theta - \varphi)) + L_w + L_{ht} - \varphi D_w - (\varepsilon + \varphi) D_{ht} - W \end{cases}$$

φ é posto em evidência em ambas as equações.

$$(2.10) \quad \begin{cases} D_w + D_{ht} + \varepsilon L_{ht} - T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) \right) = -\varphi (L_w + L_{ht}) \\ L_w + L_{ht} - \varepsilon D_{ht} - W + \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) \right) = \\ = \varphi \left(D_w + D_{ht} - T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} - \alpha_{L0_w} \right) \right) \right) \end{cases}$$

A solução do sistema é apresentada pela tripla igualdade.

$$(2.11) \quad \begin{cases} \varphi = \frac{T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) \right) - D_w - D_{ht} - \varepsilon L_{ht}}{L_w + L_{ht}} = \\ = \frac{L_w + L_{ht} - \varepsilon D_{ht} - W + \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) \right)}{D_w + D_{ht} - T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} - \alpha_{L0_w} \right) \right)} \end{cases}$$

Tendo em vista a natureza do modelo das forças, pode-se encaminhar a obtenção da solução do sistema. Multiplicando em cruz o segundo e terceiro membro da acima, obtém-se um polinômio de quarta ordem na velocidade com seus coeficientes parametrizados por CL_w .

$$\begin{aligned}
(2.12) \quad & T^2 \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) \right) + L_w^2 + 2L_w L_{ht} + L_{ht}^2 - L_w W - L_{ht} W + D_w^2 + 2D_w D_{ht} + D_{ht}^2 + \\
& + \left(\left(\frac{C_{L_w}}{C_{L\alpha_w}} + \alpha_{L0_w} \right) (L_w + L_{ht}) - 2D_w - 2D_{ht} - \varepsilon L_{ht} \right) T \left(V \cos \left(\frac{C_{L_w}}{C_{L\alpha_w}} - \alpha_{L0_w} \right) \right) + \\
& - \varepsilon L_w D_{ht} - \varepsilon L_{ht} D_{ht} + \varepsilon D_w L_{ht} + \varepsilon D_{ht} L_{ht} = 0
\end{aligned}$$

O polinômio de quarta ordem pode ser facilmente resolvido numericamente, sendo que há apenas uma raiz com significado físico. A raiz a ser escolhida pode ser identificada quando o φ associado a ela é calculado, e esse pertence ao intervalo válido na expansão da variável na série de Euler, ou seja, ângulos em torno do zero. Caso isso não ocorra, a aeronave deve encontrar-se em mergulho ao ascensão praticamente verticais o que exige orientação diferenciada da força peso e do sistema de coordenadas.

Para ilustração do que foi discutido até agora e a familiarização do leitor com o fenômeno e a abordagem matemática aqui apresentado, um exemplo de aeronave [11] é formulado abaixo e suas curvas características de vôo em regime permanente são interpretadas.

2.5. Exemplo

bi=2.31;	% Envergadura da asa
Si=1.0672;	% Área da asa
Ari=bi^2/Si;	% Aspect ratio da asa
ci=0.4791;	% Corda media da asa
CLalphai=4.04;	% CLalpha da asa
CD0i=0.0178;	% CD0 da asa
ei=0.8839;	% Oswald da asa
epsiloni=0*pi/180;	% Downwash na asa
CMi=-0.2398;	% CM da asa
CLimax=1.6;	
eipista=2;	
CLipista=1.2;	
CLalphaipista=8.1629;	
alphaL0i=-0.82/CLalphai;	
%-----%	
bp=0.822;	% Envergadura do profundor
Sp=0.27 ;	% Area do profundor
Arp=bp^2/Sp;	% Aspect ratio do profundor
cp=0.329;	% Corda media do profundor
CLalphap=1.5277;	% CLalpha do profundor
CD0p=0.0208;	% CD0 do profundor
epsilonp=4.6972*pi/180;	% Downwash no profundor
ep=0.3379;	% Oswald do profundor
CMp=0;	% CM do profundor
CLpmax=0.9326;	
eppista=0.6552;	
CLppista=-0.39;	
CLalphappista=2.4288;	
alphaL0p=-CLppista/CLalphappista - epsilonp;	

Primeiramente o equilíbrio de momentos é estabelecido e a relação $C_{L_{ht}} = f(C_{L_w})$ é calculada. É notável que os desvios de uma relação linear são pequenos uma vez que os momentos gerados pelo arrasto induzido que apresentam termos quadráticos são de ordem inferior aos outros, pois surgem com braços significativamente menores que os braços das forças de sustentação, o que é uma característica típica de aeronave com aft tail e com pequenos ângulos de cauda.

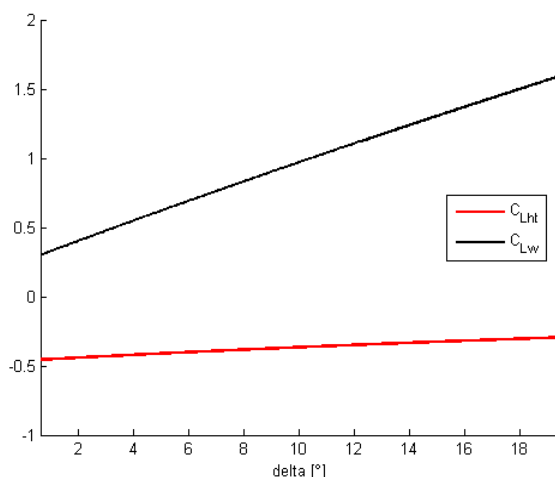


Figura 2.3 - Relação de CL: asa/profundo

Em seguida cada par V , φ é calculado para “todos” os pontos do intervalo da parametrização, ou seja, a faixa de operação do CL da asa é varrida e a cada ponto um V e φ são atribuídos. Neste momento seria possível verificar uma relação não única entre CL e (V, φ) . No entanto, o comportamento monotônico das forças aerodinâmicas garante a unicidade da relação.

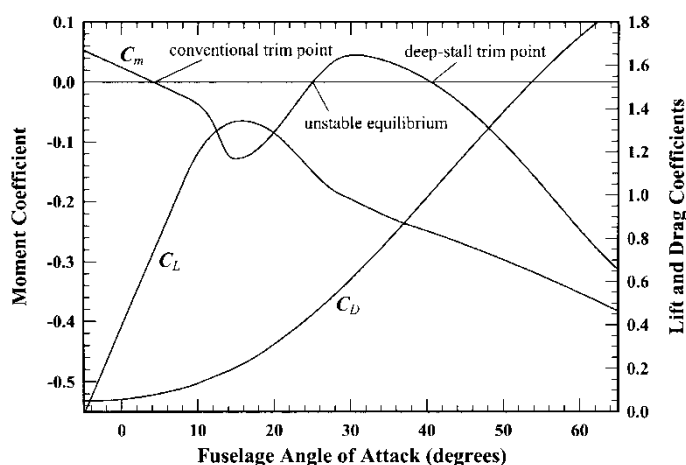


Figura 2.4 - Comportamento geral dos coeficientes

Por outro lado, o Stall não é um fenômeno discreto, após o ponto de sustentação máxima, um acréscimo ainda maior do ângulo de incidência do escoamento leva a um decréscimo gradual de sustentação e aumento de arrasto. Caso deseje-se operar nesta faixa pós Stall, a parametrização aqui adotada não é indicada, sendo o ângulo efetivo de ataque a alternativa sugerida.

Obviamente, a baixa eficiência do voo em tal condição e o comprometimento da estabilidade pela mudança de sinal da derivada do CL torna o estudo desinteressante, a não ser por condições emergenciais de recuperação, o que não é enfoque deste trabalho.

Uma vez obtidos os pares (V, φ) , o mapa de voo pode ser traçado conforme a figura abaixo.

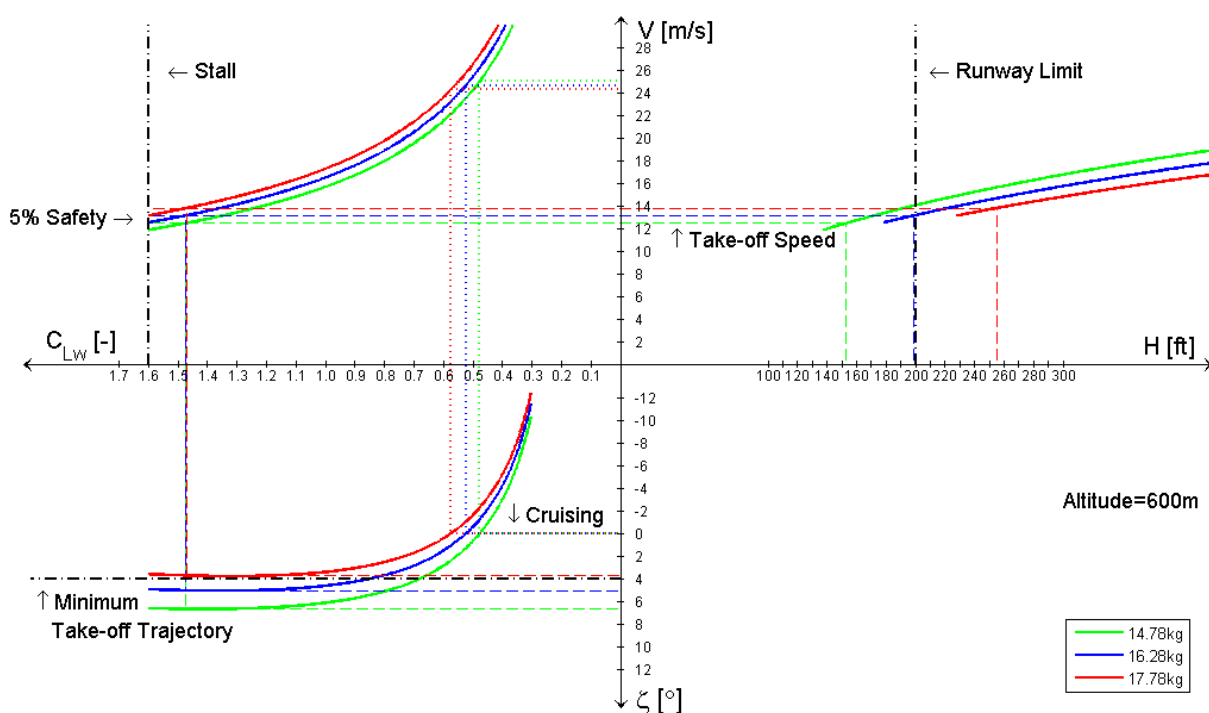


Figura 2.5 - Mapa de voo 2D

O resultado ilustrado na figura acima fornece uma noção da influência da variação de massa e do limite de Stall nas condições de voo.

A relação V, φ para a faixa de CL é altamente coerente com o exercício de voo em que se atinge o Stall de primeiro tipo. A relação de acréscimo da velocidade e decréscimo do ângulo de trajetória e CL da asa mostrou-se tal como fisicamente esperada.

A influência do vento é altamente significativa. Neste capítulo, assumiu-se vento alinhado com a horizontal, o que implica apenas numa translação das curvas de velocidade

numa mudança da trajetória aparente, sem alteração na componente vertical da velocidade. No capítulo 4, correntes de ar em direção genérica serão computadas.

Outro resultado extremamente importante é o limite superior da taxa de subida, o que define restrições decisivas no estudo da decolagem. O equacionamento da corrida em pista e a transição do regime suportado pelo conjunto de trem de pouso ao regime sustentado de vôo serão temas de próximos capítulos.

2.6. Para reflexão do leitor:

Assumindo uma aeronave estável qual é o limite para uma aproximação de vôo quasi-estático?

3. Corrida em pista

A corrida em pista é um caso particular das equações dinâmicas desenvolvidas no capítulo anterior. Assume-se que a força normal equilibra a aeronave na direção vertical, de forma a restringir a aplicabilidade da abordagem discutida a seguir até o instante em que se inicia a rolagem.

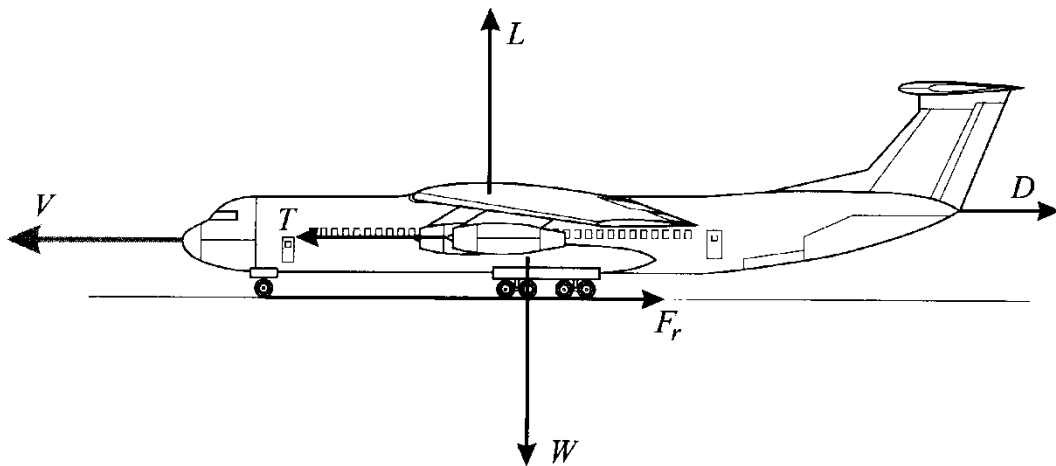


Figura 3.1 - Corrida em pista

Equação diferencial da velocidade do aeromodelo com a inclusão do termo de atrito com o solo adquire a seguinte forma.

$$(3.1) \quad m \frac{dV}{dt} = T - D - \mu N = f_1(V_{ap}) - \frac{n\rho V_{ap}^2}{2} \left[SC_{D0} + \frac{S^2 C_L^2}{\pi b^2 e} \right] - \mu mg + \frac{1}{2} \rho V_{ap}^2 n \mu SC_L$$

Condições iniciais:

$$(3.2) \quad \begin{cases} V(0) = V_0 \\ x(0) = x_0 \end{cases}$$

Essa equação diferencial pode ser resolvida analiticamente pelo método das variáveis separáveis e é extremamente útil como referência no teste e validação de simulações numéricas.

$$(3.3) \quad \int_{V_0}^V \frac{m}{T - D - \mu N} dV = \int_0^t dt \xrightarrow{V_{ap}=V+V_{vento}} \int_{V_0+V_{vento}}^{V+V_{vento}} \frac{m}{T - D - \mu N} dV_{ap} = t(V)$$

$$(3.4) \quad t(V) = m \int_{V_0+V_{vento}}^{V+V_{vento}} \frac{dV_{ap}}{b_0 + b_1 V_{ap} + b_2 V_{ap}^2}$$

$$(3.5) \quad \begin{cases} b_0 = a_0 - \mu mg \\ b_1 = a_1 \\ b_2 = a_2 - \frac{n\rho}{2} \left[SC_{D0} + \frac{S^2 C_L^2}{\pi b^2 e} - \mu SC_L \right] \end{cases}$$

Para se resolver a integral (3.4), devemos primeiramente encontrar as raízes do polinômio em seu denominador.

$$(3.6) \quad \bar{V}_{1,2} = \frac{-b_1 \pm \sqrt{b_1^2 - 4b_2 b_0}}{2b_2}$$

Dessa forma, fatora-se o polinômio e encontram-se as constantes A e B.

$$(3.7) \quad \frac{1/b_2}{b_0/b_2 + V_{ap} b_1/b_2 + V_{ap}^2} = \frac{1}{b_2} \frac{A(V_{ap} - \bar{V}_2) + B(V_{ap} - \bar{V}_1)}{(V_{ap} - \bar{V}_1)(V_{ap} - \bar{V}_2)}$$

$$(3.8) \quad \begin{cases} A + B = 0 \\ A\bar{V}_2 + B\bar{V}_1 = -1 \end{cases} \Rightarrow \begin{cases} A = \frac{-1}{\bar{V}_2 - \bar{V}_1} \\ B = \frac{1}{\bar{V}_2 - \bar{V}_1} \end{cases}$$

Resolve-se a integral e defini-se $t(V)$.

$$(3.9) \quad \int_{V_0+V_{vento}}^{V+V_{vento}} \left[\frac{A}{(V_{ap} - \bar{V}_1)} + \frac{B}{(V_{ap} - \bar{V}_2)} \right] dV_{ap} = \frac{1}{(\bar{V}_2 - \bar{V}_1)} \left[-\ln|V_{ap} - \bar{V}_1| + \ln|V_{ap} - \bar{V}_2| \right]_{V_0+V_{vento}}^{V+V_{vento}}$$

$$(3.10) \quad t(V) = \frac{m/b_2}{(\bar{V}_2 - \bar{V}_1)} \left[\ln \left| \frac{V + V_{vento} - \bar{V}_2}{V + V_{vento} - \bar{V}_1} \right| + \ln \left| \frac{V_0 + V_{vento} - \bar{V}_1}{V_0 + V_{vento} - \bar{V}_2} \right| \right]$$

$$(3.11) \quad \tilde{V}_{1,2} = \bar{V}_{1,2} - V_{vento}$$

A função obtida em (3.10) é a inversa da procurada, no entanto podemos obter $x(V)$ por subtração de áreas. A expressão abaixo descreve essa técnica de inversão de funções.

$$(3.12) \quad \Delta x \Big|_{V_1}^{V_2} = x_2 - x_1 = \int_{t(V_1)}^{t(V_2)} V(t) dt = V_2 t(V_2) - V_1 t(V_1) - \int_{V_1}^{V_2} t(V) dV$$

Integrando $t(V)$, obtem-se a função necessária para que (3.12) fique completamente determinada.

$$(3.13) \quad \int_{V_1}^{V_2} t(V) dV = \frac{m/b_2}{(\tilde{V}_2 - \tilde{V}_1)} \left\{ \left[(V - \tilde{V}_2) \ln |V - \tilde{V}_2| - (V - \tilde{V}_1) \ln |V - \tilde{V}_1| \right]_{V_1}^{V_2} + (V_2 - V_1) \ln \left| \frac{V_0 - \tilde{V}_1}{V_0 - \tilde{V}_2} \right| \right\}$$

Para $x_0 = V_0 = x_1 = V_1 = 0$, segue:

$$(3.14) \quad x(V) = \frac{m/b_2}{(\tilde{V}_2 - \tilde{V}_1)} \left[\tilde{V}_2 \ln \left| \frac{V - \tilde{V}_2}{\tilde{V}_2} \right| - \tilde{V}_1 \ln \left| \frac{V - \tilde{V}_1}{\tilde{V}_1} \right| \right]$$

Para $x_0 = x_1$ e $V_0 = V_1$, segue:

$$(3.15) \quad x(V) - x_0 = \frac{m/b_2}{(\tilde{V}_2 - \tilde{V}_1)} \left[\tilde{V}_2 \ln \left| \frac{V - \tilde{V}_2}{V_0 - \tilde{V}_2} \right| - \tilde{V}_1 \ln \left| \frac{V - \tilde{V}_1}{V_0 - \tilde{V}_1} \right| \right]$$

E eq. (3.15) fornece o comprimento necessário de pista para que a aeronave atinja uma dada velocidade desejada.

4. Equacionamento Tridimensional

4.1. Discussão inicial

A abordagem bidimensional do problema mostra-se suficiente para análises de casos específicos como decolagem e vôo retilíneo com vento alinhado. Por outro lado, a exploração completa do fenômeno requer a descrição do movimento no espaço, abrindo um leque de dificuldades matemáticas, como por exemplo, momentos giroscópicos altamente não lineares e velocidades aparentes induzidas por rotações de corpo rígido. Dessa forma, um estudo minucioso do modelo pode permitir a eliminação de termos inconvenientes para o desenvolvimento analítico, “limpando” as equações de maneira a garantir um manuseio intuitivo para o projetista e possibilitar otimizações por meio de derivadas explícitas e a localização de pontos ótimos.

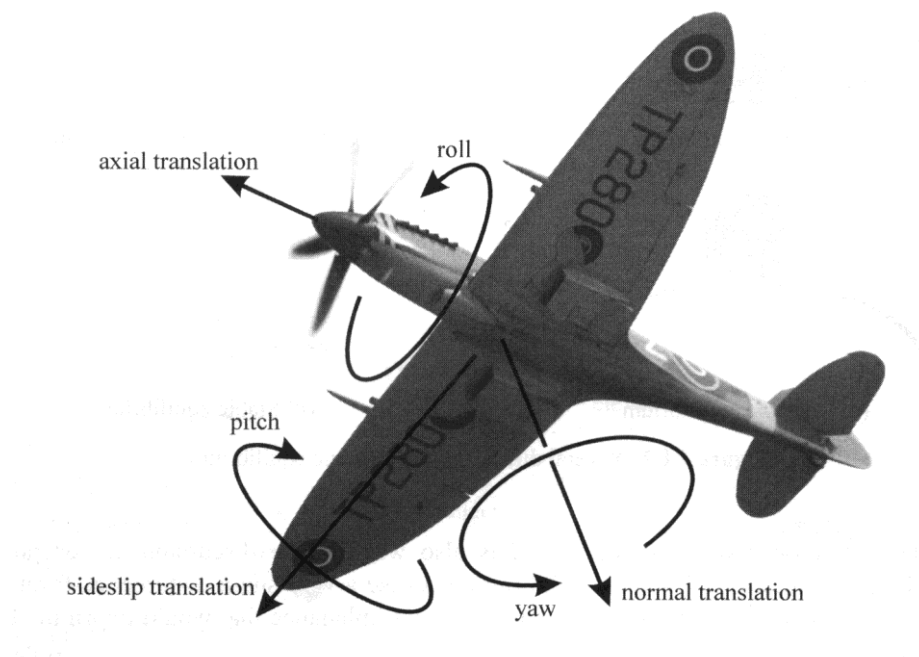


Figura 4.1 - 6 graus de liberdade

Na bibliografia usual de graduação certas simplificações cinemáticas são usualmente adotadas, mas raramente justificadas. Logo, o formando propõe-se a deduzir sem aproximações as equações do movimento e discutir tais simplificações usuais e suas implicações. Dificuldades são esperadas como consequência da introdução dos modelos aerodinâmicos, pois essas forças e momentos apresentam-se normalmente acopladas e têm influências em diversos graus de liberdade simultaneamente. Portanto, a eliminação de certos

efeitos cruzados pode tornar-se imprescindível para a abordagem analítica, o que levanta o questionamento sobre perdas e ganhos em capacidade de projeto nessa diretriz.

Exceto pelo caso de voo em parafuso, qualquer regime de voo permanente pode ser projetado por sobre uma helicóide. O limite para o raio infinito da hélice leva ao caso de voo retilíneo. Tendo esse espaços de soluções em mente, pretende-se direcionar o desenvolvimento das equações em sua escolha de bases de forma a explicitar de maneira isolada as variáveis: raio, velocidade periférica e ângulo de hélice.

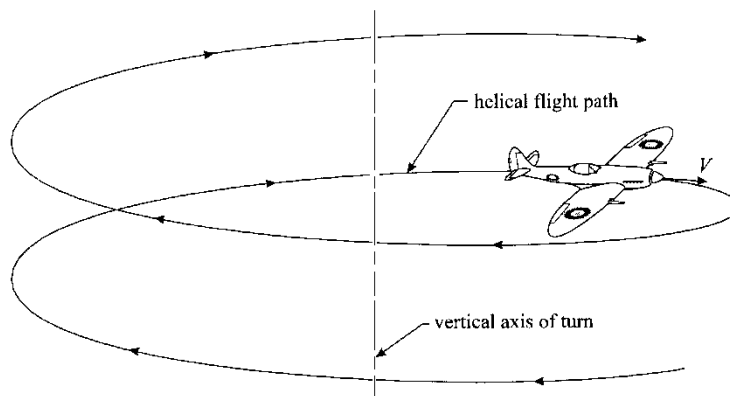


Figura 4.2 - Voo helicoidal

Uma vez que os modelos aerodinâmicos fornecem coeficientes para forças sempre paralelas ou perpendiculares ao escoamento, o alinhamento do sistema de coordenadas com o mesmo tende a evitar diversas componentes de projeção dessas forças. Logo, uma base que possua um versor na direção da velocidade, outro na direção da aceleração centrípeta e o terceiro de forma a garantir uma base ortogonal, deve implicar em equações do movimento naturalmente desacopladas. Analogamente, para as rotações podem ser utilizados os ângulos efetivos de ataque das superfícies sustentadoras para o direcionamento da escolha da base.

4.2. TMB

Para a formulação da mudança de coordenadas a ser introduzida abaixo, parte-se das coordenadas do centro de massa da aeronave, x, y, z em relação a um sistema cartesiano inercial e de u, v, w , que representam as componentes do vento nas três direções desse sistema. O novo sistema deve possuir o módulo da velocidade e dois ângulos que fornecem a direção da mesma como novas coordenadas. Pode-se dizer que o novo sistema assemelha-se fortemente com um sistema esférico de coordenadas, sendo que o módulo da velocidade aparente do escoamento substitui a coordenada do raio.

$$(4.1) \quad \begin{cases} \dot{x} - u = V \cos \varphi \cos \psi \\ \dot{y} - v = V \cos \varphi \sin \psi \\ \dot{z} - w = V \sin \varphi \end{cases}$$

Derivando a quantidade de movimento nas três direções e aplicando a 2ª Lei de Newton obtêm-se as equações diferenciais do movimento do centro de massa. No entanto, ainda necessita-se de um certo trabalho algébrico para que as acelerações se desacoplem.

$$(4.2) \quad \begin{cases} \frac{d(m\dot{x})}{dt} = \frac{d(mV \cos \varphi \cos \psi + mu)}{dt} = F_x \\ \frac{d(m\dot{y})}{dt} = \frac{d(mV \cos \varphi \sin \psi + mv)}{dt} = F_y \\ \frac{d(m\dot{z})}{dt} = \frac{d(mV \sin \varphi + mw)}{dt} = F_z \end{cases}$$

$$\begin{pmatrix} m\ddot{x} + m\dot{x} \\ m\ddot{y} + m\dot{y} \\ m\ddot{z} + m\dot{z} \end{pmatrix} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} = \begin{pmatrix} \dot{m}V \cos \varphi \cos \psi + m\dot{V} \cos \varphi \cos \psi - m\dot{\varphi}V \sin \varphi \cos \psi - m\dot{\psi}V \cos \varphi \sin \psi + \dot{m}u + m\dot{u} \\ \dot{m}V \cos \varphi \sin \psi + m\dot{V} \cos \varphi \sin \psi - m\dot{\varphi}V \sin \varphi \sin \psi + m\dot{\psi}V \cos \varphi \cos \psi + \dot{m}v + m\dot{v} \\ \dot{m}V \sin \varphi + m\dot{V} \sin \varphi + m\dot{\varphi}V \cos \varphi + \dot{m}w + m\dot{w} \end{pmatrix}$$

Conforme discutido, é de se esperar que as equações acima apresentem-se de forma mais simples quando descritas na base que se alinha com a velocidade e o centro instantâneo de curva.

Analisando as direções dos versores de tal base, pode-se facilmente definir a matriz de mudança de base que efetua a transformação desejada.

$$(4.3) \quad K = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\} \quad K' = \{\hat{t}; \hat{r}; \hat{n}\} = \left\{ \begin{pmatrix} \cos \varphi \cos \psi \\ \cos \varphi \sin \psi \\ \sin \varphi \end{pmatrix}; \begin{pmatrix} -\sin \psi \\ \cos \psi \\ 0 \end{pmatrix}; \begin{pmatrix} -\sin \varphi \cos \psi \\ -\sin \varphi \sin \psi \\ \cos \varphi \end{pmatrix} \right\}$$

$$M_{K'K} = \begin{pmatrix} \cos \varphi \cos \psi & \cos \varphi \sin \psi & \sin \varphi \\ -\sin \psi & \cos \psi & 0 \\ -\sin \varphi \cos \psi & -\sin \varphi \sin \psi & \cos \varphi \end{pmatrix}$$

Aplicando a transformação nas equações diferenciais, o sistema resultante torna-se satisfatoriamente simples, uma vez que os diversos e inconvenientes senos e cossenos dos ângulos de coordenada permanecem presentes apenas para os casos que estudam a influência dos ventos.

$$(4.4) \quad M_{K'K} \begin{pmatrix} m\dot{x} + m\ddot{x} \\ m\dot{y} + m\ddot{y} \\ m\dot{z} + m\ddot{z} \end{pmatrix}_K = \begin{pmatrix} F_t \\ F_r \\ F_n \end{pmatrix}_{K'} = m \begin{pmatrix} \dot{V} \\ \psi \dot{V} \cos \varphi \\ \dot{\varphi} V \end{pmatrix}_{K'} + M_{K'K} \begin{pmatrix} m \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + \dot{m} \begin{pmatrix} V \cos \varphi \cos \psi + u \\ V \cos \varphi \sin \psi + v \\ V \sin \varphi + w \end{pmatrix} \end{pmatrix}_K$$

$$(4.5) \quad m \begin{pmatrix} \dot{V} \\ \psi \dot{V} \cos \varphi \\ \dot{\varphi} V \end{pmatrix}_{K'} = \begin{pmatrix} F_t - \dot{m} V \\ F_r \\ F_n \end{pmatrix}_{K'} - M_{K'K} \begin{pmatrix} m \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} + \dot{m} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \end{pmatrix}_K$$

Deve-se recordar que o primeiro objetivo é a obtenção da família de soluções helicoidais de regime permanente. Para esses casos, o sistema simplifica-se ainda mais com a introdução da relação cinemática $\dot{\psi} = \frac{V \cos \varphi}{R}$.

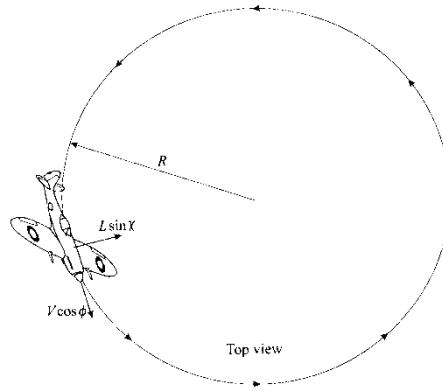


Figura 4.3 - Vista superior

A variação da massa no tempo será considerada nula pois a condição buscada representa um “equilíbrio dinâmico” em que a massa deve variar de forma suficientemente lenta para que se possa tratar de uma massa constante no intervalo.

Implicações da influência da rotação da terra [7] e [9] poderiam ter sido introduzidas no início da seção, mas tal sofisticação apenas desviaria o foco da discussão e portanto, será considerada não pertencente ao escopo do trabalho.

As resultantes de forças são compostas pelo sistema de propulsão, as forças aerodinâmicas e a força peso. Após a dedução das equações do movimento para o momento angular, tanto as forças quanto momentos resultantes serão equacionados.

$$(4.6) \quad \begin{pmatrix} 0 \\ \frac{mV^2 \cos^2 \varphi}{R} \\ 0 \end{pmatrix}_{K'} = \begin{pmatrix} F_t \\ F_r \\ F_n \end{pmatrix}_{K'}$$

4.3. TMA

Após o alinhamento da base K' com a velocidade, mais duas rotações são necessárias para que os versores concordem com os eixos de referência da aeronave. A rotação pelo ângulo χ em torno do eixo da velocidade permite a inclinação lateral da aeronave e implica diretamente em componentes centrais da sustentação ou seja, trajetórias curvilíneas. A rotação subsequente é dada pelo ângulo θ e define o ângulo de ataque entre o eixo longitudinal de referência e a velocidade. Esse último ângulo terá implicação direta no ângulo efetivo de ataque enxergado pela asa.

Vale ressaltar que em se tratando de rotações finitas, a ordem de tais operações é absolutamente relevante, pois o eixo de referência para a próxima rotação é consequência das rotações anteriores.

As matrizes de rotação são dedutíveis de forma simples mais trabalhosa e portanto serão apenas apresentadas.

A três bases deve o leitor se familiarizar para que a leitura e compreensão das passagens seguintes seja clara.

Sucintamente podem-se descrever as bases da seguinte maneira: A primeira é a base K' (mesma do TMB, cujos versores alinham-se respectivamente com a velocidade, a direção central da helicóide e ao produto vetorial dos dois primeiros); A base K'' é fruto da rotação χ , e inclina o horizonte baixando a ponta esquerda da asa e levantando a direita quando positiva. A última base, K''' , resulta da rotação de θ , parâmetro que controla a elevação do nariz da aeronave.

$$\begin{aligned}
 M_{K'K} &= \begin{pmatrix} \cos \varphi \cos \psi & \cos \varphi \sin \psi & \sin \varphi \\ -\sin \psi & \cos \psi & 0 \\ -\sin \varphi \cos \psi & -\sin \varphi \sin \psi & \cos \varphi \end{pmatrix} \\
 K'' = \{\hat{t}; \hat{s}, \hat{p}\} \quad M_{K''K'} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \chi & -\sin \chi \\ 0 & \sin \chi & \cos \chi \end{pmatrix} \\
 K''' = \{\hat{a}; \hat{b}; \hat{c}\} \quad M_{K'''K''} &= \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}
 \end{aligned}
 \tag{4.7}$$

Não é conveniente apresentá-las na forma multiplicada, mas sim, para a visualização, a idéia de operadores a serem encadeados é mais adequada.

É admitido que a velocidade do centro de massa permanece contida no plano de simetria longitudinal da aeronave, ou seja, o CG não passará a enxergar o escoamento com uma componente lateral após a realização dessas rotações. É natural aceitar que os regimes de vôo buscados possuem todos tal característica de alinhamento do eixo da aeronave com a velocidade pois intuitivamente relacionam-se a conforto e eficiência.

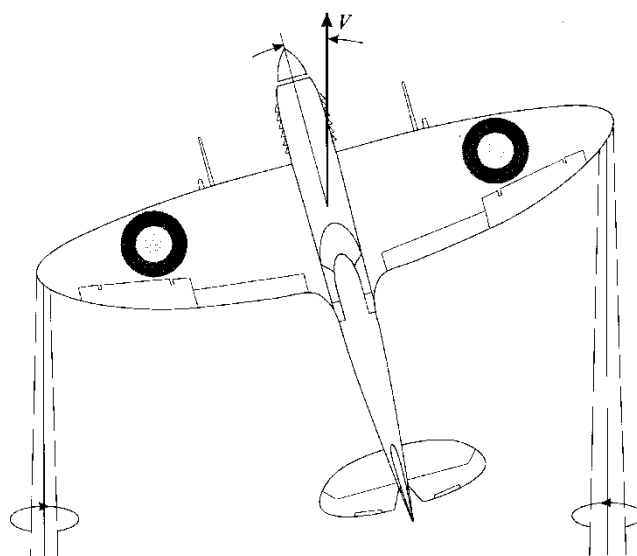


Figura 4.4 - Derrapagem

Por outro lado, é imprescindível lembrar que a relações cinemáticas de corpo rígido implicam em velocidades aparentes diferentes ao longo da asa. Tais efeitos devem ser cuidadosamente discutidos na modelagem das forças aerodinâmicas, mas não têm influência sobre a aplicação do TMA.

As mudanças de base foram realizadas principalmente com o intuito de obter um sistema fixo ao corpo em que o tensor de inércia não varie. Obviamente, o consumo de combustível necessariamente implica em uma variação temporal desse tensor, mas a utilidade prática da fixação da base ao corpo e se possível a concordância da mesma com os eixos centrais de inércia ainda prevalece.

A base K''' , possui as características enunciadas acima. Logo, o TMA será aplicado de forma a ser descrito em relação a base K''' , tomado o imprescindível cuidado com o fato da mesma não se tratar de um referencial inercial.

Caso o eixo de referência longitudinal para a definição do ângulo de arfagem seja tomado como paralelo ao solo quando a aeronave encontra-se em pista, o que comumente feito, genericamente o tensor de inércia apresentará um de seus produtos de inércia não nulo na base K''' .

Produtos de inércia introduzem acoplamentos altamente indesejados e portanto será aqui convencionado que os eixos centrais de inércia serão sempre utilizados como referência. Para isso basta apenas redefinir o zero aerodinâmico (referência para os ângulos de ataque), o que será discutido posteriormente. Impondo os eixos centrais de inércia coincidentes com K''' , tem-se:

$$(4.8) \quad J_{K'''K'''} = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix}_{K'''K'''}$$

O Momento angular do sistema, escrito na base fixa ao corpo é, conseqüentemente:

$$(4.9) \quad (\vec{H}_{CG})_{K'''} = J_{K'''K'''}(\vec{\Omega})_{K'''}$$

A derivação necessária para a aplicação do TMA deve ser executada obrigatoriamente em relação a um referencial inercial. Por outro lado, é desejado que se mantenham explícitos os vetores de momentos em relação à base K''' .

$$(4.10) \quad \sum (\vec{M}_{CG})_K = \frac{d((\vec{H}_{CG})_K)}{dt} = \frac{d(M_{KK'''}(\vec{H}_{CG})_{K'''})}{dt}$$

$$M_{K'''K} \sum (\vec{M}_{CG})_K = \sum (\vec{M}_{CG})_{K'''} = M_{K'''K} \frac{d(M_{KK'''} J_{K'''K'''} M_{K''K}(\vec{\Omega})_K)}{dt}$$

A partir da igualdade acima, pode-se realizar a derivação com total pertinência física, garantindo que as influências não-inerciais da base móvel irão surgir naturalmente no sistema de equações diferenciais.

$$(4.11) \quad \sum (\vec{M}_{CG})_{K'''} = J_{K'''K'''}(\dot{\vec{\Omega}})_{K'''} + M_{K'''K} \dot{M}_{KK'''} J_{K'''K'''}(\vec{\Omega})_{K'''} + J_{K'''K'''} \dot{M}_{K''K}(\vec{\Omega})_K + J_{K'''K'''} M_{K''K}(\dot{\vec{\Omega}})_K$$

Possivelmente o leitor está familiarizado com uma forma análoga do conjuntos de equações acima, que é normalmente apresentada em livros de mecânica vetorial [10].

Os termos são idênticos sendo que os dois últimos da equação deduzida equivalem ao terceiro termo da equação na sua forma mais clássica apresentada abaixo.

$$(4.12) \quad \sum (\vec{M}_{CG})_{K'''} = \dot{J}_{K''K'''} (\vec{\Omega})_{K'''} + (\vec{\Omega} \times \vec{H}_{CG})_{K'''} + J_{K''K'''} \frac{d((\vec{\Omega})_{K'''})}{dt}$$

Para que as equações diferenciais fiquem completamente determinadas, deve-se substituir o vetor de rotação na equação acima.

Uma vez que se busca o regime permanente helicoidal, a única rotação à qual o sistema está submetido é $\dot{\psi}$ constante:

$$(4.13) \quad (\vec{\Omega})_K = \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}_K \Rightarrow (\vec{\Omega})_{K''} = \dot{\psi} \begin{pmatrix} \sin \varphi \cos \theta + \cos \varphi \cos \chi \sin \theta \\ -\cos \varphi \sin \chi \\ -\sin \varphi \sin \theta + \cos \varphi \cos \theta \cos \chi \end{pmatrix}_{K''}$$

Logo, o único termo não nulo do segundo membro é:

$$(4.14) \quad (\vec{\Omega} \times \vec{H}_{CG})_{K''} = \dot{\psi}^2 \begin{vmatrix} \hat{a} & \hat{b} & \hat{c} \\ \sin \varphi \cos \theta + \cos \varphi \cos \chi \sin \theta & -\cos \varphi \sin \chi & -\sin \varphi \sin \theta + \cos \varphi \cos \theta \cos \chi \\ J_{xx}(\sin \varphi \cos \theta + \cos \varphi \cos \chi \sin \theta) & -J_{yy} \cos \varphi \sin \chi & J_{zz}(-\sin \varphi \sin \theta + \cos \varphi \cos \theta \cos \chi) \end{vmatrix} =$$

$$= \dot{\psi}^2 \begin{pmatrix} (J_{zz} - J_{yy})(\cos \varphi \sin \varphi \sin \chi \sin \theta - \cos^2 \varphi \sin \chi \cos \theta \cos \chi) \\ (J_{xx} - J_{zz})(1/2 \sin 2\theta (\cos^2 \varphi \cos^2 \chi - \sin^2 \varphi) + \sin \varphi \cos \varphi \cos \chi \cos 2\theta) \\ (J_{yy} - J_{xx})(-\sin \varphi \cos \varphi \cos \theta \sin \chi - \cos^2 \varphi \cos \chi \sin \theta \sin \chi) \end{pmatrix}_{K''}$$

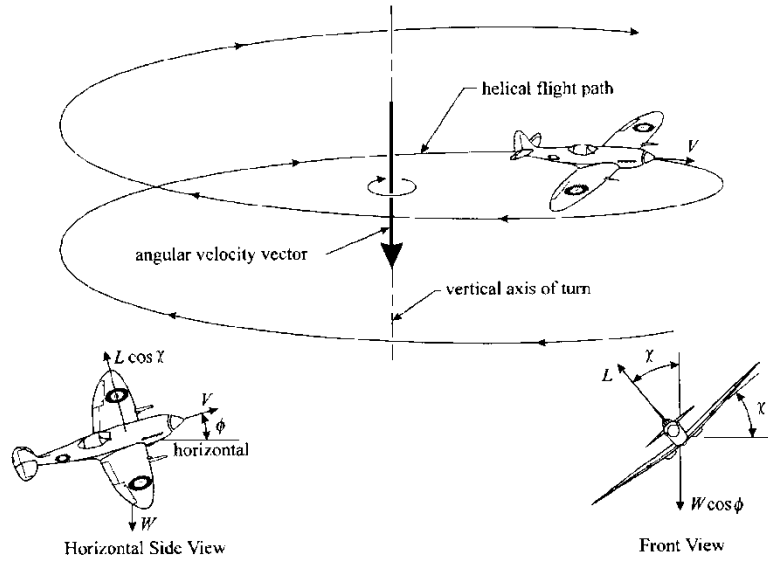


Figura 4.5 - Ângulos de trajetória

Utilizando novamente a relação cinemática, $\dot{\psi} = \frac{V \cos \varphi}{R}$, obtêm-se as equações de “equilíbrio dinâmico” para o momento angular, explícitas apenas nas variáveis livres: V , R , φ , θ , χ .

$$(4.15) \quad \sum (\bar{M}_{CG})_{K''} = \frac{V^2 \cos^4 \varphi}{R^2} \begin{pmatrix} (J_{zz} - J_{yy})(\tan \varphi \sin \theta - \cos \theta \cos \chi) \sin \chi \\ (J_{xx} - J_{zz}) \left(\frac{1}{2} \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi \right) \\ (J_{yy} - J_{xx})(-\tan \varphi \cos \theta - \sin \theta \cos \chi) \sin \chi \end{pmatrix}_{K''}$$

Antes que se prossiga para a introdução das forças e momentos aerodinâmicos no sistema, é conveniente ressaltar que o problema discutido possui 6 graus de liberdade e 6 equações do movimento foram deduzidas. No entanto, até o momento apenas 5 variáveis livres foram utilizadas (a simetria de guinada elimina a sexta variável). Isso não significa sobre-determinação do sistema, pois ainda devem-se contabilizar mais três variáveis oriundas das superfícies móveis, leme, aileron e profundor.

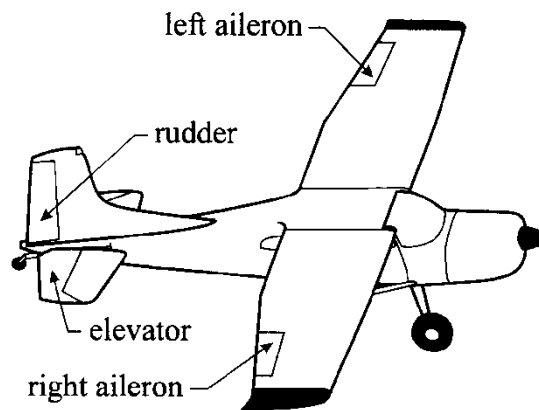


Figura 4.6 - Superfícies de comando

Deve-se entender que apenas V , R e φ estão relacionadas à trajetória, as deflexões das superfícies relacionam-se intimamente com o equilíbrio de momentos, enquanto θ e χ representam os “ângulos do horizonte” quando visto de dentro da aeronave. Tendo essa interpretação em mente, a abordagem de resolução do sistema começa a se sugerir.

Para o objetivo de se traçar uma carta de vôo em que constam todas as possíveis trajetórias, da mesma forma como feito no problema bidimensional, é interessante a introdução de uma parametrização do universo de soluções. As variáveis θ e χ são candidatas ideais por possuírem fácil interpretação (ângulo efetivo ataque e ângulo do horizonte, respectivamente), por sua relação física com a trajetória descrita (a imagem vista pelo piloto) e principalmente porque dessa forma passa-se a ter um sistema 6X6 com uma família de soluções de dois parâmetros.

Naturalmente, as diferentes condições de vôo serão determinadas pelo conjunto de atuação das superfícies de controle. Tais superfícies geram as forças e momentos necessários para que se atinja o equilíbrio no ponto de operação desejado. Logo, para um dado par de parâmetros θ e χ , o equilíbrio de momentos deve ser primeiramente estabelecido e, dessa forma, as atuações necessárias bem como seus conseqüentes arrastos e sustentações implicarão em uma velocidade, raio de curva e ângulo de subida.

4.4. Esforços Aerodinâmicos

Nesta seção será necessário lidar com as principais incertezas do modelo e algumas simplificações. A vantagem de desacoplar a mecânica de corpo rígido da aeronave da mecânica do fluido a sua volta permite que as forças comunicadas pelo fluido à aeronave sejam vistas como forças de contato externas. Tais forças devem ser modeladas de forma a depender apenas dos estados da aeronave, posições angulares e velocidade. Para regime transitórios é imprescindível assumir que a dinâmica do fluido é estável e converge significativamente mais rapidamente que a dinâmica do sólido, pois é necessária a abstração de que instante a instante o escoamento teve o tempo de “formar” suas forças, mesmo com o movimento inconstante das superfícies sobre as quais ocorre o escoamento.

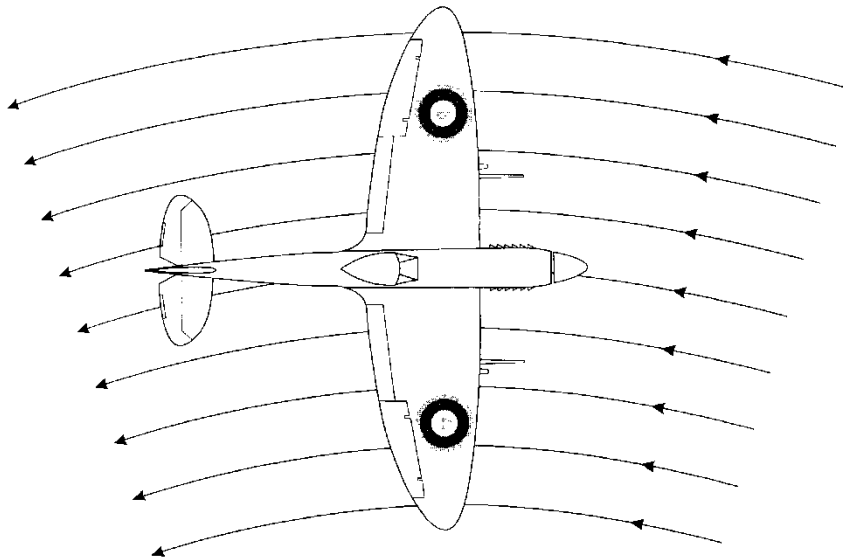


Figura 4.7 - Perfil de velocidades induzido por curva

No caso aqui abordado, o eixo de rotação apresenta direção constante quando visto pela aeronave. Apesar dessa constância, ao se introduzir a rotação de corpo rígido nas as velocidades aparentes e ângulos efetivos de ataque, surge uma dependência dos mesmos com distância do centro de massa.

$$(4.16) \quad \vec{V}_{ap} = V\hat{t} + \vec{\Omega} \times \vec{r}$$

$$(4.17) \quad \alpha_{ef} = -\arctan \left(\frac{\vec{V}_{ap} \cdot \hat{c}}{\vec{V}_{ap} \cdot \hat{a}} \right)$$

Em geral, a velocidade média não sofre alterações significativas, uma vez que a semi-asa de dentro da curva passa a ser exposta a velocidades levemente menores e vice-versa. Além disso, uma leve alteração no ângulo de ataque introduz um chamado twist aerodinâmico, ou torção, que possui influências diretas sobre a eficiência da asa e sustentação gerada [2].

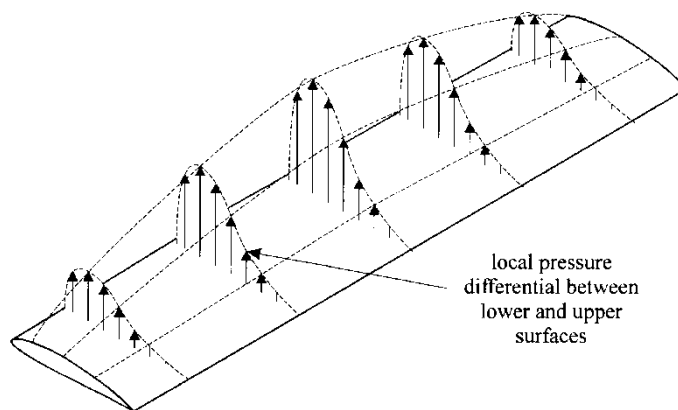


Figura 4.8 - Linhas de sustentação

A ferramenta de linhas de sustentação permite o cálculo dessas influências e fornece coeficientes adequados para condição de torção aerodinâmica e perfil de velocidades variado. É possível inserir esta ferramenta dentro do sistema de equações do movimento e calcular instante a instante, condição a condição, o comportamento do escoamento e suas forças e momentos resultantes. Contudo, a introdução de sistemas lineares de ordem elevada (>20) seria uma elegante sofisticação do ponto de vista aerodinâmico, mas não traria benefícios para a abordagem essencialmente voltada para o corpo rígido.

Portanto, ângulos e velocidades necessárias para o cálculo das forças serão aqui tomados como sendo aqueles do centro de massa e admitir-se-á que os coeficientes e eficiências aerodinâmicos já foram devidamente corrigidos ou por abordagens teóricas ou por dados experimentais. Ainda assim nada impede que esse acoplamento fluído-mecânico seja introduzido posteriormente.

Para a tranquilidade do projetista, note-se que essa simplificação é altamente pertinente desde que os raios de curva permaneçam ordens de grandeza maiores do que a envergadura. A análise de ordem de grandeza realizada no próximo capítulo visa justamente justificar e criticar essa e outras simplificações.

Lembrando que por definição a força de sustentação e a força de arrasto têm respectivamente a direção perpendicular e paralela ao escoamento, escrevem-se seus

respectivos vetores, assim como genericamente também pode haver uma força ortogonal às outras duas.

$$(4.18) \quad (\vec{L})_{K''} = L \begin{pmatrix} \sin \theta \\ 0 \\ \cos \theta \end{pmatrix}_{K''} \quad (\vec{D})_{K''} = D \begin{pmatrix} -\cos \theta \\ 0 \\ \sin \theta \end{pmatrix}_{K''} \quad (\vec{B})_{K''} = B \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}_{K''}$$

O centro aerodinâmico de pressão da superfície em questão apresenta um braço em relação ao centro de massa, que de forma genérica em coordenadas polares é definido por:

$$(4.19) \quad (\vec{r}_{CA})_{K''} = \begin{pmatrix} -l_{CA} \cos \gamma_{CA} \\ 0 \\ l_{CA} \sin \gamma_{CA} \end{pmatrix}_{K''}$$

Fazendo o produto vetorial do braço pelas forças e acrescentando os momentos aerodinâmicos, calcula-se o momento total que aquela superfície promove ao CG.

$$(4.20) \quad (\vec{M}_{CA})_{K''} = (\vec{r}_{CA} \times (\vec{L} + \vec{D} + \vec{B}))_{K''} + (\vec{M}_{MA})_{K''}$$

$$(4.21) \quad (\vec{M}_{MA})_{K''} = \begin{pmatrix} -Mr \\ -Mp \\ My \end{pmatrix}_{K''}$$

É importante lembrar que o leme encontra-se normalmente na vertical e gera sustentação para os lados.

4.5. Convenções

A seguinte convenção será adotada deste ponto em diante:

4.5.1. Índices Subscritos

“w”: Identifica grandezas exclusivamente referentes à asa.

“ht”: Diz respeito ao Profundor ou Horizontal Tail.

“f”: Alude à Fuselagem.

“vt”: Refere-se ao Leme ou Vertical Tail.

“t”: Trata de parâmetros da Propulsão.

4.5.2. Forças e Momentos de Superfícies

L: Força de Sustentação normal simultaneamente ao escoamento e à direção da envergadura.

D: Força de Arrasto na direção do Escoamento.

B: Componente de força na direção da envergadura.

M_p: Momento de Arfagem ou Pitch.

M_r: Momento de Rolagem ou Row.

M_y: Momento de Guinada ou Yaw.

l: Distância entre o centro aerodinâmico da superfície em questão e o CG.

γ : Ângulo polar da distância l medido a partir do eixo longitudinal central de inércia no sentido da cauda para o nariz.

4.5.3. Forças e Momentos de Propulsão

T: Empuxo normal ao plano da hélice ou turbina.

N: Componente vertical do empuxo contida no plano da hélice ou turbina.

K: Componente lateral do empuxo contida no plano da hélice ou turbina.

Mp: Momento de Arfagem gerado pelo Sistema de Propulsão.

Mr: Momento de Rolagem gerado pelo Sistema de Propulsão.

My: Momento de Guinada gerado pelo Sistema de Propulsão.

l: Distância do ponto de aplicação da resultante do empuxo ao CG.

γ : Ângulo polar da distância l medido do nariz para a cauda.

β : Ângulo de inclinação da resultante em relação ao eixo de aplicação. Este ângulo leva ao desalinhamento da linha de ação das forças de propulsão do centro de massa e implica em braços de força e por fim em momentos de natureza não giroscópica.

4.5.4. Sinais

Coeficientes de momento são considerados positivos, se para o piloto da aeronave, elevam o nariz, rolam para a esquerda e desviam o nariz para a esquerda.

Coeficiente de sustentação é positivo se implica em uma força no sentido crescente do eixo coordenado correspondente em K''.

E por fim, arrastos, D e B, são positivos se aplicados no sentido decrescente do eixo coordenado correspondente em K''.

4.6. Sistema Completo de Esforços

Seguindo a convenção escrevem-se os vetores de força e momentos necessários para o TMB e TMA.

Para que se evitem senos e cossenos de χ nas projeções das forças de sustentação, é conveniente, apenas neste caso em que a aceleração ϕ ponto é nula e ψ ponto é conhecida, utilizar a base K'' (a mesma que apresenta o horizonte inclinado) no equacionamento do TMB.

$$(4.22) \quad \begin{pmatrix} F_t \\ F_s \\ F_p \end{pmatrix}_{K''} = \begin{pmatrix} T \cos(\theta + \beta_t + \gamma_t) - N \sin(\theta + \beta_t + \gamma_t) - D_w - D_f - D_{ht} - D_{vt} - W \sin \varphi \\ L_{vt} + K - B_w - B_f - B_{ht} - B_{vt} + W \sin \chi \cos \varphi \\ L_w + L_f + L_{ht} - B_{vt} + T \sin(\theta + \beta_t + \gamma_t) + N \cos(\theta + \beta_t + \gamma_t) - W \cos \chi \cos \varphi \end{pmatrix}_{K''}$$

E para o vetor de momentos:

$$(4.23) \quad \begin{pmatrix} M_{CGr} \\ M_{CGp} \\ M_{CGy} \end{pmatrix}_{K''} = \begin{pmatrix} \begin{pmatrix} -Mr_w - Mr_f - Mr_{ht} - Mr_{vt} + Mr_t + l_w \sin \gamma_w B_w + l_f \sin \gamma_f B_f + \\ + l_{ht} \sin \gamma_{ht} B_{ht} - l_{vt} \sin \gamma_{vt} L_{vt} - l_t \sin \gamma_t K \end{pmatrix} \\ \begin{pmatrix} l_w \cos(\theta - \gamma_w) L_w + l_w \sin(\theta - \gamma_w) D_w + l_f \cos(\theta - \gamma_f) L_f + l_f \sin(\theta - \gamma_f) D_f + \\ + l_{ht} \cos(\theta - \gamma_{ht}) L_{ht} + l_{ht} \sin(\theta - \gamma_{ht}) D_{ht} + l_{vt} \sin(\theta - \gamma_{vt}) D_{vt} - l_{vt} \cos(\theta - \gamma_{vt}) B_{vt} + \\ - l_t \sin \beta_t T - l_t \cos \beta_t N - Mp_w - Mp_f - Mp_{ht} - My_{vt} + Mp_t \end{pmatrix} \\ \begin{pmatrix} My_w + My_f + My_{ht} + Mp_{vt} + My_t + l_w \cos \gamma_w B_w + l_f \cos \gamma_f B_f + \\ + l_{ht} \cos \gamma_{ht} B_{ht} - l_{vt} \cos \gamma_{vt} L_{vt} + l_t \cos \gamma_t K \end{pmatrix} \end{pmatrix}_{K''}$$

Apesar de serem muitos os termos, cada um é de fácil interpretação e referem-se explicitamente às superfícies que os geram. Nos próximos capítulos, muitos deles irão mostrar-se nulos ou negligenciáveis por questões de simetria, considerações de ordens de grandeza e análise de sensibilidade do modelo.

5. Análise e Simplificações dos Esforços

Neste capítulo pretende-se analisar as importâncias das forças aerodinâmicas introduzidas no modelo, de forma a avaliar possíveis simplificações oriundas de simetria.

Há acoplamentos característicos do fenômeno, por exemplo arrasto induzido por sustentação e relações de *roll* e *yaw*, cuja eliminação privaria o modelo de seu potencial de fiel descrição do problema. Por outro lado, é interessante a simplificação de outros acoplamentos, como por exemplo em se tratando do raio de curva, pois apenas assim torna-se possível o isolamento algébrico de certas variáveis. Uma avaliação cuidadosa é portanto necessária para que se encontre o comprometimento adequado entre o preciosismo e algum resultado prático.

O objetivo não deixa de ser manter o maior número possível de termos e evitar linearizações, mas deve-se compreender que o custo de uma solução analítica surge geralmente em detrimento de termos não-lineares. Como discutido abaixo, a arrasto induzido será um foco de dificuldades devido a sua natureza quadrática e portanto principal candidato a simplificações.

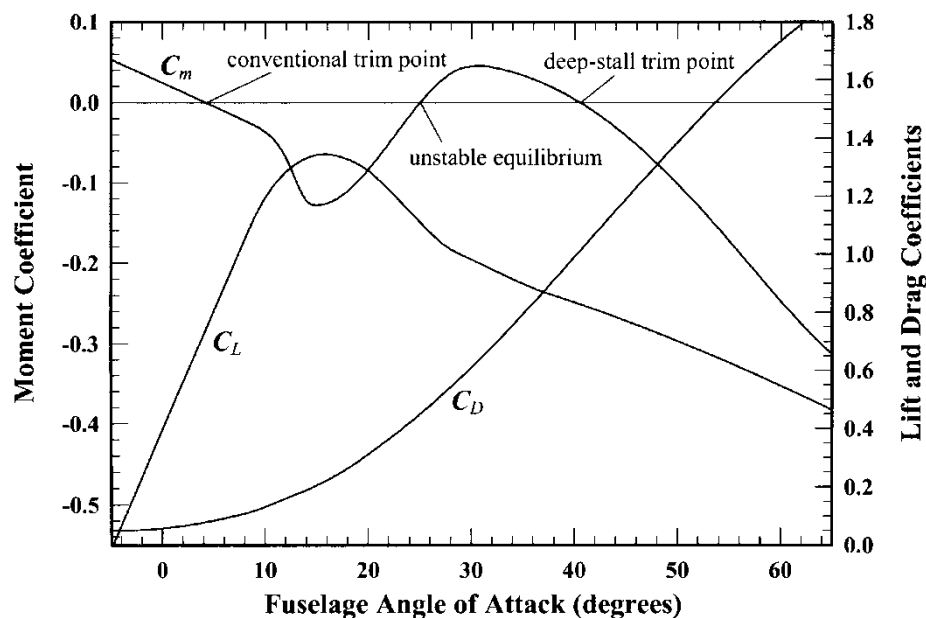


Figura 5.1 - Coeficientes: C_L , C_D e C_m

5.2. Asa

$L_w = \frac{1}{2} \rho V^2 S_w CL_w$: É a parcela mais importante na oposição da força peso e é gerada

pela superfície de maior área.

$D_w = \frac{1}{2} \rho V^2 S_w (CD_{0w} + CD_{iw} + CD_{iw\delta} \delta_w)$: É a parcela em que o termo de arrasto

induzido não deve ser desconsiderado. É interessante que se mantenha a dependência quadrática nesta parcela pois é por ela que se espera introduzir a característica de variação não linear do arrasto. Além disso, essa inclusão não implica em dificuldades algébricas pois o argumento quadrático é um parâmetro e não uma variável.

A último termo computa a parcela de arrasto gerado pela atuação dos ailerons. Um projetista experiente não terá dificuldade em incluí-lo desde que se atente a possíveis inversões de sinal indesejadas. Contudo, trata-se de um arrasto secundário e uma série de termos extras surgem devido a sua inclusão. Apesar disso, ele será por hora mantido.

$$(5.1) \quad \begin{cases} CL_w = CL_{w\alpha} (\theta - \theta_0 - \alpha_{L0w}) \\ CD_w = CD_{0w} + \frac{CL_w^2}{\pi \lambda_w e_w} + CD_{iw\delta} \delta_w \end{cases}$$

$B_w = 0$: A simetria da asa induz o negligenciamento deste termo de arrasto lateral, apesar de se tratar de um escoamento com perfil de velocidades levemente distorcido. O projetista pode incluir este termo em futuras aplicações desta técnica, desde que tome o cuidado de não introduzir acoplamentos não previstos pelo método de resolução aqui apresentado.

$Mp_w = \frac{1}{2} \rho V^2 S_w c_w CMp_w$: Em geral, asas apresentam perfis cambados e portanto, um

momento de arfagem. A determinação correta do centro aerodinâmico da asa deve permitir a consideração do coeficiente como constante.

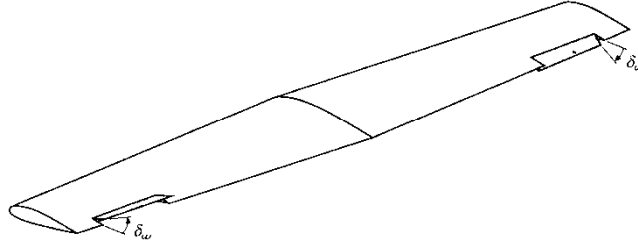


Figura 5.2 - Ailerons

$$Mr_w = \frac{1}{2} \rho V^2 S_w c_w CM_{r_{w\delta}} \delta_w : \text{ Neste esforço a entrada de controle dos ailerons implica em}$$

um momento de rolagem. A natureza linear é imprescindível para aplicabilidade do método, mas não se distancia da natureza real do esforço que já é observado como inerentemente proporcional à atuação [8].

$$My_w = \frac{1}{2} \rho V^2 S_w c_w CM_{y_{w\delta}} \delta_w : \text{ O acoplamento row-yaw é uma característica notável [8]}$$

da atuação de ailerons e é, portanto, incluído.

5.3. Profundor

$$L_{ht} = \frac{1}{2} \rho V^2 S_{ht} CL_{ht} : \text{ A sustentação gerada no profundor apresenta-se normalmente}$$

uma ordem de grandeza abaixo da sustentação da asa. Esta força é a principal responsável pelo equilíbrio de momento de pitch na aeronave, e deve ser controlada pela atuação da superfície móvel. A sustentação deve sofrer alterações em função da rotação de corpo rígido de toda a aeronave e a consequente mudança de ângulo de ataque, e outra parcela deve ser determinada pela entrada de controle de forma a garantir a sustentação desejada.

$$D_{ht} = \frac{1}{2} \rho V^2 S_{ht} (CD_{0ht} + CD_{iht} + CD_{iht\delta} \delta_{ht}) : \text{ O custo de sustentação é arrasto. Uma vez}$$

que a sustentação do profundor implica no equilíbrio de momentos, é natural atribuir um custo a esse equilíbrio. O arrasto da empenagem fica portanto dividido em três contribuintes. O arrasto de forma constante, o arrasto induzido pela sustentação caso não haja atuação do controle e por fim a influência da atuação da superfície móvel. A linearização desse último termo deve ser feita de forma cuidadosa pois ela restringe o intervalo de atuações, uma vez que o arrasto nunca pode sofrer inversões de sinal, mesmo com a inversão da direção da sustentação. Uma simples verificação posterior a obtenção da solução verifica o possível erro.

$$(5.2) \quad \begin{cases} CL_{ht} = CL_{ht\alpha} (\theta - \theta_0 - \alpha_{L0ht}) + CL_{ht\delta} \delta_{ht} \\ CD_{ht} = CD_{0ht} + \frac{CL_{ht\alpha}^2 (\theta - \theta_0 - \alpha_{L0ht})^2}{\pi \lambda_{ht} e_{ht}} + \frac{2CL_{ht\alpha} CL_{ht\delta} (\theta - \theta_0 - \alpha_{L0ht})}{\pi \lambda_{ht} e_{ht}} \delta_w \end{cases}$$

$B_{ht} = 0$: Da mesma forma como para a asa e pelos mesmos motivos, esta componente será considerada nula.

$MP_{ht} = \frac{1}{2} \rho V^2 S_{ht} c_{ht} (CMP_{0ht} + CMP_{ht\delta} \delta_{ht})$: Genericamente o perfil do profundor pode apresentar câmbier, e dessa maneira momento de arfagem. A atuação da superfície móvel altera o câmbier e conseqüentemente implica numa alteração momento [4].

$Mr_{ht} = 0$ e $My_{ht} = 0$: Uma vez que a superfície e a atuação são simétricas, e o escoamento é assumido praticamente simétrico sobre o profundor, não há motivos para que não se considerem os esforços nulos.

5.4. Fuselagem

$L_f = \frac{1}{2} \rho V^2 S_f CL_f$: Este esforço classificado como sustentação é simplesmente a projeção da resultante aerodinâmica na direção das sustentações. A consideração de uma dependência desta força com o ângulo de ataque [5] permite um refinamento do modelo sem a introdução de dificuldades, pois o argumento é parametrizado por θ , que é livre. Por simplicidade, aqui a dependência será considerada linear.

$D_f = \frac{1}{2} \rho V^2 S_f CD_f$: Da mesma forma, é a projeção da resultante na direção do escoamento.

$$(5.3) \quad \begin{cases} CL_f = CL_{f\alpha} (\theta - \theta_0 - \alpha_{L0f}) \\ CD_f = CD_{f\alpha} (\theta - \theta_0 - \alpha_{D0f}) \end{cases}$$

$B_f = 0$: A consideração de escoamento praticamente alinhado e de fuselagem simétrica permite a anulação deste termo.

$Mp_f = \frac{1}{2} \rho V^2 S_f c_f CMp_f$: Genericamente, o centro de pressões da fuselagem pode variar de posição, logo inclui-se este coeficiente para representação dessa propriedade.

$Mr_f = 0$ e $My_f = 0$: A consideração de escoamento praticamente alinhado e de fuselagem simétrica permite a anulação deste termo.

5.5. Leme

$L_{vt} = \frac{1}{2} \rho V^2 S_{vt} CL_{vt}$: A força de sustentação gerada pelo leme deve apresentar-se principalmente ortogonal à força de sustentação da asa e à direção do escoamento, para que sirva de fonte independente de momento de guinada. A função desta força é apenas o equilíbrio de momentos e deve ser completamente controlada pela atuação de comando.

$D_{vt} = \frac{1}{2} \rho V^2 S_{vt} CD_{vt}$: Este arrasto é potencialmente o menor de todos os arrastos abordados. É conveniente que se inclua apenas a parcela da arrasto de forma nesta superfície. Da mesma forma como justificado para o arrasto de atuação de ailerons na asa, uma parcela de arrasto induzido poderia ser incluída sem comprometimento de aplicabilidade do método. Por simplicidade e conveniência, mantem-se apenas o termo constante. Fica a critério do projetista a sofisticação deste esforço em uma modelagem futura.

$$(5.4) \quad \begin{cases} CL_{vt} = CL_{vt\delta} \delta_{vt} \\ CD_{vt} = CD_{0vt} \end{cases}$$

$B_{vt} = 0$, $Mp_{vt} = 0$, $Mr_{vt} = 0$ e $My_{vt} = 0$: A consideração de escoamento praticamente alinhado e simétrica permite a anulação deste termo.

5.6. Propulsão

$T(v) = a_2 v^2 + a_1 v + a_0$: A força gerada axialmente deve apresentar uma interpolação preferencialmente quadrática para que não se introduzam termos de maior potência do que aqueles obrigatoriamente já existentes. Mantido o “acelerador” do motor ou turnina constante, o empuxo gerado passa a depender apenas da velocidade com que o escoamento é admitido no sistema de propulsão. Seria possível modular a injeção de combustível ou o passo da hélice para que a força permanecesse constante. O direcionamento pelo qual aqui se opta, é manter o sistema de propulsão sempre em carga máxima e permitir com que a força varie.

$v = V \cos(\theta + \gamma_t)$: Essa é a componente normal ao plano da hélice da velocidade e deve ser o argumento da interpolação de empuxo axial.

$N \cong 0$ e $K \cong 0$: Uma vez que o fluxo de ar não incide de maneira completamente ortogonal ao plano da hélice componentes laterais podem surgir. Por hora, é desinteressante modelá-las, e devido a sua menor importância serão admitidas nulas. Caso o projetista deseje incluí-las, isso não implicará em dificuldades para método de resolução proposto e referências na literatura estão a disposição [8] e [2].

$\vec{M}_t$: Fazendo o ângulo β_t nulo, ou seja alinhando a linha de ação das forças de propulsão como centro de massa, restam apenas os momentos giroscópicos do rotor ou hélice e o torque reativo a serem modelados.

É importante distinguir qual rotação implica em momento angular e qual o arrasta. Como a aeronave está parada na base K''' , a rotação de K''' é a rotação de arrastamento.

$$(5.5) \quad (\vec{\Omega})_{K'''} = \dot{\psi} \begin{pmatrix} \sin \varphi \cos \theta + \cos \varphi \cos \chi \sin \theta \\ -\cos \varphi \sin \chi \\ -\sin \varphi \sin \theta + \cos \varphi \cos \theta \cos \chi \end{pmatrix}_{K'''}$$

Por outro lado, a rotação a ser utilizada para o cálculo do momento angular é a rotação total, ou seja a soma da rotação própria do observada em K''' , $\dot{\eta}$, e da rotação promovida por $\dot{\psi}$.

Admitindo que a rotação própria é significativamente superior à rotação proveniente da trajetória curvilínea da aeronave, pode-se admitir que o momento angular do sistema propulsor é dado apenas pela sua própria rotação:

$$(5.6) \quad (\vec{H}_{CGt})_{K''} \cong \dot{\eta} \begin{pmatrix} J_{tt} \cos \gamma_t \\ 0 \\ J_{tt} \sin \gamma_t \end{pmatrix}_{K''}$$

Note-se que os termos em $\dot{\psi}$ são desprezados pois, em geral, $\dot{\eta}$ é superior em três ordens de grandeza. Da mesma forma como feito na aplicação do TMA no capítulo anterior, o produto vetorial da rotação de arrastamento do sistema pelo momento angular é o único termo a fornecer componentes não nulas.

$$(5.7) \quad (\vec{M}_t)_{K''} = \begin{pmatrix} Mr_t \\ Mp_t \\ My_t \end{pmatrix}_{K''} = -(\vec{\Omega} \times \vec{H}_{CGt})_{K''} + R_t \begin{pmatrix} \cos \gamma_t \\ 0 \\ \sin \gamma_t \end{pmatrix}_{K''} =$$

$$= J_{tt} \dot{\eta} \dot{\psi} \begin{pmatrix} \sin \gamma_t \cos \varphi \sin \chi \\ \cos \gamma_t (\sin \varphi \sin \theta - \cos \varphi \cos \theta \cos \chi) + \sin \gamma_t (\sin \varphi \cos \theta + \cos \varphi \cos \chi \sin \theta) \\ - \cos \gamma_t \cos \varphi \sin \chi \end{pmatrix}_{K''} + R_t \begin{pmatrix} \cos \gamma_t \\ 0 \\ \sin \gamma_t \end{pmatrix}_{K''}$$

É importante lembrar que o produto vetorial fornece os momentos aplicados sobre o sistema de propulsão para que ele faça a curva. Portanto, por ação e reação, o sinal de menos adéqua a cálculo aos momentos giroscópicos, que juntamente com o torque reativo no eixo compõem os momentos que a aeronave sofre.

5.7. Resumo

Eliminando os termos nulos, conforme discutido acima o TMB e TMA passam a apresentar a seguinte forma:

$$(5.8) \quad \begin{pmatrix} 0 \\ \frac{mV^2 \cos^2 \varphi \cos \chi}{R} \\ \frac{mV^2 \cos^2 \varphi \sin \chi}{R} \end{pmatrix}_{K''} = \begin{pmatrix} T \cos(\theta + \gamma_t) - D_w - D_f - D_{ht} - D_{vt} - W \sin \varphi \\ L_{vt} + W \sin \chi \cos \varphi \\ L_w + L_f + L_{ht} + T \sin(\theta + \gamma_t) - W \cos \chi \cos \varphi \end{pmatrix}_{K''}$$

$$(5.9) \quad \frac{V^2 \cos^4 \varphi}{R^2} \begin{pmatrix} (J_{zz} - J_{yy})(\tan \varphi \sin \theta - \cos \theta \cos \chi) \sin \chi \\ (J_{xx} - J_{zz})(1/2 \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi) \\ (J_{yy} - J_{xx})(-\tan \varphi \cos \theta - \sin \theta \cos \chi) \sin \chi \end{pmatrix}_{K''} =$$

$$= \begin{pmatrix} -Mr_w - l_{vt} \sin \gamma_{vt} L_{vt} \\ (l_w \cos(\theta - \gamma_w) L_w + l_w \sin(\theta - \gamma_w) D_w + l_f \cos(\theta - \gamma_f) L_f + l_f \sin(\theta - \gamma_f) D_f + l_{ht} \cos(\theta - \gamma_{ht}) L_{ht} + \\ + l_{ht} \sin(\theta - \gamma_{ht}) D_{ht} + l_{vt} \sin(\theta - \gamma_{vt}) D_{vt} - Mp_w - Mp_f - Mp_{ht} \\ My_w - l_{vt} \cos \gamma_{vt} L_{vt} \end{pmatrix}_{K''}$$

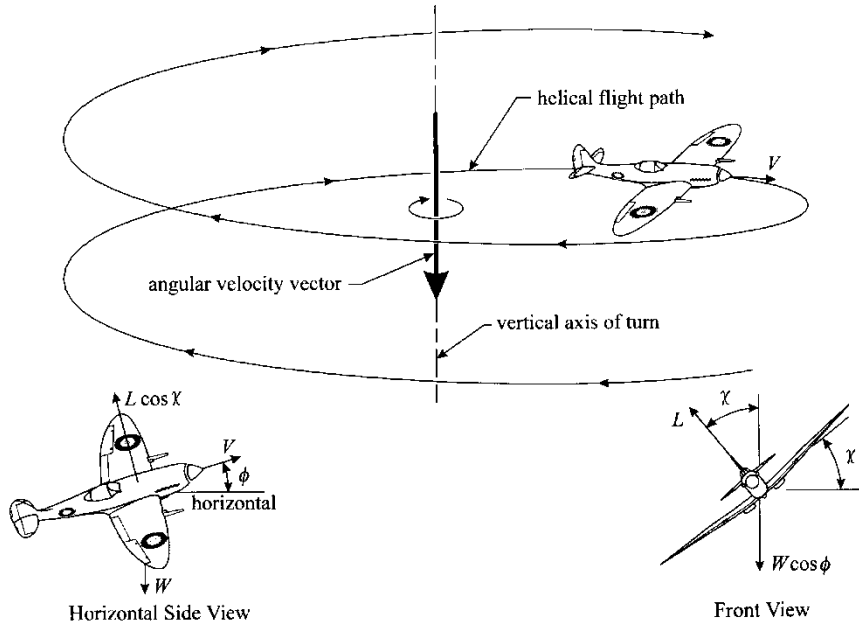


Figura 5.3 - Resumo: Trajetória helicoidal

6. Equilíbrio Tridimensional de Momentos

As considerações do capítulo anterior permitiram simplificações no sistema suficientes para possibilitar a expressão analítica de forma independente dos comandos necessários para que haja equilíbrio de momentos em um dado ponto de operação desejado.

Analisando a primeira e terceira equação do TMA, é possível perceber que a combinação linear das duas implica no desacoplamento dos termos de atuação de aileron dos termos de atuação de leme.

$$(6.1) \quad \frac{V^2 \cos^4 \varphi}{R^2} \begin{pmatrix} (J_{zz} - J_{yy}) (\tan \varphi \sin \theta - \cos \theta \cos \chi) \sin \chi \\ (J_{xx} - J_{zz}) (1/2 \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi) \\ (J_{yy} - J_{xx}) (-\tan \varphi \cos \theta - \sin \theta \cos \chi) \sin \chi \end{pmatrix}_{K''} =$$

$$= \begin{pmatrix} -Mr_w - l_{vt} \sin \gamma_{vt} L_{vt} \\ l_w \cos(\theta - \gamma_w) L_w + l_w \sin(\theta - \gamma_w) D_w + l_f \cos(\theta - \gamma_f) L_f + l_f \sin(\theta - \gamma_f) D_f + l_{ht} \cos(\theta - \gamma_{ht}) L_{ht} + \\ + l_{ht} \sin(\theta - \gamma_{ht}) D_{ht} + l_{vt} \sin(\theta - \gamma_{vt}) D_{vt} - Mp_w - Mp_f - Mp_{ht} \\ My_w - l_{vt} \cos \gamma_{vt} L_{vt} \end{pmatrix}_{K''}$$

Para facilitar a compreensão dos passos da resolução a seguir apenas as equações para os momentos de roll e yaw foram combinadas linearmente pelo seno e cosseno de γ_{vt} e copiadas abaixo.

$$(6.2) \quad \begin{cases} \frac{V^2 \cos^4 \varphi \sin \chi}{R^2} [\cos \gamma_{vt} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) + \sin \gamma_{vt} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)] = \\ = \cos \gamma_{vt} Mr_w + \sin \gamma_{vt} My_w \\ \frac{V^2 \cos^4 \varphi \sin \chi}{R^2} [\sin \gamma_{vt} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - \cos \gamma_{vt} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)] = \\ = \sin \gamma_{vt} Mr_w - \cos \gamma_{vt} My_w + l_{vt} L_{vt} \end{cases}$$

Na primeira equação o segundo membro apresenta termos que dependem apenas da atuação dos ailerons da asa. Na segunda, surge também a atuação do leme. Os termos forçantes são expandidos.

$$(6.3) \quad \begin{cases} \cos \gamma_{vt} Mr_w + \sin \gamma_{vt} My_w = \frac{1}{2} \rho V^2 S_w c_w (\cos \gamma_{vt} CM_{r_{w\delta}} + \sin \gamma_{vt} CM_{y_{w\delta}}) \delta_w \\ \sin \gamma_{vt} Mr_w - \cos \gamma_{vt} My_w + l_{vt} L_{vt} = \frac{1}{2} \rho V^2 S_w c_w (\sin \gamma_{vt} CM_{r_{w\delta}} - \cos \gamma_{vt} CM_{y_{w\delta}}) \delta_w + \frac{1}{2} \rho V^2 S_{vt} l_{vt} CL_{vt\delta} \delta_{vt} \end{cases}$$

Era natural esperar a ligação entre as atuações, pois a acoplamento row-yaw é uma característica propositalmente incluída no modelo.

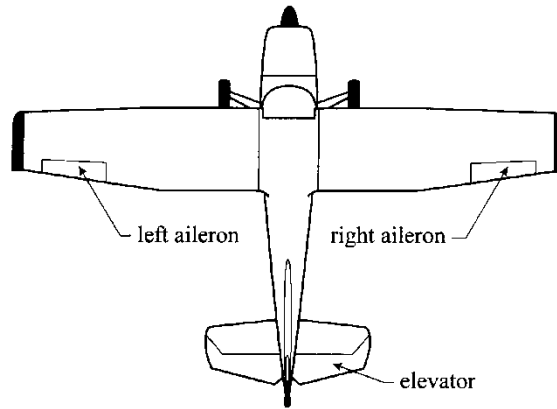


Figura 6.2 - Vista superior: Ailerons e Profundor

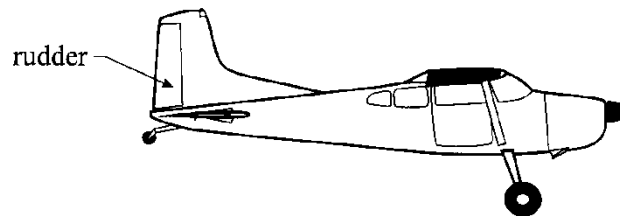


Figura 6.3 - Vista lateral: Leme

Isolando a atuação dos ailerons da primeira equação e substituindo na segunda, resolve-se o sistema para ambos os comandos.

$$(6.4) \quad \begin{cases} \delta_w = \frac{2 \cos^4 \varphi \sin \chi [\cos \gamma_{vt} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) + \sin \gamma_{vt} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{\rho R^2 S_w c_w (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CM y_{w\delta})} \\ \delta_{vt} = \frac{2 \cos^4 \varphi \sin \chi [CM y_{w\delta} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - CMr_{w\delta} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{\rho R^2 S_{vt} l_{vt} CL_{vt\delta} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CM y_{w\delta})} \end{cases}$$

Note-se que as atuações acima para o regime permanente de vôo surgem como consequência do fato de que a rotação da aeronave em curva não coincide com os eixos centrais de inércia e portanto produz momentos. Para o vôo retilíneo χ é nulo e o raio de curva é infinito, o que implica automaticamente em comandos nulos.

Ainda é necessário o equilíbrio de pitch. Nessa direção há significativamente mais fontes de momentos que devem ser balanceados pela atuação do profundor. Apenas a equação referente a essa condição apresenta-se abaixo.

$$(6.5) \quad \left\{ \frac{V^2 \cos^4 \varphi (J_{xx} - J_{zz}) [1/2 \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi]}{R^2} = l_w \cos(\theta - \gamma_w) L_w + l_w \sin(\theta - \gamma_w) D_w + \right. \\ \left. + l_f \cos(\theta - \gamma_f) L_f + l_f \sin(\theta - \gamma_f) D_f + l_{ht} \cos(\theta - \gamma_{ht}) L_{ht} + l_{ht} \sin(\theta - \gamma_{ht}) D_{ht} + l_{vt} \sin(\theta - \gamma_{vt}) D_{vt} - Mp_w - Mp_f - Mp_{ht} \right.$$

Ao se substituírem as expressões para todos os esforços na equação acima, obtém-se uma igualdade em que a deflexão do profundor pode ser facilmente isolada.

Neste momento, a exclusão de termos de arrasto gerado pela atuação do leme mostra-se estratégica. Como sugerido no capítulo anterior, o projetista pode incluir tal influência sem que a natureza da equação abaixo se altere. No entanto, ela passa a adquirir uma formulação ainda mais extensa, não conveniente, em troca de um ganho secundário em precisão.

Arranjando os termos, pode-se reconhecer cada superfície e suas contribuições.

$$(6.6) \quad \left\{ \frac{2 \cos^4 \varphi (J_{xx} - J_{zz}) [1/2 \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi]}{\rho R^2} = \right. \\ = S_w [l_w CL_w \cos(\theta - \gamma_w) + l_w (CD_{0w} + CD_{iw} + CD_{iw\delta} \delta_w) \sin(\theta - \gamma_w) - c_w CMp_w] + \\ + S_f [l_f CL_f \cos(\theta - \gamma_f) + l_f CD_f \sin(\theta - \gamma_f) - c_f CMp_f] + \\ + S_{vt} l_{vt} CD_{0vt} \sin(\theta - \gamma_{vt}) + \\ + S_{ht} [l_{ht} CL_{0ht} \cos(\theta - \gamma_{ht}) + l_{ht} (CD_{0ht} + CD_{iht}) \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{0ht}] + \\ \left. + S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{iht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{ht\delta}] \delta_{ht} \right.$$

Ao isolar o comando do profundor percebe-se que o mesmo é fruto da contribuição de dois termos. Um oriundo do estado de vôo em curva e outro semelhante ao equilíbrio bidimensional.

$$(6.7) \quad \left\{ \delta_{ht} = \frac{2 \cos^4 \varphi (J_{xx} - J_{zz}) [1/2 \sin 2\theta (\cos^2 \chi - \tan^2 \varphi) + \cos 2\theta \cos \chi \tan \varphi]}{\rho R^2 S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{iht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{ht\delta}]} + \right. \\ \left. - \frac{S_w l_w CD_{iw\delta} \sin(\theta - \gamma_w) \delta_w}{S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{iht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{ht\delta}]} + \Phi_{ht} \right. \\ \left. \Phi_{ht} = \frac{-1}{S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{iht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{ht\delta}]} \cdot \right. \\ \left. \left\{ S_w [l_w CL_w \cos(\theta - \gamma_w) + l_w (CD_{0w} + CD_{iw}) \sin(\theta - \gamma_w) - c_w CMp_w] + \right. \right. \\ \left. + S_f [l_f CL_f \cos(\theta - \gamma_f) + l_f CD_f \sin(\theta - \gamma_f) - c_f CMp_f] + S_{vt} l_{vt} CD_{0vt} \sin(\theta - \gamma_{vt}) + \right. \\ \left. + S_{ht} [l_{ht} CL_{0ht} \cos(\theta - \gamma_{ht}) + l_{ht} (CD_{0ht} + CD_{iht}) \sin(\theta - \gamma_{ht}) - c_{ht} CMp_{0ht}] \right\} \right.$$

Neste ponto, estão determinados os comandos necessários das superfícies de controle para que o equilíbrio ocorra em um dado ponto R, φ, χ, θ desejado. A velocidade não influi sobre os comandos pois todos os termos fonte de momento são igualmente proporcionais à velocidade.

Caso houvesse, por exemplo, desalinhamento da resultante do sistema de propulsão em relação ao centro de massa, outros termos fonte de momento surgiriam e a independência da velocidade não seria genericamente mantida. Isso não é uma implicação crítica e o projetista experiente não terá dificuldade em incluí-la.

Diferentemente da abordagem bidimensional do início deste trabalho, a restrição de propulsão alinhada deixou de se estabelecer a partir do momento em que se linearizou a termo de arrasto induzido no profundor. Esse comprometimento é um avanço significativo, pois é um exemplo de caso em que se abre mão de um preciosismo matemático de segunda ordem em prol de uma flexibilidade de projeto com influências de ordem potencialmente superior.

Uma hipótese crucial existente por traz de todo capítulo é a estabilidade. É possível calcular a amplitude de comando que implica em equilíbrio, mas a natureza desse equilíbrio bem como a migração de um ponto para outro é algo que não pode ser analisado pela presente abordagem. Felizmente, o comportamento monotônico dos esforços e a observação do voo de aeronaves existentes consideradas satisfatoriamente estáveis [6] sugerem que a migração de uma condição para outra ocorre de forma suave e que a relação conjunto de comandos e ponto de operação é única.

Quanto ao estudo de estabilidade, deve-se por hora assumir que a escolha de braços é adequada e garante as características restituidoras necessárias para o equilíbrio estável. Análises de desempenho como a aqui desenvolvida dependem de uma análise de estabilidade normalmente realizada anteriormente. Por outro lado, as equações de estabilidade derivam das equações do movimento, por isso optou-se por discutir estabilidade mais a frente e lançar mão da hipótese de uma aeronave a priori estável.

7. Análise de Sensibilidade para o Raio Instantâneo de Curva

Uma vez resolvido o TMA, começa-se a direcionar esforços na resolução do TMB que fornecerá as informações necessárias para que a trajetória seja traçada. Das três equações abaixo devem ser isoladas três variáveis V , R , φ .

$$(7.1) \quad \begin{pmatrix} 0 \\ \frac{mV^2 \cos^2 \varphi \cos \chi}{R} \\ \frac{mV^2 \cos^2 \varphi \sin \chi}{R} \end{pmatrix}_{K''} = \begin{pmatrix} T \cos(\theta + \gamma_t) - D_w - D_f - D_{ht} - D_{vt} - W \sin \varphi \\ L_{vt} + W \sin \chi \cos \varphi \\ L_w + L_f + L_{ht} + T \sin(\theta + \gamma_t) - W \cos \chi \cos \varphi \end{pmatrix}_{K''}$$

Não se pode esquecer que os comandos apresentam dependência da variável R . Logo, o segundo membro apresenta parcelas que impedem o isolamento direto de R , tanto na segunda quanto na terceira equação acima.

A segunda equação sugere uma relação simples entre o raio e as outras variáveis desde que a sustentação do leme possa ser desprezada como força lateral agindo sobre o centro de massa. De fato, em diversas formulações na literatura a força do leme é desconsiderada automaticamente e a relação simplificada é aplicada sem discussões mais profundas.

$$(7.2) \quad \frac{V^2 \cos \varphi}{R} \cong g \tan \chi$$

Com o intuito de verificar a pertinência dessa simplificação e utilizá-la se possível, propõe-se determinar explicitamente o erro introduzido pela adoção da mesma.

Parte-se da equação completa, recordando que a função do leme é o equilíbrio de momentos de yaw. Como calculado no capítulo anterior, já é possível determinar a atuação necessária do leme para o equilíbrio em qualquer ponto de operação desejado.

$$(7.3) \quad \frac{mV^2 \cos^2 \varphi \cos \chi}{R} = L_{vt} + W \sin \chi \cos \varphi = \frac{1}{2} \rho V^2 S_{vt} CL_{vt\delta} \delta_{vt} + W \sin \chi \cos \varphi$$

Substituindo a expressão para o δv_t , obtém-se uma equação de segundo grau para a curvatura.

$$(7.4) \quad \frac{1}{R} - \frac{1}{R^2} \frac{\cos^2 \varphi \tan \chi [CMy_{w\delta} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - CMr_{w\delta} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{ml_{vt} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMy_{w\delta})} = \frac{g \tan \chi}{V^2 \cos \varphi}$$

Rearranjando os termos de forma mais conveniente, pode-se perceber a influência da atuação do leme de forma mais clara.

A constante Λ representa os desvios devidos à atuação do leme e a introdução de uma componente de força na direção lateral.

$$(7.5) \quad \begin{cases} \frac{g \tan \chi}{V^2 \cos \varphi} R^2 - R + \Lambda = 0 \\ \Lambda = \frac{\cos^2 \varphi \tan \chi [CMy_{w\delta} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - CMr_{w\delta} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{ml_{vt} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMy_{w\delta})} \end{cases}$$

Aplicando a fórmula para a solução de uma equação de segundo grau e pondo em evidência a solução aproximada normalmente utilizada, fica nítida a natureza da influência de Λ .

$$(7.6) \quad R = \frac{V^2 \cos \varphi}{g \tan \chi} \left(\frac{1}{2} + \sqrt{\frac{1}{4} - \Lambda \frac{g \tan \chi}{V^2 \cos \varphi}} \right) = \frac{V^2 \cos \varphi}{g \tan \chi} \Delta_R$$

É possível definir um determinante de desvio ΔR .

$$(7.7) \quad \Delta_R = \frac{1}{2} + \sqrt{\frac{1}{4} - \frac{g}{mV^2} \frac{\cos \varphi \tan^2 \chi [CMy_{w\delta} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - CMr_{w\delta} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{l_{vt} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMy_{w\delta})}}$$

Conforme ΔR se desviar de 1, o raio sofrerá alterações em relação àquele previsto pela aproximação. Para quantificar essas alterações, é suficiente plotar a curva de ΔR para uma ampla faixa de χ , fixando as outras variáveis em torno de um caso extremo.

Estimando $J \approx ml_{vt}^2$, sendo l_{vt} a dimensão característica da aeronave que se assemelha à envergadura, mostra-se que para tal dimensão característica da ordem de 10m, velocidade da ordem de 10m/s, coeficientes de momento típicos da ordem de 0,1 e uma faixa de 60° de ângulo χ , ou seja, um caso absolutamente extremo de baixa velocidade e altíssimo momento de inércia, o desvio máximo não passa de 8%.

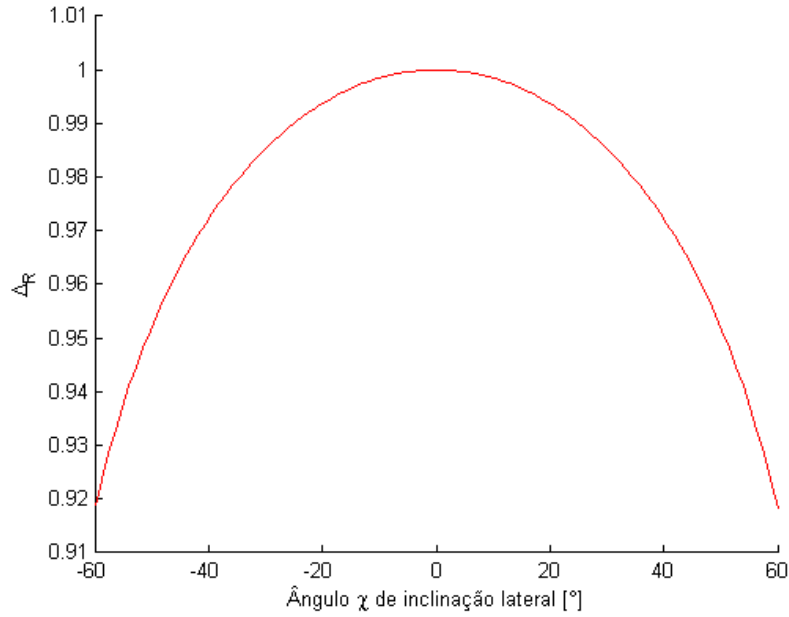


Figura 7.2 - Erro da aproximação

Analogamente uma análise de ordem de grandeza da equação original do TMB já poderia indicar que as influências são fortemente secundárias. Os colchetes representam ordem de grandeza da variável.

Como dito anteriormente a sustentação na asa é a principal força a equilibrar o peso.

$$(7.8) \quad [m] + [g] = \left[\frac{1}{2} \rho \right] + 2[V] + [S_w] + [CL_w]$$

Uma vez que o produto $\frac{1}{2} \rho CL_w$ varia em torno de 1 podendo apresentar-se em muitos casos com coeficientes de sustentação baixos, menores que $\frac{1}{2}$, na pior das hipóteses é admitida ordem de grandeza -1 para o produto. A relação torna-se, portanto:

$$(7.9) \quad [m] - 2[V] + 2 = [S_w]$$

Comumente a área do leme relaciona-se com a área da asa por uma diferença de duas ordens, $[S_w] - 2 = [S_{vt}]$, o que estabelece:

$$(7.10) \quad [m] - 2[V] = [S_{vt}]$$

A partir das relações acima pode-se limitar a força máxima que o leme é capaz de produzir, lembrando que sua geometria não deve permitir coeficientes de sustentação tão altos quanto a asa.

$$(7.11) \quad [L_{vt}] = \left[\frac{1}{2} \rho \right] + 2[V] + [S_{vt}] + [C_{Lvt}] \leq [m] - 1$$

Retrabalhando a equação oriunda do TMB, obtem-se:

$$(7.12) \quad \frac{V^2 \cos^2 \varphi}{R} = \frac{L_{vt} \sec \chi}{m} + g \tan \chi \cos \varphi = \sec \chi \left(\frac{L_{vt}}{m} + g \sin \chi \cos \varphi \right)$$

$$(7.13) \quad 2[V] + 2[\cos \varphi] - [R] = [\sec \chi] + \langle [L_{vt}] - [m]; [g] + [\sin \chi] + [\cos \varphi] \rangle$$

O operador $\langle \ ; \ \rangle$ realiza a soma mantendo apenas o termo de maior ordem de grandeza. O primeiro argumento está limitado superiormente pela ordem -1 enquanto o segundo, inferiormente por 0.

$$(7.14) \quad 2[V] - [R] = [\sec \chi] + \langle \leq -1; 1 + [\sin \chi] \rangle = 1 + [\tan \chi] \quad \Leftrightarrow \quad \frac{V^2 \cos \varphi}{R} \cong g \tan \chi$$

Logo, é possível utilizar a equação aproximada sem a introdução de erros de ordem significativa. Deste ponto em diante o TMB passará a apresentar-se a seguinte forma:

$$(7.15) \quad \left(\begin{array}{c} 0 \\ \frac{V^2 \cos \varphi}{R} \\ \frac{m V^2 \cos^2 \varphi \sin \chi}{R} \end{array} \right)_{K^*} = \left(\begin{array}{c} T \cos(\theta + \gamma_t) - D_w - D_f - D_{ht} - D_{vt} - W \sin \varphi \\ g \tan \chi \\ L_w + L_f + L_{ht} + T \sin(\theta + \gamma_t) - W \cos \chi \cos \varphi \end{array} \right)_{K^*}$$

8. Resolução Semi-Analítica

Todos as considerações feitas até agora tiverem o objetivo de trabalhar o sistema de equações para que a família de trajetórias helicoidais possa ser calculada analiticamente, prezando pela alta fidelidade do modelo ao fenômeno.

Pode-se fazer um pequeno resumo dos resultados obtidos até agora.

O TMB possui a seguinte forma com a introdução da relação discutida no capítulo anterior.

$$(8.1) \quad \begin{cases} 0 = T \cos(\theta + \gamma_t) - D_w - D_f - D_{ht} - D_{vt} - W \sin \varphi \\ 0 = L_w + L_f + L_{ht} + T \sin(\theta + \gamma_t) - W \cos \varphi \sec \chi \\ \frac{1}{R} = \frac{g \tan \chi}{V^2 \cos \varphi} \end{cases}$$

O TMA fornece os comandos necessários para que o equilíbrio ocorra no ponte de operação desejado.

$$(8.2) \quad \begin{cases} \delta_w = \frac{2g^2 \sin^3 \chi \cos^2 \varphi [\cos \gamma_{vt} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) + \sin \gamma_{vt} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{\rho V^4 S_w c_w (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMY_{w\delta}) \cos^2 \chi} \\ \delta_{vt} = \frac{2g^2 \sin^3 \chi \cos^2 \varphi [CMY_{w\delta} (J_{yy} - J_{zz}) (\tan \varphi \sin \theta - \cos \chi \cos \theta) - CMr_{w\delta} (J_{xx} - J_{yy}) (\tan \varphi \cos \theta + \cos \chi \sin \theta)]}{\rho V^4 S_{vt} l_{vt} CL_{vt\delta} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMY_{w\delta}) \cos^2 \chi} \\ \delta_{ht} = \frac{2g^2 \tan^2 \chi (J_{xx} - J_{zz}) [1/2 \sin 2\theta (\cos^2 \chi \cos^2 \varphi - \sin^2 \varphi) + \cos 2\theta \cos \chi \sin \varphi \cos \varphi]}{\rho V^4 S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{ht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} Cmp_{ht\delta}]} + \\ + \frac{S_w l_w CD_{hw\delta} \sin(\theta - \gamma_w) \delta_w}{S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{ht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} Cmp_{ht\delta}]} + \Phi_{ht} \end{cases}$$

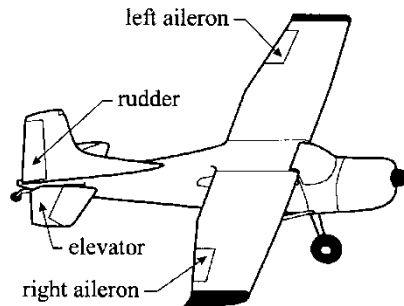


Figura 8.2 - Comandos

Resta apenas a tarefa de resolverem-se os pares V, φ , que devem surgir a partir das duas primeiras equações do TMB acima.

O sistema apresenta apenas potencias inteiras de V , cujos coeficientes dependem de φ . Como os comandos implicam em termos da quarta potência da velocidade não se podem exprimir genericamente as raízes de um polinômio de tal grau. Logo, a solução buscada é analítica para todas as variáveis exceto para a velocidade, que deve provir da simples solução numérica de um polinômio.

$$(8.3) \quad \begin{cases} 0 = T \cos(\theta + \gamma_t) - \frac{1}{2} \rho V^2 [S_w (CD_{0w} + CD_{iw} + CD_{iw\delta} \delta_w) + S_f CD_f + S_{ht} (CD_{0ht} + CD_{iht} + CD_{iht\delta} \delta_{ht}) + S_{vt} CD_{0vt}] - W \sin \varphi \\ 0 = T \sin(\theta + \gamma_t) + \frac{1}{2} \rho V^2 [S_w CL_w + S_f CL_f + S_{ht} (CL_{0ht} + CL_{ht\delta} \delta_{ht})] - W \sec \chi \cos \varphi \end{cases}$$

Duas formas de resolução são propostas.

A primeira parte de duas soluções em paralelo, para um dado φ , tanto o primeiro quanto segundo polinômio devem ser resolvidos independentemente e a raiz V comum aos dois é a solução do sistema. Para isso uma simples rotina lógica pode ser empregada. Contudo, para um desejado ponto de operação deve haver apenas um φ que permita raízes comuns aos dois polinômios. Portanto, esse método trata de uma busca relativamente extensa e não será empregado.

O outro método isola φ de ambos os polinômios e define um novo polinômio de maior grau que não apresenta φ em nenhum de seus coeficientes. Em contrapartida, é necessário que se linearizem os coeficientes dependentes de φ para que o isolamento seja possível. Uma vez que φ é sempre argumento de senos e cossenos, aproximações de pequenos ângulos levam a erros inferiores a 6% desde que os mesmos não excedam 20°. Caso, desejem-se soluções para maiores φ , sugere-se o primeiro método.

Aplicando as relações de pequenos ângulos para os comandos, observa-se que eles são aproximados por duas parcelas, uma dependente de φ e outra não.

$$(8.4) \quad \begin{cases} \delta_w \equiv \frac{\varphi [2g^2 \sin^3 \chi (\sin \theta \cos \gamma_{vt} (J_{yy} - J_{zz}) + \cos \theta \sin \gamma_{vt} (J_{xx} - J_{yy}))] + [2g^2 \sin^3 \chi (-\cos \chi \cos \theta \cos \gamma_{vt} (J_{yy} - J_{zz}) + \cos \chi \sin \theta \sin \gamma_{vt} (J_{xx} - J_{yy}))]}{\rho V^4 S_w c_w (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMY_{w\delta}) \cos^2 \chi} \\ \delta_{vt} \equiv \frac{\varphi [2g^2 \sin^3 \chi (CMY_{w\delta} \sin \theta (J_{yy} - J_{zz}) - CMr_{w\delta} \cos \theta (J_{xx} - J_{yy}))] + [2g^2 \sin^3 \chi (-CMY_{w\delta} \cos \chi \cos \theta (J_{yy} - J_{zz}) - CMr_{w\delta} \cos \chi \sin \theta (J_{xx} - J_{yy}))]}{\rho V^4 S_{vt} l_{vt} CL_{vt\delta} (\cos \gamma_{vt} CMr_{w\delta} + \sin \gamma_{vt} CMY_{w\delta}) \cos^2 \chi} \\ \delta_{ht} \equiv \frac{\varphi [2g^2 \tan \chi \sin \chi \cos 2\theta (J_{xx} - J_{zz})] + [g^2 \sin^2 \chi \sin 2\theta (J_{xx} - J_{zz})]}{\rho V^4 S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{ht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} Cmp_{ht\delta}]} + \frac{S_w J_w CD_{w\delta} \sin(\theta - \gamma_w) \delta_w}{S_{ht} [l_{ht} CL_{ht\delta} \cos(\theta - \gamma_{ht}) + l_{ht} CD_{ht\delta} \sin(\theta - \gamma_{ht}) - c_{ht} Cmp_{ht\delta}]} + \Phi_{ht} \end{cases}$$

$$(8.5) \quad \begin{cases} \delta_w \equiv \frac{\varphi N_{w\varphi} + N_w}{\rho V^4 S_w R_w} \\ \delta_{vt} \equiv \frac{\varphi N_{vt\varphi} + N_{vt}}{\rho V^4 S_{vt} R_{vt}} \\ \delta_{ht} \equiv \frac{\varphi N_{ht\varphi} + N_{ht}}{\rho V^4 S_{ht} R_{ht}} + \Phi_{ht} \end{cases}$$

Ao se substituírem os comandos no par de equações e por meio do isolamento de φ em ambas, o sistema resume-se a:

$$(8.6) \quad \left\{ \begin{array}{l} \varphi = \frac{V^2 T \cos(\theta + \gamma_t) - \frac{1}{2} \rho V^4 \Sigma_D - \frac{CD_{iht\delta} N_{ht}}{2R_{ht}} - \frac{CD_{iw\delta} N_w}{2R_w}}{\left(V^2 W + \frac{CD_{iht\delta} N_{ht\varphi}}{2R_{ht}} + \frac{CD_{iw\delta} N_{w\varphi}}{2R_w} \right)} \\ \varphi = - \frac{V^2 T \sin(\theta + \gamma_t) + \frac{1}{2} \rho V^4 \Sigma_L - \sec \chi V^2 W + \frac{CL_{ht\delta} N_{ht}}{2R_{ht}}}{\left(\frac{CL_{ht\delta} N_{ht\varphi}}{2R_{ht}} \right)} \\ \Sigma_L = [S_w CL_w + S_f CL_f + S_{ht} (CL_{0ht} + CL_{ht\delta} \Phi_{ht})] \\ \Sigma_D = [S_w (CD_{0w} + CD_{iw}) + S_f CD_f + S_{ht} (CD_{0ht} + CD_{iht} + CD_{iht\delta} \Phi_{ht}) + S_{vt} CD_{0vt}] \end{array} \right.$$

Por fim igualam-se as duas primeiras expressões e um polinômio em V , de coeficientes parametrizados por θ e χ apenas, é obtido.

$$(8.7) \quad \left\{ \begin{array}{l} 0 = \left(V^2 W + \frac{CD_{iht\delta} N_{ht\varphi}}{2R_{ht}} + \frac{CD_{iw\delta} N_{w\varphi}}{2R_w} \right) \left[V^2 T \sin(\theta + \gamma_t) + \frac{1}{2} \rho V^4 \Sigma_L - \sec \chi V^2 W + \frac{CL_{ht\delta} N_{ht}}{2R_{ht}} \right] + \\ + \left(\frac{CL_{ht\delta} N_{ht\varphi}}{2R_{ht}} \right) \left[V^2 T \cos(\theta + \gamma_t) - \frac{1}{2} \rho V^4 \Sigma_D - \frac{CD_{iht\delta} N_{ht}}{2R_{ht}} - \frac{CD_{iw\delta} N_w}{2R_w} \right] \end{array} \right.$$

Expandindo todos os termos e inserindo a curva de propulsão como modelada anteriormente, definem-se os coeficientes:

$$(8.8) \quad \left\{ \begin{array}{l} V^6 : Wa_2 \sin(\theta + \gamma_t) \cos^2(\theta + \gamma_t) + \frac{1}{2} \rho W \Sigma_L \\ V^5 : Wa_1 \sin(\theta + \gamma_t) \cos(\theta + \gamma_t) \\ V^4 : Wa_0 \sin(\theta + \gamma_t) - W^2 \sec \chi + \left(\frac{N_{ht\varphi}}{2R_{ht}} \right) \left[\frac{1}{2} \rho (CD_{iht\delta} \Sigma_L - CD_{ht\delta} \Sigma_D) + a_2 \cos^2(\theta + \gamma_t) (CD_{ht\delta} \cos(\theta + \gamma_t) + CD_{iht\delta} \sin(\theta + \gamma_t)) \right] \\ + \left(\frac{CD_{iw\delta} N_{w\varphi}}{2R_w} \right) \left[\frac{1}{2} \rho \Sigma_L + a_2 \sin(\theta + \gamma_t) \cos^2(\theta + \gamma_t) \right] \\ V^3 : \left(\frac{N_{ht\varphi}}{2R_{ht}} \right) [a_1 \cos(\theta + \gamma_t) (CL_{ht\delta} \cos(\theta + \gamma_t) + CD_{iht\delta} \sin(\theta + \gamma_t))] + \left(\frac{CD_{iw\delta} N_{w\varphi}}{2R_w} \right) [a_1 \sin(\theta + \gamma_t) \cos(\theta + \gamma_t)] \\ V^2 : \left(\frac{CD_{iht\delta} N_{ht\varphi}}{2R_{ht}} + \frac{CD_{iw\delta} N_{w\varphi}}{2R_w} \right) [a_0 \sin(\theta + \gamma_t) - W \sec \chi] + W \left(\frac{CL_{ht\delta} N_{ht}}{2R_{ht}} \right) + \left(\frac{CL_{ht\delta} N_{ht\varphi}}{2R_{ht}} \right) a_0 \cos(\theta + \gamma_t) \\ V : 0 \\ 1 : CD_{iw\delta} CL_{ht\delta} \left(\frac{N_{w\varphi} N_{ht} - N_w N_{ht\varphi}}{4R_w R_{ht}} \right) \end{array} \right.$$

É notável que para o caso de vôo retilíneo o grau do polinômio se rebaixa e a solução bidimensional é resgatada.

9. Aplicação do Método

A aeronave a ser utilizada na exemplificação do método, foi projetada pela equipe Keep Flying da Escola Politécnica da USP, para participação do SEA Aerodesign East 2009. Uma vez que o formando participou do desenvolvimento do aeromodelo, a escolha do exemplo deu-se devido à grande disponibilidade de dados. A planta resumida do aeromodelo apresentam-se no ANEXO II.

9.1. Velocidade

Era esperado o acréscimo de velocidade para pequenos ângulos θ , situação de mergulho, em oposição a situações de elevados ângulos de ataque na asa que levam a maiores taxas de subida, coeficientes de arrasto e conseqüentemente menores velocidades.

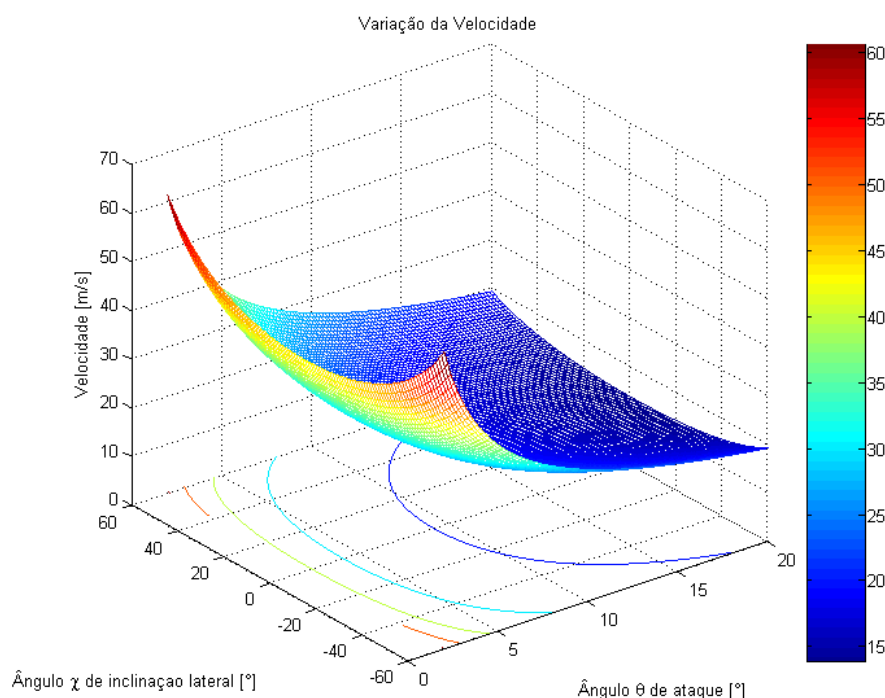


Figura 9.1 - Velocidade

Pode-se observar que para θ próximos de zero a velocidade aproximadamente quadruplica. Isso deve-se ao fato de se ter mantido a propulsão em carga máxima e deve ser evitado por questões estruturais.

9.2. Ângulo de Hélice

Na vertente de desempenho é esta a variável mais importante, pois é ela que determina o poder de ascensão da aeronave. O acréscimo da massa da aeronave, descola a superfície para baixo e, portanto, por essa ferramenta de projeto pode-se determinar a massa máxima ao se estabelecendo uma taxa de subida mínima desejada, tanto em curva quanto em linha reta.

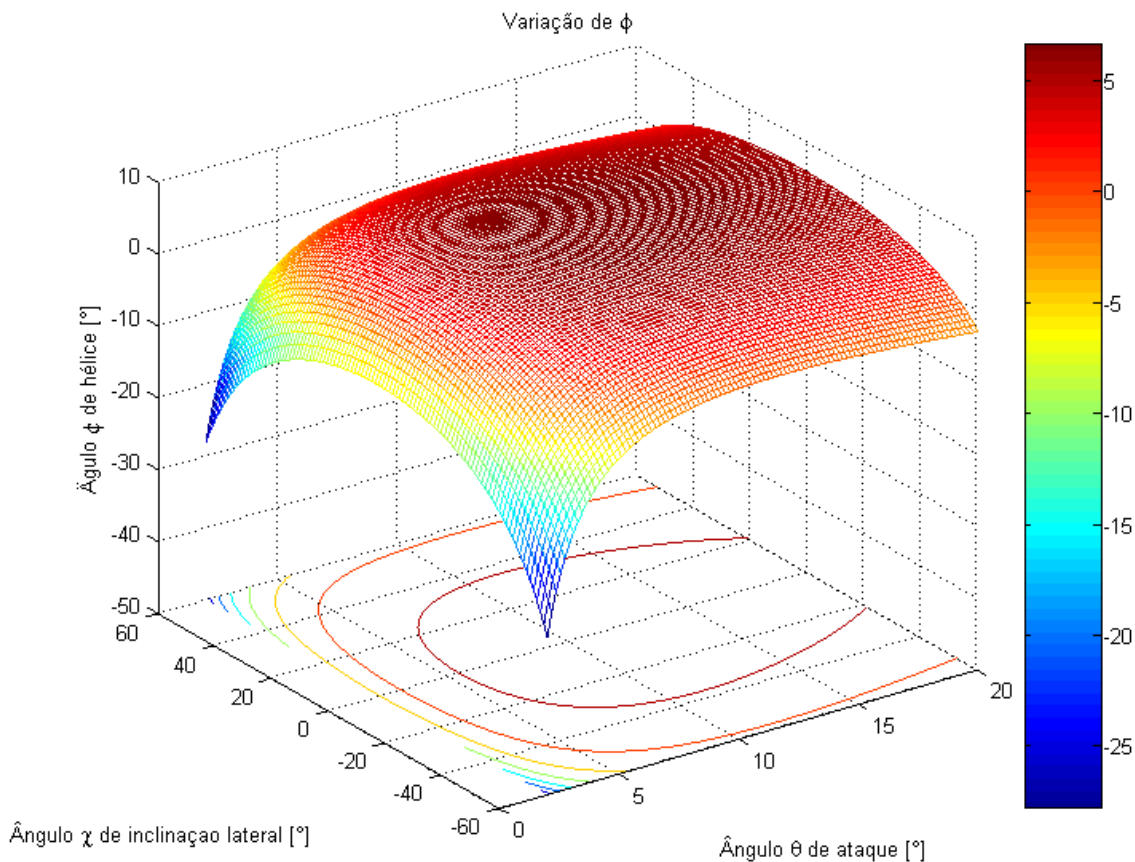


Figura 9.2 - Ângulo de hélice

Para este exemplo, foi utilizada uma massa que não permite com que ângulo de subida ultrapasse aproximadamente 5°, ou seja uma massa absolutamente alta. Normalmente, não se carregam aeronaves de forma a privá-las de taxas até 30° antes do Stall. Como a aeronave em questão competiu na disputa de levantamento máximo de carga, foi-se estipulado em projeto que 5° de “autonomia” eram suficientes para o termino do trajeto.

9.3. Raio de Curva

Verifica-se o comportamento anti-simétrico esperado entre curvas para a direita e esquerda. Observa-se a descontinuidade na superfície plotada, uma vez que o vôo retilíneo obtido para $\chi=0$ é representado por um raio infinito. Para menores ângulos de ataque, ou seja escoamentos menos “estressados” sobre a asa, mostra-se necessária uma maior inclinação lateral da aeronave para que se atinjam menores raios de curva. Isso pode ser explicado, pelo acréscimo de velocidade inerente dessa condição de mergulho.

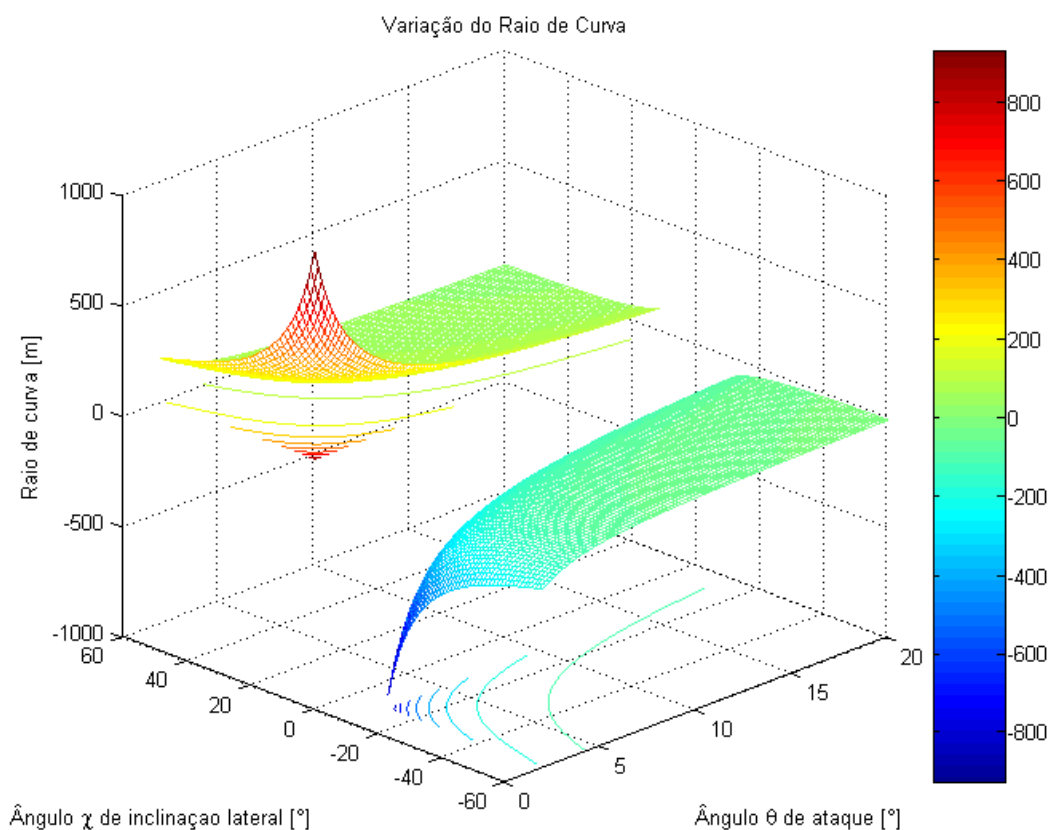


Figura 9.3 - Raio de curva

A interpretação conjunta dos diagramas de velocidade, ângulo φ e raio de curva facilita a compreensão de suas variações.

É interessante calcular o erro introduzido no cálculo do raio ao se desprezar a aceleração lateral do centro de massa conseqüente a atuação do leme. Deve-se recordar que o leme está atuando para garantir o equilíbrio de momentos, promovendo uma pequena força a um grande braço.

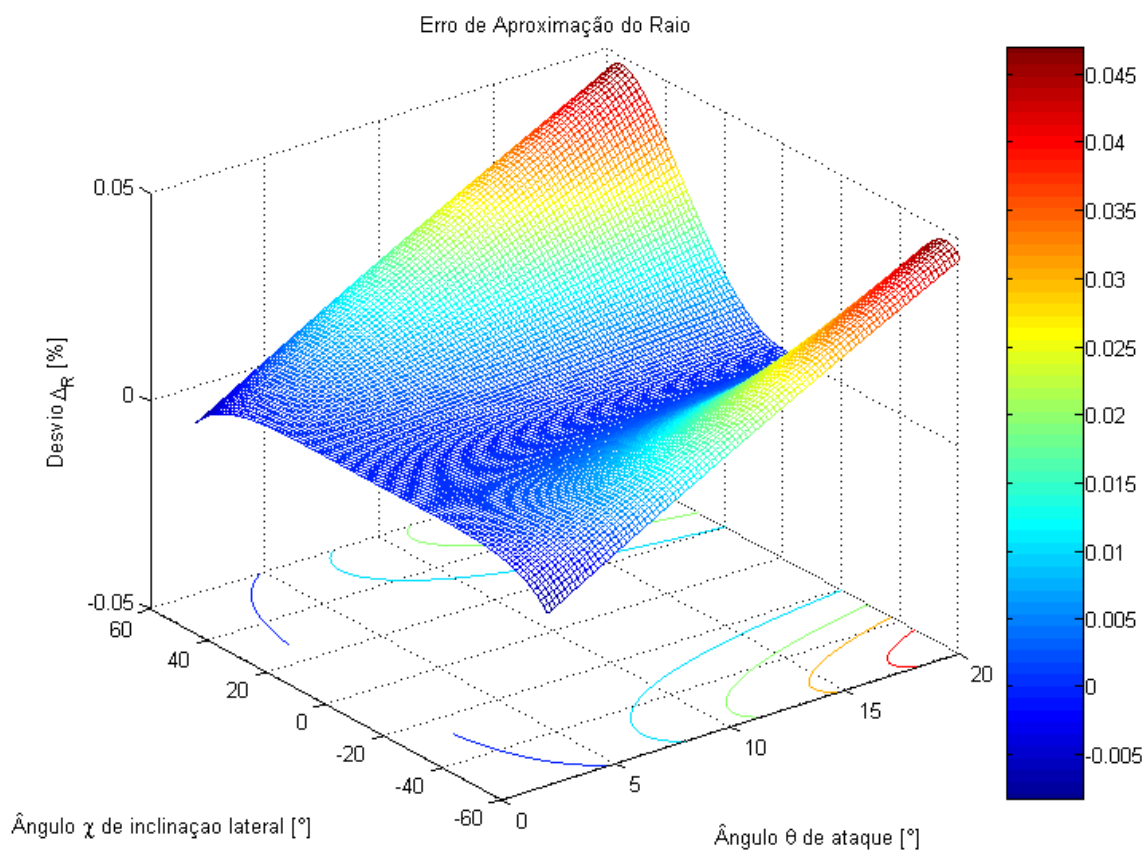


Figura 9.4 - Erro cometido

Observa-se que a influência é absolutamente desprezível, pois a força necessária para a geração do momento requerido é totalmente insignificante quando comparada às outras forças agindo naquela direção.

9.4. Atuação dos Comandos

É notável que o raio de curva não possui influência significativa sobre o equilíbrio de momentos longitudinal. Essa relação seria mais acentuada se os arrastos gerados pela atuação dos outros dois comandos (necessários para vôos em curva) implicassem em momentos da mesma ordem daqueles já avaliados no equilíbrio bidimensional.

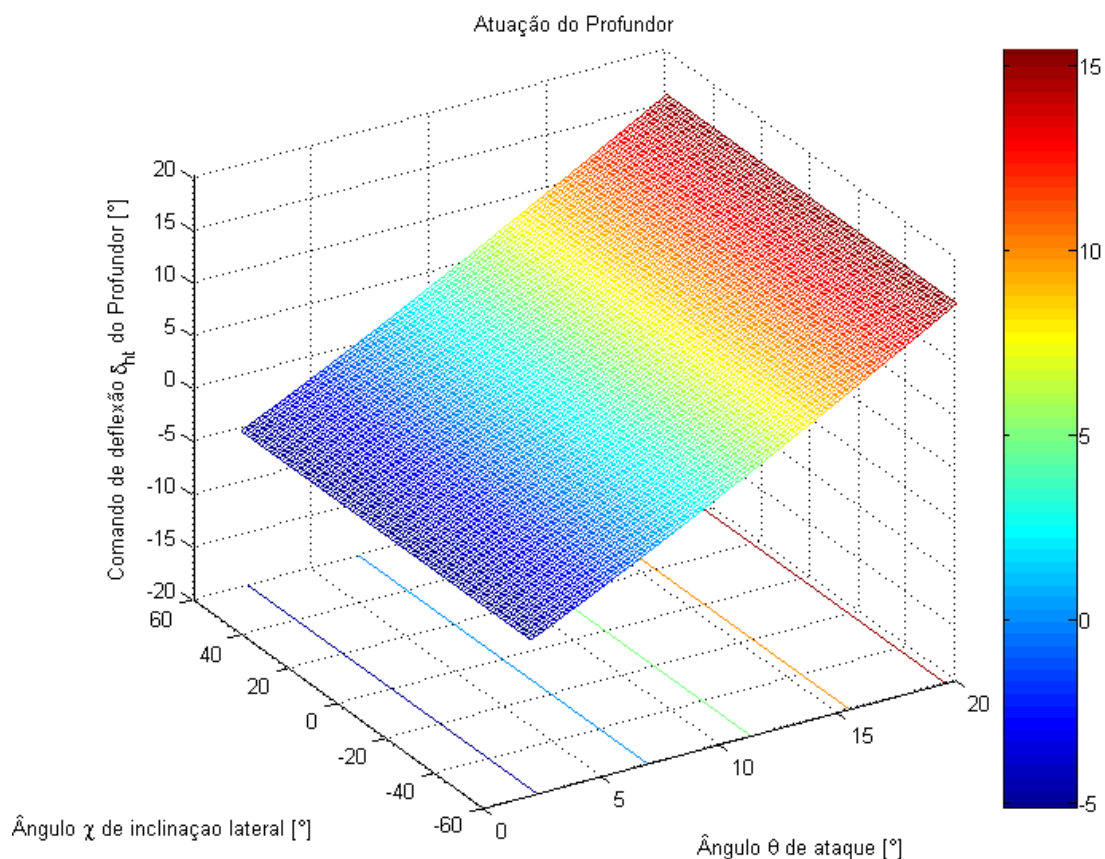


Figura 9.5 - Atuação do profundor

Portanto, o profundor comporta-se de maneira praticamente idêntica tanto para trajetórias retilíneas quanto curvilíneas.

Já em respeito ao leme, por outro lado, os momentos provenientes do vôo em curva geram a necessidade de atuações significativas desse comando.

Tais atuações mostraram-se tão mais necessárias quanto mais próxima a condição de Stall, o que leva a se cogitar possíveis otimizações nessa influência.

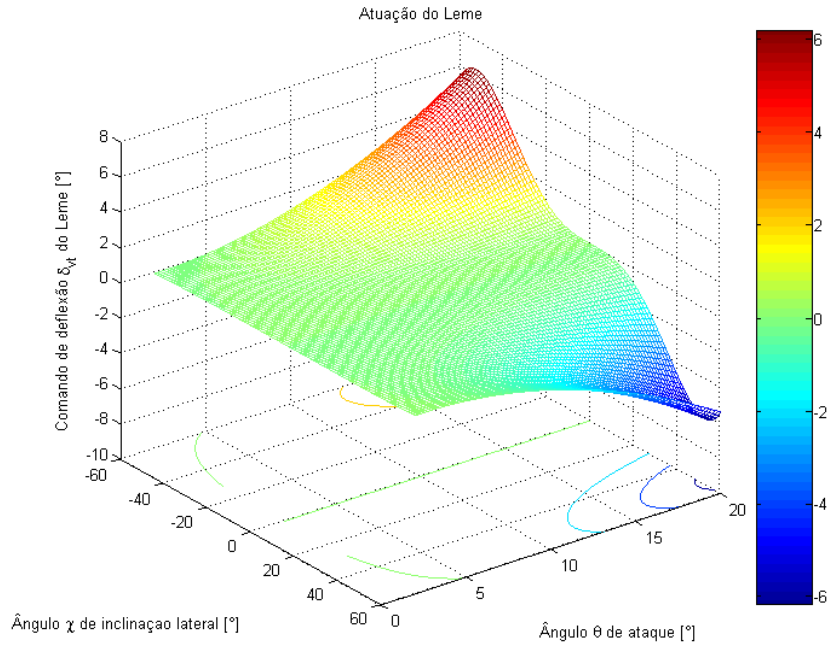


Figura 9.6 - Atuação do Leme

O efeito da atuação dos ailerons mostrou-se o menos representativo. No entanto, as causas de “necessidade de atuação” consideradas foram apenas de natureza cinemática. Caso se considere a não simetria do escoamento ao longo da envergadura da asa devido à rotação da aeronave, amplitudes maiores de atuação são esperadas [8].

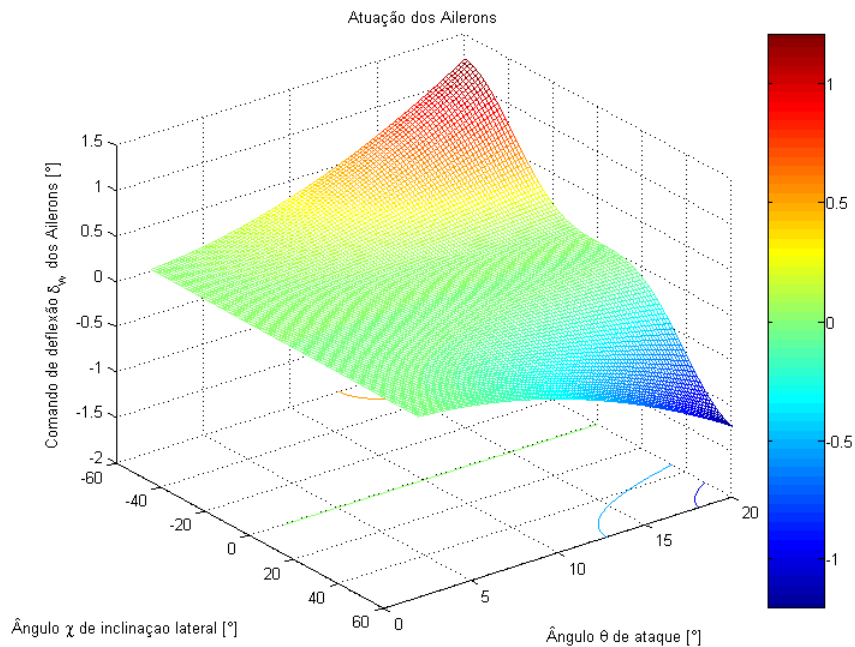


Figura 9.7 - Atuação dos ailerons

10. Considerações sobre a Primeira Parte

No estudo da família de soluções helicoidais a velocidade poderia ter sido fixada, o que é muito comum na literatura, desde que houvesse a garantia de grande versatilidade do sistema de propulsão. Em casos limite, em que a potência disponível é fortemente limitada e a massa carregada deve ser máxima (transporte de carga, estudos de economia de combustível e o Aerodesign, por exemplo), a restrição da variação da velocidade para as diferentes condições de voo privaria o estudo fortemente de sensibilidade.

Por isso, considera-se um avanço o método aqui proposto, que se diferencia de outras abordagens da literatura, não apenas por manter livre a variação de velocidade, mas também por realizá-la de forma polinomial com grau a critério do projetista.

Consolidadas as equações do movimento, bem como um método de obtenção de pontos de operação, as atenções podem começar a ser voltadas para a sofisticação da modelagem aerodinâmica.

As implicações das diferentes velocidades e ângulos de ataque aparentes ao longo da envergadura da asa induzidos pelo movimento curvilíneo é um ponto crucial que requer maior compreensão e aprofundamento.

Pode-se verificar que simplificações normalmente feitas, como por exemplo a relação simplificada para o raio de curva discutida no capítulo 7, são pertinentes e não comprometem o resultado em ordens de grandeza significativas. Ainda assim muitas delas são realizadas em certas referências literárias sem que se calcule anteriormente da maneira completa para que se saiba o que está se desprezando.

Quanto à inclusão dos momentos gerados pela rotação em torno de um eixo que não coincide com nenhum eixo central de inércia, considera-se a trabalho bastante cuidadoso. Apesar de mencionados pela literatura, tais esforços não foram encontrados equacionados nem bem discutidos na bibliografia estudada. Mostraram-se, de fato, menos significativos que os principais esforços normalmente considerados, mas suficientes para comprometer ajustes de trimagem dos comandos.

11. Segunda Etapa

A partir deste ponto surge a necessidade de que se desenvolva uma estrutura computacional para realização dos cálculos propostos na 1ª metade do trabalho, bem como o ambiente virtual que posteriormente servirá para emulação do simulador.

Utilizou-se MATLAB, devido a sua semelhança com linguagem C, bibliotecas abundantes e simplicidade de *debugagem*. Todo código escrito encontra-se em anexo por extenso e em forma digital para que o leitor possa conhecer a interface, e para que os membros da banca também possam avaliá-la. Um tutorial simplificado encontra-se no capítulo 14.

Uma vez que os cálculos consistem em grande parte na integração vetorial numérica, e para que essa possa ser realizada rapidamente, um conjunto de bases foi especificamente deduzido nesta etapa. As bases são absolutamente equivalentes àquelas até então utilizadas, no entanto, não dependem do conhecimento das rotações finitas para sua definição. A dedução desenvolveu-se em Maple e é foco dos capítulos 12 e 13 deste relatório. O leitor familiarizado com mecânica vetorial deve achar as seções interessantes.

Em seguida, discutir-se-á a ambientação gráfica, a metodologia do software, problemas identificados após a implementação e possíveis melhorias da modelo.

12. Bases Numéricas

12.1. As cinco rotações finitas e seu equivalente numérico

O sistema de coordenadas utilizado para definir o vetor velocidade do centro de massa da aeronave e duas outras direções perpendiculares a ele possui três variáveis livres: (V , ψ , φ). Os versores formam a base ortonormal, K'' , descrita abaixo por seus versores.

$$(12.1) \quad t := \begin{bmatrix} \cos(\phi) \cos(\psi) \\ \cos(\phi) \sin(\psi) \\ \sin(\phi) \end{bmatrix} \quad r := \begin{bmatrix} -\sin(\psi) \\ \cos(\psi) \\ 0 \end{bmatrix} \quad n := \begin{bmatrix} -\sin(\phi) \cos(\psi) \\ -\sin(\phi) \sin(\psi) \\ \cos(\phi) \end{bmatrix}$$

A matriz de mudança de base do sistema inercial, $\{K, i, j, k\}$, para o $\{K'', t, r, n\}$ é facilmente definida, pois trata-se de uma transformação entre bases ortonormais.

$$(12.2) \quad M_{k i i k} := \begin{bmatrix} \cos(\phi) \cos(\psi) & \cos(\phi) \sin(\psi) & \sin(\phi) \\ -\sin(\psi) & \cos(\psi) & 0 \\ -\sin(\phi) \cos(\psi) & -\sin(\phi) \sin(\psi) & \cos(\phi) \end{bmatrix}$$

Além disso, $M_{k i i k}$ pode ser interpretada como a transformação que realiza duas rotações finitas ordenadas em função dos ângulos ψ e φ respectivamente, tais que o versor i de K torne-se o t de K'' .

A partir de K'' , genericamente ainda são necessárias mais três operações de rotação para que a base obtida alinhe-se como os eixos centrais de inércia da aeronave.

Primeiramente, realiza-se a inclinação lateral, por um movimento de rolagem em torno do eixo da velocidade, aplicando-se o ângulo χ .

$$(12.3) \quad M_{k i i i k i i} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\chi) & -\sin(\chi) \\ 0 & \sin(\chi) & \cos(\chi) \end{bmatrix}$$

Em seguida, realiza-se a rotação pelo ângulo θ , por um movimento de arfagem em torno do eixo da envergadura.

$$(12.4) \quad M_{kiiiikiii} := \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

E finalmente, por um movimento de guinada, rotaciona-se a aeronave de η , em torno do eixo central “vertical”, na ótica do piloto.

$$(12.5) \quad M_{kiiiiikiii} := \begin{bmatrix} \cos(\eta) & \sin(\eta) & 0 \\ -\sin(\eta) & \cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dessa forma, a matriz completa resultante das operações descritas acima é obtida pelo produto das três matrizes de transformação e possui a forma:

$$(12.6) \quad M_{kiiiiikii} = \begin{bmatrix} \cos(\eta) \cos(\theta) & \sin(\eta) \cos(\chi) + \cos(\eta) \sin(\theta) \sin(\chi) & -\sin(\eta) \sin(\chi) + \cos(\eta) \sin(\theta) \cos(\chi) \\ -\sin(\eta) \cos(\theta) & \cos(\eta) \cos(\chi) - \sin(\eta) \sin(\theta) \sin(\chi) & -\cos(\eta) \sin(\chi) - \sin(\eta) \sin(\theta) \cos(\chi) \\ -\sin(\theta) & \cos(\theta) \sin(\chi) & \cos(\theta) \cos(\chi) \end{bmatrix}$$

Caso deseje-se definir a transformação de K para K''', basta que se multipliquem as transformações K-K'' e K''-K'''.

$$(12.7) \quad M_{kiiiiik} := M_{kiiiiikii} . M_{kiiik}$$

Feita esta pequena revisão das transformações lineares utilizadas na modelagem do sistema dinâmico, pode-se começar discutir o problema numérico computacional.

Algebricamente é praticamente inviável a derivação $M_{kiiiiik}$, para que se possam utilizar as próprias variáveis (ψ , ϕ , χ , θ , η) na obtenção do TMA. Logo, passa a ser interessante a introdução de novas variáveis que definam a posição da base de K''' sem a necessidade que se conheçam diretamente (ψ , ϕ , χ , θ , η).

Dessa maneira, estabelece-se um paralelo. Para K''' haverá duas representações, uma em função das cinco rotações finitas e outra em função do valor numérico de seus versores. Portanto, seja $\{K''', a, b, c\}$, as nove coordenadas que definem os versores (a, b, c) serão variáveis, cujos valores numéricos devem representar os mesmos versores que os obtidos se os mesmos fossem calculados pela $M_{kiiiiik}$ em função de (ψ , ϕ , χ , θ , η).

$$(12.8) \quad ak := \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \quad bk := \begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} \quad ck := \begin{bmatrix} c1 \\ c2 \\ c3 \end{bmatrix}$$

Naturalmente, (a, b, c) não são por hora conhecidos, mas é possível afirmar que sempre existirá esse paralelo, desde que (a, b, c) formem uma base ortonormal. Além disso, a correspondência não é biunívoca, mas essa questão será discutida mais à frente.

Partindo da restrição de ortonormalidade do conjunto (a, b, c) pode-se definir a matriz numérica de mudança de base K-K''''.

$$(12.9) \quad Nkiiiiik := \begin{bmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{bmatrix}$$

A propriedade de transformações ortogonais é muito conveniente na definição da matriz inversa.

$$(12.10) \quad Nkkiiii := \begin{bmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{bmatrix}$$

É importante lembrar que o erro numérico da representação de K'''' é tão maior quanto a matriz abaixo se distanciar da identidade.

$$(12.11) \quad \begin{aligned} check &:= simplify(Nkiiiiik.Nkiiiiik) \\ check &:= \begin{bmatrix} a3^2 + a1^2 + a2^2 & a3 b3 + a1 b1 + a2 b2 & a3 c3 + a1 c1 + a2 c2 \\ a3 b3 + a1 b1 + a2 b2 & b3^2 + b1^2 + b2^2 & b3 c3 + b1 c1 + b2 c2 \\ a3 c3 + a1 c1 + a2 c2 & b3 c3 + b1 c1 + b2 c2 & c3^2 + c1^2 + c2^2 \end{bmatrix} \end{aligned}$$

12.2. Vetor Rotação

Para que se possa aplicar o TMA é preciso conhecer o vetor de rotação do corpo rígido.

Tal vetor é composto pela velocidade angular instantânea em torno de três eixos ortogonais entre si. Convenientemente, utilizam-se os próprios eixos centrais da aeronave para essa descrição. Lembrando que $\{K''''', a, b, c\}$ é por definição esse sistema, obtem-se Ω_{kiiii} em função das velocidades angulares de rolagem, arfagem e guinada que o piloto “enxerga” a cada instante.

$$(12.12) \quad \Omega_{kiiii} := \begin{bmatrix} \alpha \dot{} \\ \beta \dot{} \\ \gamma \dot{} \end{bmatrix}$$

$(\alpha \dot{}, \beta \dot{}, \gamma \dot{})$ são essencialmente entidades numéricas, mas têm sua interpretação cinemática como a composição das taxas de rotação em seus respectivos eixos de todas as operações de rotação finita.

Lançando mão da transformação N_{kiiii} , define-se Ω_k , necessário para a aplicação do TMA.

$$(12.13) \quad \Omega_k := N_{kiiii} \Omega_{kiiii}$$

$$\Omega_k := \begin{bmatrix} a1 \alpha \dot{} + b1 \beta \dot{} + c1 \gamma \dot{} \\ a2 \alpha \dot{} + b2 \beta \dot{} + c2 \gamma \dot{} \\ a3 \alpha \dot{} + b3 \beta \dot{} + c3 \gamma \dot{} \end{bmatrix}$$

Vale a pena ressaltar que o regime permanente estudado na primeira parte do trabalho é o caso particular:

$$(12.14) \quad \Omega_k := \begin{bmatrix} 0 \\ 0 \\ \psi \dot{} \end{bmatrix}$$

E a relação abaixo será importante para a definição das condições iniciais de $(\alpha \dot{}, \beta \dot{}, \gamma \dot{})$ no processo de integração.

$$\Omega_{kiiii} := N_{kiiii} \cdot \Omega_k$$

$$\Omega_{kiiii} := \begin{bmatrix} a_3 \dot{\psi} \\ b_3 \dot{\psi} \\ c_3 \dot{\psi} \end{bmatrix}$$

(12.15)

12.3. Derivação dos Versores

Uma vez que optou-se por utilizar $(\alpha\dot{}, \beta\dot{}, \gamma\dot{})$ definidas no sistema móvel para a descrição do vetor Ω , não se pode esquecer que a base (a, b, c) também varia no tempo.

Logo, na aplicação do TMA será necessário derivar o vetor Ω e irão surgir as derivadas de (a, b, c) .

Para isso recorre-se a relação cinemática clássica que permite definir a vetor taxa de variação, dado o vetor original e seu vetor rotação.

$$\begin{aligned} \dot{a}_{dotk} &:= \text{simplify}(\Omega_k \&x a_k) \\ \dot{a}_{dotk} &:= \begin{bmatrix} a_3 b_2 \beta\dot{} + a_3 c_2 \gamma\dot{} - a_2 b_3 \beta\dot{} - a_2 c_3 \gamma\dot{} \\ a_1 b_3 \beta\dot{} + a_1 c_3 \gamma\dot{} - a_3 b_1 \beta\dot{} - a_3 c_1 \gamma\dot{} \\ a_2 b_1 \beta\dot{} + a_2 c_1 \gamma\dot{} - a_1 b_2 \beta\dot{} - a_1 c_2 \gamma\dot{} \end{bmatrix} \\ \dot{b}_{dotk} &:= \text{simplify}(\Omega_k \&x b_k) \\ \dot{b}_{dotk} &:= \begin{bmatrix} b_3 a_2 \alpha\dot{} + b_3 c_2 \gamma\dot{} - b_2 a_3 \alpha\dot{} - b_2 c_3 \gamma\dot{} \\ b_1 a_3 \alpha\dot{} + b_1 c_3 \gamma\dot{} - b_3 a_1 \alpha\dot{} - b_3 c_1 \gamma\dot{} \\ b_2 a_1 \alpha\dot{} + b_2 c_1 \gamma\dot{} - b_1 a_2 \alpha\dot{} - b_1 c_2 \gamma\dot{} \end{bmatrix} \\ \dot{c}_{dotk} &:= \text{simplify}(\Omega_k \&x c_k) \\ \dot{c}_{dotk} &:= \begin{bmatrix} c_3 a_2 \alpha\dot{} + c_3 b_2 \beta\dot{} - c_2 a_3 \alpha\dot{} - c_2 b_3 \beta\dot{} \\ c_1 a_3 \alpha\dot{} + c_1 b_3 \beta\dot{} - c_3 a_1 \alpha\dot{} - c_3 b_1 \beta\dot{} \\ c_2 a_1 \alpha\dot{} + c_2 b_1 \beta\dot{} - c_1 a_2 \alpha\dot{} - c_1 b_2 \beta\dot{} \end{bmatrix} \end{aligned}$$

(12.16)

Com isso, (a, b, c) também podem ser numericamente integrados e sofrerão as mesmas rotações que a aeronave sofrer, ou seja, a acompanharão.

12.4. Resgate das Rotações Finitas

Os modelos de esforços utilizam como argumento os ângulos (χ, θ, η) . Justamente por isso, escolheram-se essas operações de rotação finita. Uma vez que trabalha-se computacionalmente com o sistema $\{K''''', a, b, c\}$, é necessário que se resgatem os valores de (χ, θ, η) a partir das coordenadas de (a, b, c) .

Para isso, impõe-se a equivalência proposta em (2.1.), buscando uma fórmula para o cálculo de (χ, θ, η) em função de estados conhecidos.

Como (ψ, ϕ) são estados provenientes do TMB, não há problemas em assumí-los conhecidos nestas equações. Logo, isolam-se as incógnitas na matriz $M_{kiiiiikii}$ apenas.

$$\begin{aligned}
 \text{sys} &:= \{N_{kiiiiikii} = M_{kiiiiikii}\} \\
 \text{sys} &= \left[\begin{array}{l} \cos(\phi) \cos(\psi) a1 + \cos(\phi) \sin(\psi) a2 + \sin(\phi) a3 - \sin(\psi) a1 + \cos(\psi) a2 - \sin(\phi) \cos(\psi) a1 - \sin(\phi) \sin(\psi) a2 + \cos(\phi) a3 \\ \cos(\phi) \cos(\psi) b1 + \cos(\phi) \sin(\psi) b2 + \sin(\phi) b3 - \sin(\psi) b1 + \cos(\psi) b2 - \sin(\phi) \cos(\psi) b1 - \sin(\phi) \sin(\psi) b2 + \cos(\phi) b3 \\ \cos(\phi) \cos(\psi) c1 + \cos(\phi) \sin(\psi) c2 + \sin(\phi) c3 - \sin(\psi) c1 + \cos(\psi) c2 - \sin(\phi) \cos(\psi) c1 - \sin(\phi) \sin(\psi) c2 + \cos(\phi) c3 \end{array} \right] \\
 &= \left[\begin{array}{l} \cos(\eta) \cos(\theta) \sin(\chi) \cos(\chi) + \cos(\eta) \sin(\theta) \sin(\chi) - \sin(\eta) \sin(\chi) + \cos(\eta) \sin(\theta) \cos(\chi) \\ -\sin(\eta) \cos(\theta) \cos(\chi) \cos(\chi) - \sin(\eta) \sin(\theta) \sin(\chi) - \cos(\eta) \sin(\chi) - \sin(\eta) \sin(\theta) \cos(\chi) \\ -\sin(\theta) \cos(\theta) \sin(\chi) \cos(\chi) \end{array} \right]
 \end{aligned}
 \tag{12.17}$$

De onde escrevem-se as seguintes fórmulas para (χ, θ, η) :

$$\begin{aligned}
 Eq1 &:= \chi = \arctan\left(\frac{N_{kiiiiikii}[3, 2]}{N_{kiiiiikii}[3, 3]}\right) & Eq1 &:= \chi = \arctan\left(\frac{-\sin(\psi) c1 + \cos(\psi) c2}{-\sin(\phi) \cos(\psi) c1 - \sin(\phi) \sin(\psi) c2 + \cos(\phi) c3}\right) \\
 Eq2 &:= \eta = \arctan\left(-\frac{N_{kiiiiikii}[2, 1]}{N_{kiiiiikii}[1, 1]}\right) & Eq2 &:= \eta = -\arctan\left(\frac{\cos(\phi) \cos(\psi) b1 + \cos(\phi) \sin(\psi) b2 + \sin(\phi) b3}{\cos(\phi) \cos(\psi) a1 + \cos(\phi) \sin(\psi) a2 + \sin(\phi) a3}\right) \\
 Eq3 &:= \theta = \arcsin(-N_{kiiiiikii}[3, 1]) & Eq3 &:= \theta = -\arcsin(\cos(\phi) \cos(\psi) c1 + \cos(\phi) \sin(\psi) c2 + \sin(\phi) c3)
 \end{aligned}
 \tag{12.18}$$

Uma vez que as funções seno e tangente são ímpares, essas fórmulas devem funcionar bem quando centradas no zero. No entanto, para valores superiores ou inferiores a $+\pi/2$ ou $-\pi/2$ respectivamente, é necessário que se utilizem as igualdades dos outros termos das matrizes e uma rotina lógica.

Por outro lado, (χ, θ, η) não podem, segundo o próprio modelo de esforços, se afastarem do zero além do domínio delimitado pelo Stall. Portanto, exceto pelo caso de manobras específicas, tratadas apenas posteriormente no simulador, não devem surgir problemas por hora.

13. TMA Completo

13.1. Nova Base

Toda a discussão do capítulo 2 visou fornecer a “base” para que o TMA pudesse ser aplicado em relação a coordenadas definidas no referencial da aeronave e a integração das equações obtidas fosse facilitada.

Lembrando que K'''' é definido em alinhamento com os eixos centrais da aeronave, a matriz de inércia assume a forma diagonal abaixo.

$$(13.1) \quad J_{k''''ik''''i} := \begin{bmatrix} J1 & 0 & 0 \\ 0 & J2 & 0 \\ 0 & 0 & J3 \end{bmatrix}$$

As coordenadas do momento angular em K'''' são portanto:

$$(13.2) \quad H_{k''''i} := J_{k''''ik''''i} \cdot \Omega_{k''''i} \quad H_{k''''i} := \begin{bmatrix} J1 \, \alpha \dot{} \\ J2 \, \beta \dot{} \\ J3 \, \gamma \dot{} \end{bmatrix}$$

Contudo, para que possa aplicar a segunda lei de Newton, o momento angular deve estar descrito em relação a um sistema inercial, no caso K.

$$(13.3) \quad H_k := N_{k''''i} \cdot J_{k''''ik''''i} \cdot \Omega_{k''''i} \quad H_k := \begin{bmatrix} a1 \, J1 \, \alpha \dot{} + b1 \, J2 \, \beta \dot{} + c1 \, J3 \, \gamma \dot{} \\ a2 \, J1 \, \alpha \dot{} + b2 \, J2 \, \beta \dot{} + c2 \, J3 \, \gamma \dot{} \\ a3 \, J1 \, \alpha \dot{} + b3 \, J2 \, \beta \dot{} + c3 \, J3 \, \gamma \dot{} \end{bmatrix}$$

Pela regra do produto, deriva-se Hk.

$$\begin{aligned}
 dHdk &:= Ndotkiiii.Jkiiiiii.\Omega kiii + Nkiiii.Jdotkiiiiii.\Omega kiii + Nkiiii.Jkiiiiii.\Omega dotkiiii \\
 (13.4) \quad dHdk &= Ndotkiiii. \begin{bmatrix} J1 \ \alpha dot \\ J2 \ \beta dot \\ J3 \ \gamma dot \end{bmatrix} + \begin{bmatrix} a1 \ b1 \ c1 \\ a2 \ b2 \ c2 \\ a3 \ b3 \ c3 \end{bmatrix} .Jdotkiiiiii. \begin{bmatrix} \alpha dot \\ \beta dot \\ \gamma dot \end{bmatrix} + \begin{bmatrix} a1 \ J1 \ b1 \ J2 \ c1 \ J3 \\ a2 \ J1 \ b2 \ J2 \ c2 \ J3 \\ a3 \ J1 \ b3 \ J2 \ c3 \ J3 \end{bmatrix} .\Omega dotkiiii
 \end{aligned}$$

As coordenadas do vetor dHdt pode então ser transformadas de volta para o sistema K'''' .

$$\begin{aligned}
 dHdkiiii &:= Nkiiii.Ndotkiiii.Jkiiiiii.\Omega kiii + Jdotkiiiiii.\Omega kiii + Jkiiiiii.\Omega dotkiiii \\
 (13.5) \quad dHdkiiii &= \begin{bmatrix} a1 \ a2 \ a3 \\ b1 \ b2 \ b3 \\ c1 \ c2 \ c3 \end{bmatrix} .Ndotkiiii. \begin{bmatrix} J1 \ \alpha dot \\ J2 \ \beta dot \\ J3 \ \gamma dot \end{bmatrix} + Jdotkiiiiii. \begin{bmatrix} \alpha dot \\ \beta dot \\ \gamma dot \end{bmatrix} + \begin{bmatrix} J1 \ 0 \ 0 \\ 0 \ J2 \ 0 \\ 0 \ 0 \ J3 \end{bmatrix} .\Omega dotkiiii
 \end{aligned}$$

Onde:

$$\begin{aligned}
 Ndotkiiiiii &:= \begin{bmatrix} adotk_1 \ bdotk_1 \ cdotk_1 \\ adotk_2 \ bdotk_2 \ cdotk_2 \\ adotk_3 \ bdotk_3 \ cdotk_3 \end{bmatrix} \\
 (13.6)
 \end{aligned}$$

$$\begin{aligned}
 Jdotkiiiiii &:= \begin{bmatrix} dJ1dt & 0 & 0 \\ 0 & dJ2dt & 0 \\ 0 & 0 & dJ3dt \end{bmatrix} \\
 (13.7)
 \end{aligned}$$

$$\begin{aligned}
 \Omega dotkiiii &:= \begin{bmatrix} \alpha 2dot \\ \beta 2dot \\ \gamma 2dot \end{bmatrix} \\
 (13.8)
 \end{aligned}$$

Os esforços aerodinâmicos serão definidos pelo vetor de momentos em relação ao centro de massa explícitos na base K''''.

$$(13.9) \quad T_{kiiii} := \begin{bmatrix} T_a \\ T_b \\ T_c \end{bmatrix}$$

Uma vez que o pólo escolhido para a definição dos momentos é o centro de massa, o TMA possui a forma direta:

$$(13.10) \quad TMA := \{T_{kiiii} - N_{kiiii} \cdot \dot{N}_{kiiii} \cdot J_{kiiii} \cdot \dot{\Omega}_{kiiii} - J_{\dot{\Omega}_{kiiii} kiiii} \cdot \dot{\Omega}_{kiiii} = J_{kiiii} \cdot \dot{\Omega}_{kiiii}\}$$

Três equações para as acelerações angulares (α_2 , β_2 , γ_2) são obtidas e todos os estados passam a ter suas derivadas definidas, logo o problema torna-se integrável.

$$(13.11) \quad \begin{aligned} & \text{simplify}(TMA(1)) \\ & \{J1 \cdot \alpha_2 = T_a - J2 \cdot \beta_2 \cdot a1 \cdot b3 \cdot c2 \cdot \gamma_2 + J2 \cdot \beta_2 \cdot a1 \cdot b2 \cdot c3 \cdot \gamma_2 - J2 \cdot \beta_2 \cdot a2 \cdot b1 \cdot c3 \cdot \gamma_2 + J2 \cdot \beta_2 \cdot a2 \cdot b3 \cdot c1 \cdot \gamma_2 - J2 \cdot \beta_2 \cdot a3 \cdot b2 \cdot c1 \cdot \gamma_2 \\ & + J2 \cdot \beta_2 \cdot a3 \cdot b1 \cdot c2 \cdot \gamma_2 - J3 \cdot \gamma_2 \cdot a1 \cdot c3 \cdot b2 \cdot \beta_2 + J3 \cdot \gamma_2 \cdot a1 \cdot c2 \cdot b3 \cdot \beta_2 - J3 \cdot \gamma_2 \cdot a2 \cdot c1 \cdot b3 \cdot \beta_2 + J3 \cdot \gamma_2 \cdot a2 \cdot c3 \cdot b1 \cdot \beta_2 - J3 \cdot \gamma_2 \cdot a3 \cdot c2 \cdot b1 \cdot \beta_2 \\ & + J3 \cdot \gamma_2 \cdot a3 \cdot c1 \cdot b2 \cdot \beta_2 - dJ1 \cdot \alpha_2\} \\ & \text{simplify}(TMA(2)) \\ & \{J2 \cdot \beta_2 = T_b - J1 \cdot \alpha_2 \cdot a3 \cdot b1 \cdot c2 \cdot \gamma_2 + J1 \cdot \alpha_2 \cdot a2 \cdot b1 \cdot c3 \cdot \gamma_2 - J1 \cdot \alpha_2 \cdot a1 \cdot b2 \cdot c3 \cdot \gamma_2 + J1 \cdot \alpha_2 \cdot a3 \cdot b2 \cdot c1 \cdot \gamma_2 - J1 \cdot \alpha_2 \cdot a2 \cdot b3 \cdot c1 \cdot \gamma_2 \\ & + J1 \cdot \alpha_2 \cdot a1 \cdot b3 \cdot c2 \cdot \gamma_2 - J3 \cdot \gamma_2 \cdot b1 \cdot c3 \cdot a2 \cdot \alpha_2 + J3 \cdot \gamma_2 \cdot b1 \cdot c2 \cdot a3 \cdot \alpha_2 - J3 \cdot \gamma_2 \cdot b2 \cdot c1 \cdot a3 \cdot \alpha_2 + J3 \cdot \gamma_2 \cdot b2 \cdot c3 \cdot a1 \cdot \alpha_2 - J3 \cdot \gamma_2 \cdot b3 \cdot c2 \cdot a1 \cdot \alpha_2 \\ & + J3 \cdot \gamma_2 \cdot b3 \cdot c1 \cdot a2 \cdot \alpha_2 - dJ2 \cdot \beta_2\} \\ & \text{simplify}(TMA(3)) \\ & \{J3 \cdot \gamma_2 = T_c - J1 \cdot \alpha_2 \cdot a3 \cdot c1 \cdot b2 \cdot \beta_2 + J1 \cdot \alpha_2 \cdot a2 \cdot c1 \cdot b3 \cdot \beta_2 - J1 \cdot \alpha_2 \cdot a1 \cdot c2 \cdot b3 \cdot \beta_2 + J1 \cdot \alpha_2 \cdot a3 \cdot c2 \cdot b1 \cdot \beta_2 - J1 \cdot \alpha_2 \cdot a2 \cdot c3 \cdot b1 \cdot \beta_2 \\ & + J1 \cdot \alpha_2 \cdot a1 \cdot c3 \cdot b2 \cdot \beta_2 - J2 \cdot \beta_2 \cdot b3 \cdot c1 \cdot a2 \cdot \alpha_2 + J2 \cdot \beta_2 \cdot b2 \cdot c1 \cdot a3 \cdot \alpha_2 - J2 \cdot \beta_2 \cdot b1 \cdot c2 \cdot a3 \cdot \alpha_2 + J2 \cdot \beta_2 \cdot b3 \cdot c2 \cdot a1 \cdot \alpha_2 - J2 \cdot \beta_2 \cdot b2 \cdot c3 \cdot a1 \cdot \alpha_2 \\ & + J2 \cdot \beta_2 \cdot b1 \cdot c3 \cdot a2 \cdot \alpha_2 - dJ3 \cdot \gamma_2\} \end{aligned}$$

13.2. Prova Real

Dada a grande extensão das fórmulas e deduções é conveniente possuir um método para testar o integrador. Partindo das soluções de regime permanente, (V, R, ϕ) conhecidos, busca-se uma fórmula para os momentos que a aeronave deve sofrer, para que a trajetória seja uma helicóide perfeita.

Se essa fórmula for utilizada para fornecer as entradas de esforços do integrador e o mesmo retornar uma helicóide com o raio e passo predeterminados, então não há erros de programação e dedução.

Parte-se do vetor rotação de regime permanente.

$$(13.12) \quad \Omega_{kiiii} := \begin{bmatrix} (\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \psi_{dot} \\ -\sin(\chi) \cos(\phi) \psi_{dot} \\ (-\sin(\theta) \sin(\phi) + \cos(\theta) \cos(\chi) \cos(\phi)) \psi_{dot} \end{bmatrix}$$

Aplica-se o TMA sem termos de variação de inércia nem aceleração angular.

$$(13.13) \quad dH_{dtkiiii} := M_{kiiii} \cdot \dot{M}_{kiki} \cdot M_{kiki} \cdot M_{kiiii} \cdot J_{kiiii} \cdot \Omega_{kiiii}$$

Recorre-se ao fato de que o único ângulo com derivada não nula é ψ .

$$(13.14) \quad \dot{M}_{kiki} := \psi_{dot} \begin{bmatrix} -\sin(\psi) & -\cos(\psi) & 0 \\ \cos(\psi) & -\sin(\psi) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

E obtem-se uma fórmula para os momentos aplicados ao centro de massa que garantem a helicóide desejada.

$$(13.15) \quad TMA_{RP} := \left(T_{kiiii} = dH_{dtkiiii} \right) \quad TMA_{RP} := \left[\begin{array}{c} T_a \\ T_b \\ T_c \end{array} \right] = \left[\begin{array}{c} \psi_{dot}^2 (J3 - J2) \sin(\chi) \cos(\phi) (\sin(\theta) \sin(\phi) - \cos(\theta) \cos(\chi) \cos(\phi)) \\ \psi_{dot}^2 (J1 - J3) (-\sin(\theta) \cos(\theta) \sin(\phi)^2 + 2 \cos(\chi) \cos(\theta)^2 \cos(\phi) \sin(\phi) + \cos(\chi)^2 \cos(\theta) \cos(\phi)^2 \sin(\theta) - \sin(\phi) \cos(\chi) \cos(\phi)) \\ \psi_{dot}^2 (J1 - J2) (\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \sin(\chi) \cos(\phi) \end{array} \right]$$

14. O Programa

O programa para realização dos cálculos e simulações é escrito de forma a manter a solução de regime permanente como eixo central da análise. Todos os *inputs* do usuário são utilizados para a determinação de estados de vôo compensados e esses são empregados como condições iniciais e finais das simulações.

Uma seqüência de *callbacks* é estabelecida para a atualização de valores mediante a alteração de alguma entrada. O disparo dessas atualizações é automático após a alteração. Caso o computador utilizado para a execução do programa seja suficientemente potente, o usuário terá a impressão de poder “ajustar” as saídas em função de parâmetros de entrada apenas monitorando a interface gráfica.

Divide-se o programa em quinze rotinas. Infelizmente, elas devem em sua maioria ser executadas seqüencialmente e a paralelização do programa não é possível. No entanto, a rotina para o cálculo dos regimes de vôo pode ser paralelizada internamente o que potencialmente pode acelerar o *feedback* ao usuário, uma vez que é justamente a rotina em questão a mais extensa e demorada.

A vertente prática do trabalho, permitir e facilitar o desenvolvimento e estudo de aeronaves, depende da “acessibilidade” do usuário às ferramentas. Para isso, uma interface gráfica é desenvolvida com o intuito de aproximar o projetista dos cálculos. A utilização de *callbacks* foi essencial para fornecer uma realimentação praticamente instantânea das influências que a alteração de um parâmetro tem no comportamento dinâmico.

14.1. Estrutura

A estrutura esquemática é apresentada na figura abaixo. A alteração de qualquer dado no primeiro nível da cadeia implica na atualização dos demais.

Funções operacionais ou secundárias não estão representadas.

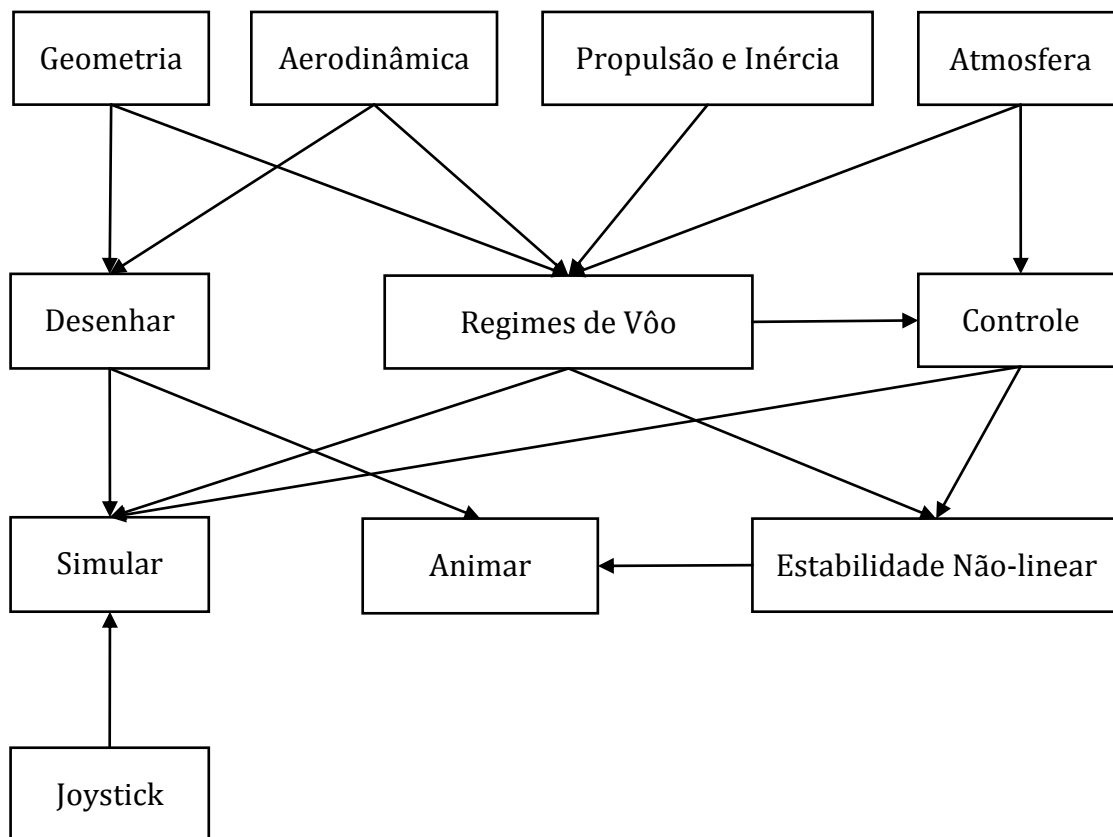


Figura 14.1 - Hierarquia das funções

14.2. Funções

As funções que apresentam janela gráfica são nesta sub-seção descritas. Deseja-se fornecer um tutorial simplificado.

14.2.1. Main

Cada botão abre uma janela com diferentes funções. Os quatro primeiros permitem a entradas de parâmetros que definem a aeronave. Os três seguintes referem-se a operações sobre o esboço da aeronave. “Regimes de Vôo”, “Estabilidade” e “Simulador” abrem as janelas correspondentes. O botão do simulador é apenas ativado se um regime de vôo está selecionado e a aeronave foi posteriormente esboçada. “Save” e “Load” exporta/importa dados do modelo em questão para execuções futuras do programa.

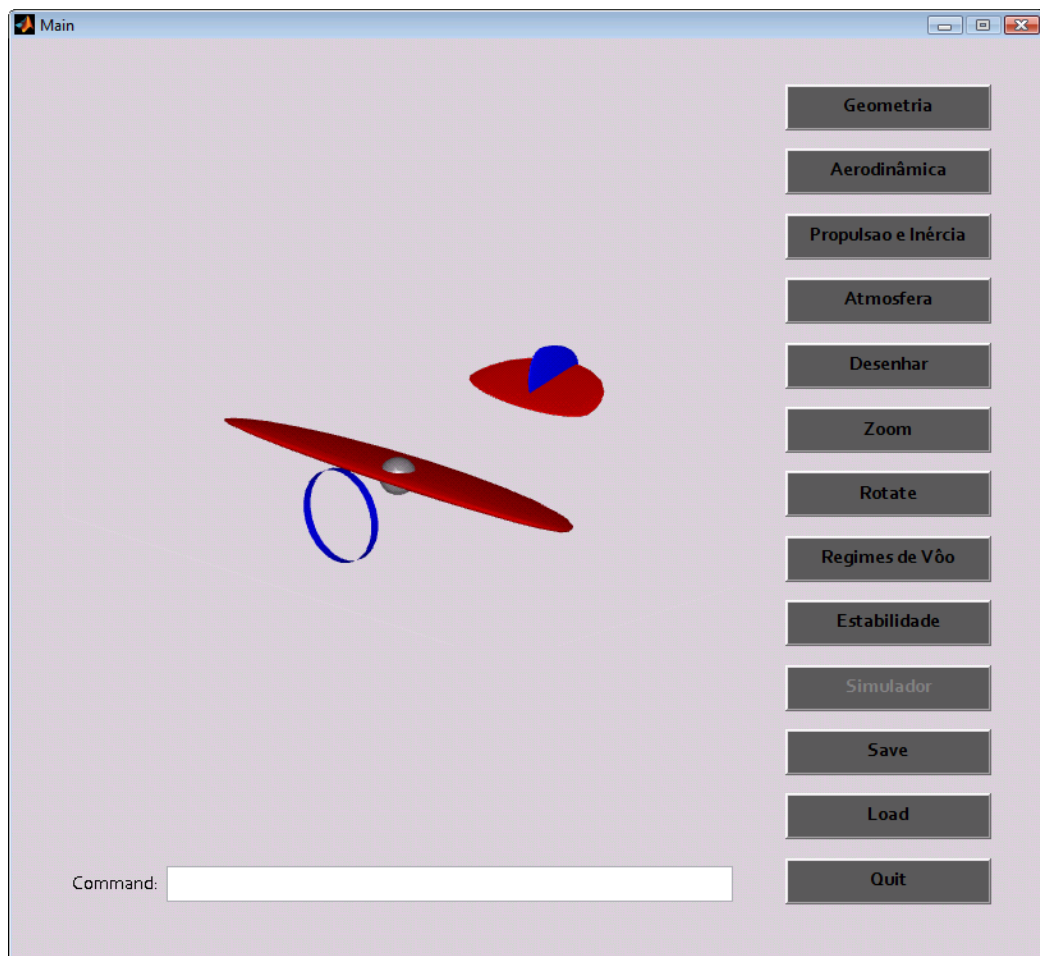


Figura 14.2 - Main

14.2.2. Parâmetros e Constantes

A janela de Geometria recebe os valores de área e envergadura das superfícies sustentadoras, bem como as coordenadas polares do centro aerodinâmico das fontes de esforços: Asa, Profundor, Leme, Fuselagem, Propulsão.

A janela de Aerodinâmica recebe valores relacionados ao modelos de sustentação, arrasto e momentos da respectiva fonte.

A janela de Propulsão e Inércia salva grandezas da inércia da aeronave e coeficientes de uma curva de segundo grau para a força tratora do sistema de propulsão em função da velocidade do escoamento.

A janela de Atmosfera define vento ao longe e densidade do ar.

The image shows a screenshot of a software interface with four panels, each containing input fields for different aerodynamic parameters. The panels are titled 'Geometria', 'Aerodinamica', 'Propulsao e Ine...', and 'Atmosfera'.

Geometria Panel:

- Asa:** Envergadura (2.3 m), Area (1 m²), Braço CA (0.0316 m), Angulo CA (18°).
- Profundor:** Envergadura (0.8 m), Area (0.27 m²), Braço CA (0.91 m), Angulo CA (168°).
- Leme:** Envergadura (0.3 m), Area (0.08 m²), Braço CA (1 m), Angulo CA (165°).
- Fuselagem:** Area (0 m²), Braço CA (0 m), Angulo CA (0°).
- Propulsao:** Braço (0.4 m), gamma (-15°), beta (15°).

Aerodinamica Panel:

- Asa:** CL alpha (4.04 1/rad), alpha L=O (-11.63°), CDO (0.02 -), Ar (5.29 -), Oswald (0.88 -), Corda (0.47 m), CMO Pitch (-0.24 -), CM Y/deltaw (-0.5 1/rad), CM R/deltaw (1 1/rad), CM R/Y (0.1 -), CM R/R (-1 -), CM Y/Y (-1 -), CM Y/R (0.05 -).
- Profundor:** CL alpha (1.53 1/rad), alpha L=O (15°), CDO (0.02 -), CL delta (-1.5 1/rad), Ar (2.3704 -), Oswald (0.33 -), Corda (0.329 m), CMO Pitch (0 -), CM del Pitch (0 1/rad).
- Leme:** Ar (1.125 -), CL alpha (1 1/rad), CL delta (-1.5 1/rad), CDO (0.03 -).
- Fuselagem:** CLO (0 -), CL alpha (0 1/rad), CDO (0 -), CD alpha (0 1/rad), Corda (0 m), CMO (0 -), CM alpha P (0 1/rad).

Propulsao e Ine... Panel:

- Propulsao:** ao (39.39 N), a1 (-0.4153 kg/s), a2 (-0.0205 kg/m), p (1 -).
- Inércia Inicial:** Massa (15 kg), J1 (0.3 kg.m²), J2 (0.4 kg.m²), J3 (0.5 kg.m²).
- Inércia Variável:** dm/dt (0 kg/s), dJ1/dt (0 kg.m²/s), dJ2/dt (0 kg.m²/s), dJ3/dt (0 kg.m²/s).
- Referência:** thetao (0°), rhoo (1.088 kg/m³).

Atmosfera Panel:

- Vento:** u (0 m/s), v (0 m/s), w (0 m/s).
- Rajada:** du (0 m/s), dv (0 m/s), dw (0 m/s).
- Altitude/Densidade:** rho (1.109 kg/m³), Alt (1026 m).

Figura 14.3 - Parâmetros e Constantes

14.2.3. Regimes de Voo

Nesta subrotina, as soluções de regime permanente são calculadas conforme o metodologia desenvolvida na primeira metade do trabalho. As curvas de nível das soluções (V , R , ϕ) são traçadas sobrepostas. A parametrização utilizada varia o ângulo efetivo de ataque na asa e o ângulo de inclinação lateral.

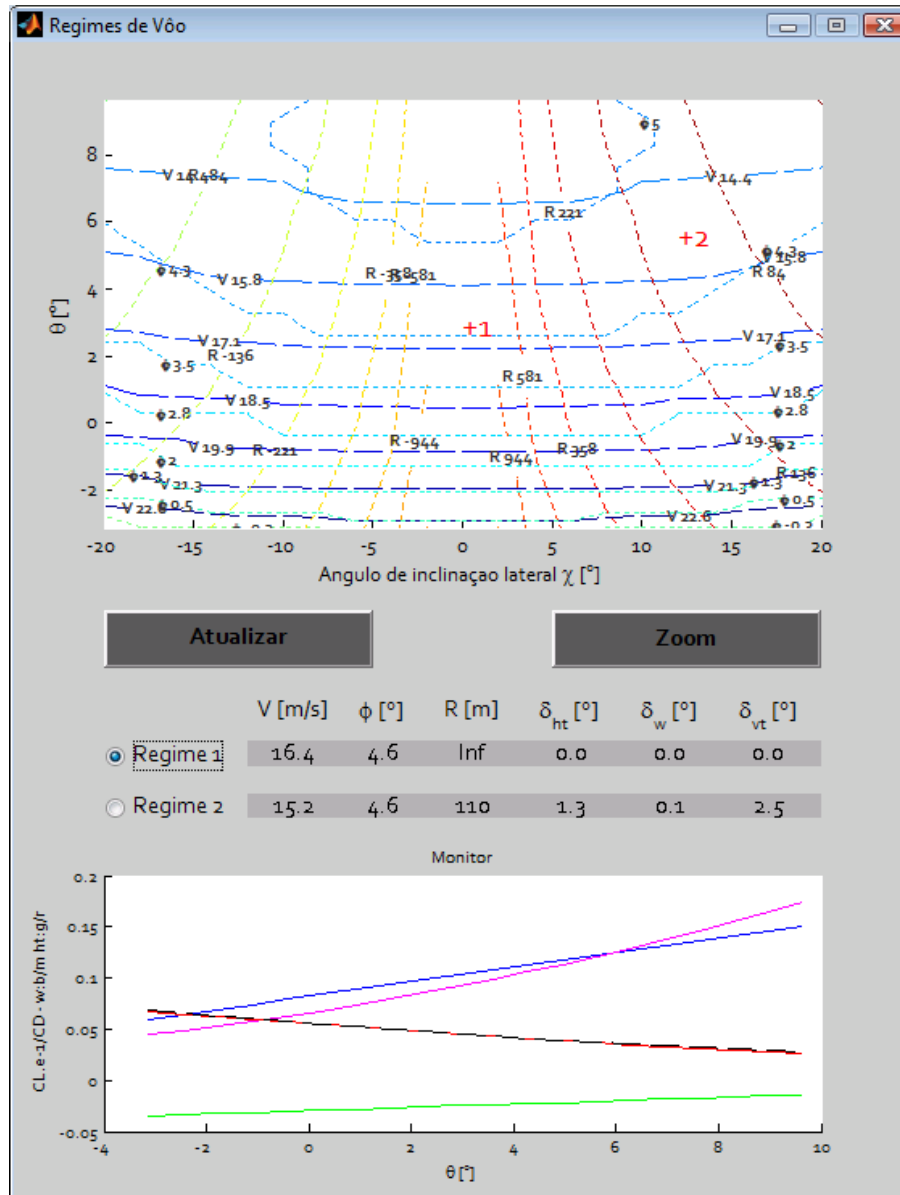


Figura 14.4 - Regimes de Voo

Os pontos 1 e 2 devem ser selecionados com o mouse e estabelecem as condições inicial e final para a rotina de estabilidade. A curva chamada “Monitor” exibe a discrepância da linearização do arrasto induzido do profundor em relação à fórmula com o termo quadrático.

14.2.4. Estabilidade Não-Linear

Após definidos os estados 1 e 2, ou seja os extremos da manobra que se deseja executar, o integrador realiza a migração de um estado para o outro variando os comandos conforme a interpolação escolhida: Degrau, Rampa, Parábola, Sigmoidal.

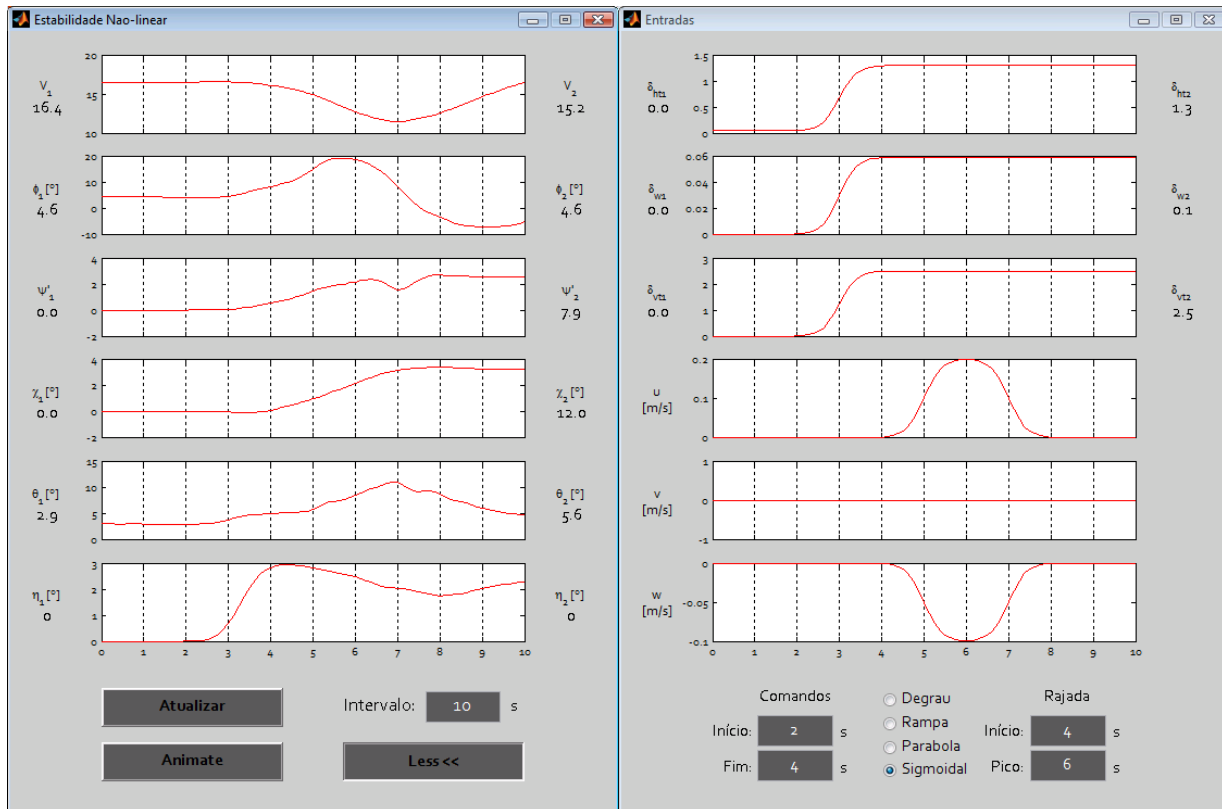


Figura 14.5 - Estabilidade e Controle

O intervalo de tempo total da simulação pode ser determinado para garantir que haja tempo para convergência ou divergência.

14.3. Problemas

Quanto a programação dos esforços aerodinâmicos na rotina de estabilidade, houve dificuldades. O modelo não estava inicialmente adequado para computar escoamento lateralmente não-alinhado com a fuselagem. Cálculos vetoriais para a decomposição de forças oblíquas precisaram ser refeitos. Além disso, a resultante da propulsão também foi re-escrita para computar guinada.

Houve também a necessidade de utilizar-se o método de busca extensiva do par (V, φ) e não o método de linearização proposto no capítulo 8. A solução linearizada apresentou baixa qualidade (resíduo de aceleração quando substituída nas equações originais), pois a linearização, ainda que para φ pequenos, afeta os termos de ordem mais significativa das equações.

Logo, as duas últimas variáveis a serem resolvidas devem ser obtidas por uma busca numérica, em que se procura um par (V, φ) que atende simultaneamente às duas equações restantes do sistema. Essa busca pode ser estruturada vetorizando $V_1(\varphi)$ e $V_2(\varphi)$, i.e. a velocidade calculada pela 1ª e 2ª equação respectivamente, e comparando os vetores.

14.3.1. Ruído Numérico

Mantendo o TMB e o TMA em equilíbrio em torno da solução de uma helicóide, (utilizando a fórmula da prova real), estuda-se a influência de erro numérico acumulado pelo solver. O função utilizada para a integração é a ode113 que é a recomendada para casos de controle máximo de erro, mas é mais lenta que a ode45, segundo bibliografia do MATLAB.

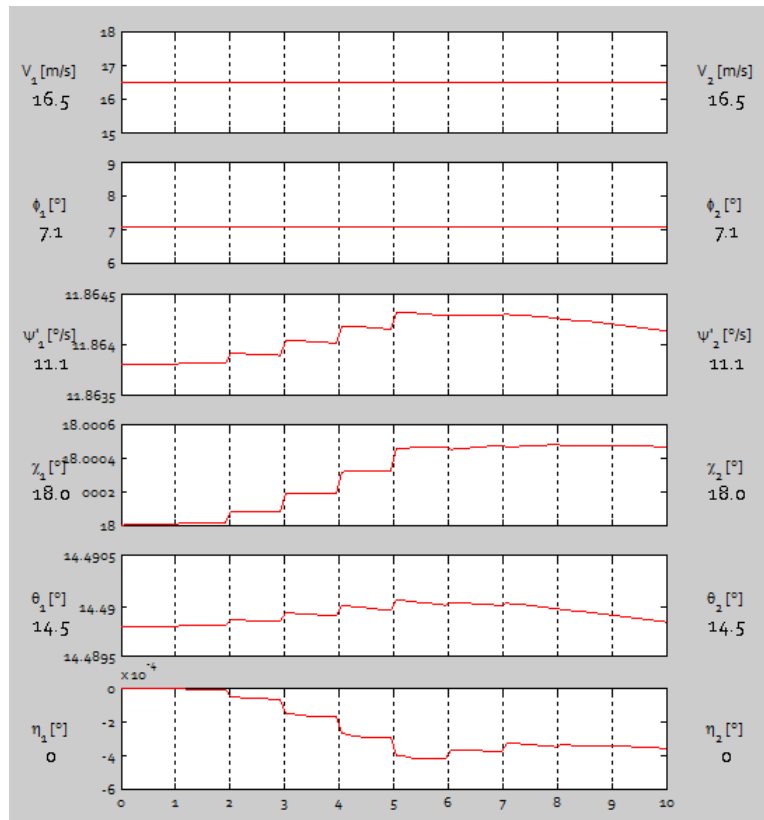


Figura 14.6 - Ruído numérico

Pode-se verificar que o resíduo acumulado em 10s de simulação, mantendo um raio de curva de aproximadamente 80m, é da ordem de 10^{-4} quando comparado com a unidade da grandeza.

Para 100s de simulação, mantendo a as mesmas condições de equilíbrio, observa-se o início da divergência.

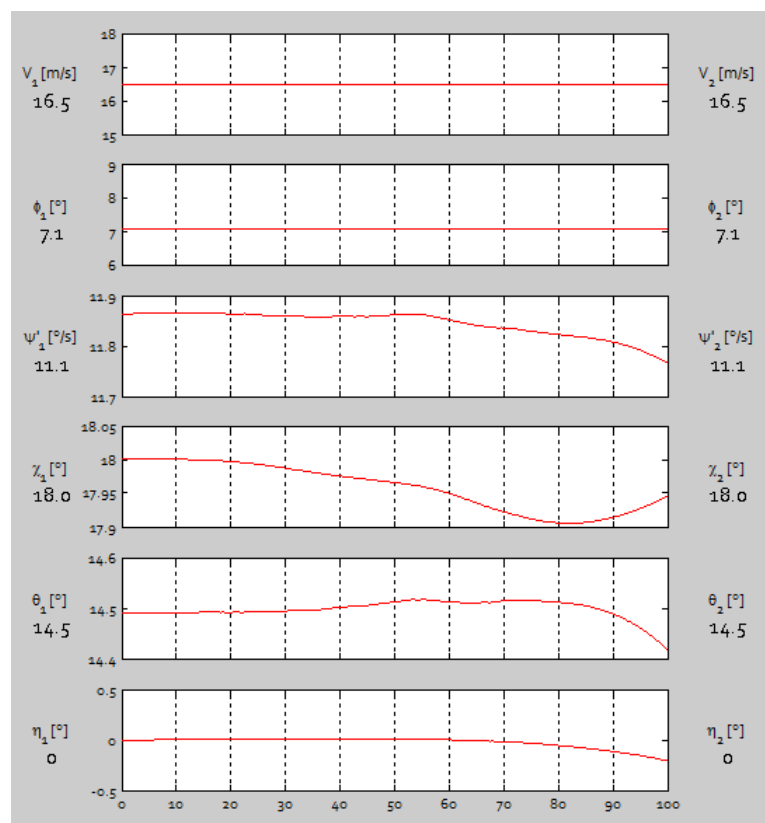


Figura 14.7 - Divergência ode113 - 100s

Visualização 3D da trajetória (cores indicam versores da base de K''''').

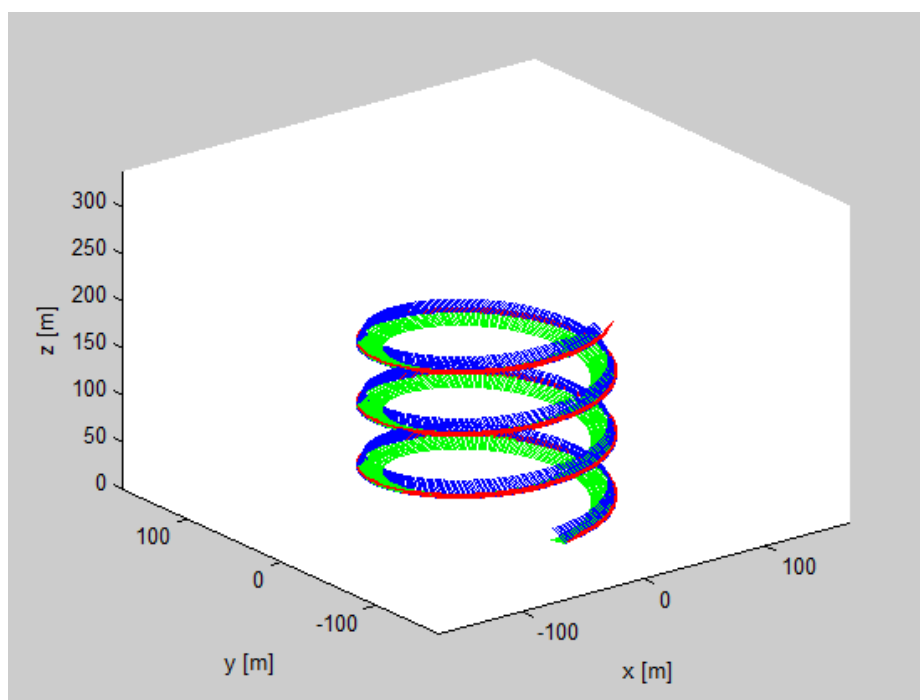


Figura 14.8 - Resultado 3D - 100s

Após 200s, fica clara a divergência.

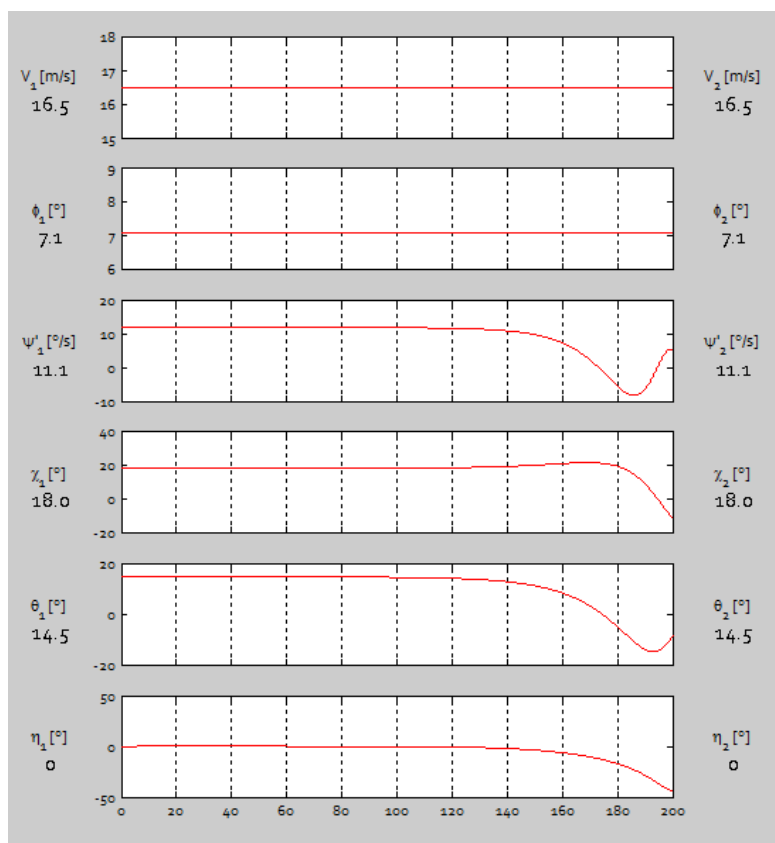


Figura 14.9 - Divergência ode113 - 200s

Visualização 3D da trajetória (cores indicam versores da base de K''''').

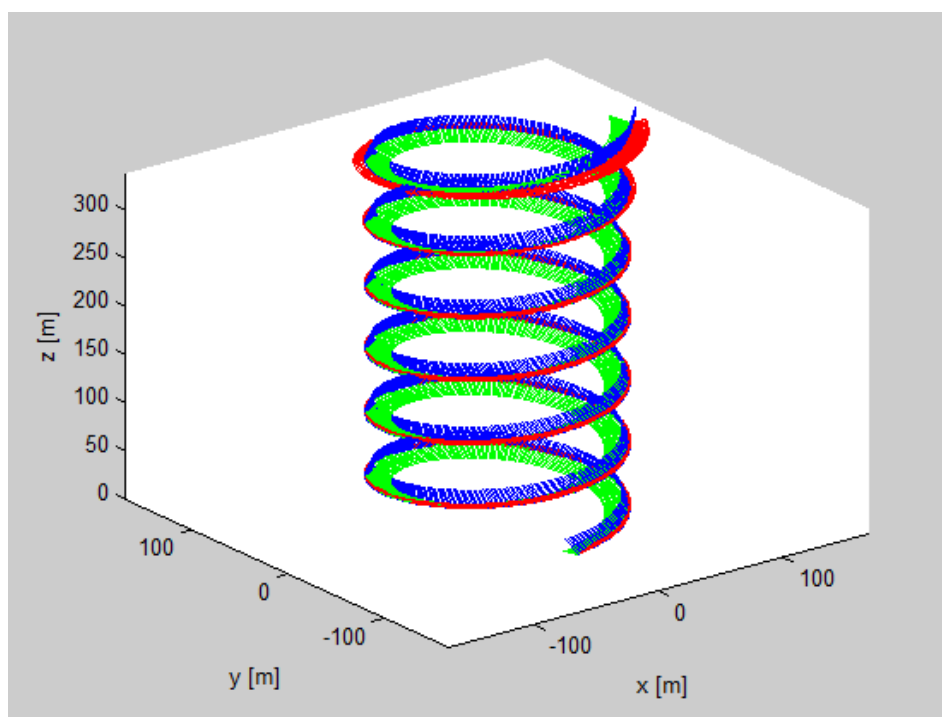


Figura 14.10 - Resultado 3D - 200s

Para efeito de comparação a mesma simulação foi realizada utilizando a ode45, e a divergência foi ocorreu em aproximadamente metade do tempo de simulação.

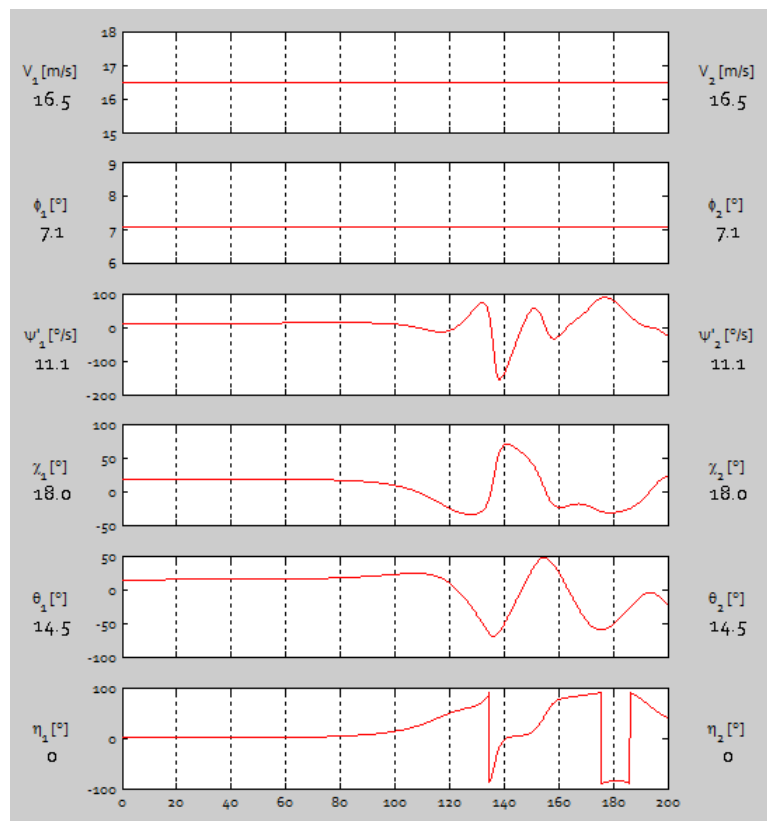


Figura 14.11 - Divergência ode45 - 200s

Visualização 3D da trajetória (cores indicam versores da base de K''''').

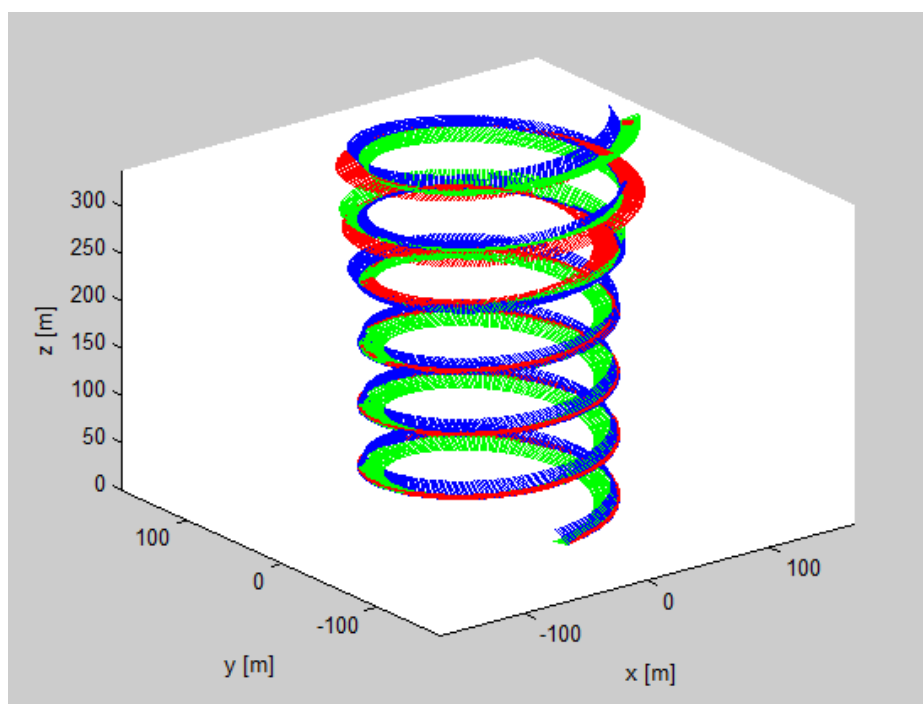


Figura 14.12 - Resultado 3D - 200s

Apesar da divergência numérica ocorrer apenas após períodos de simulação longos em comparação com o período típico de convergência esperado para o sistema, poucos segundos, é interessante que se programe uma rotina de ortonormalização da base de K'''' em paralelo com a integração para minimizar erros.

Por outro lado, quanto à velocidade de processamento, o resultado foi altamente satisfatório, pois as simulações demoram menos de dois segundos quando realizadas em um PC 2GHz/2Gb RAM.

15. Gram-Schmidt e Runge-Kutta

Uma vez que as funções de integração testadas (*ode45* e *ode113*) com o modo automático de limitação de erro, não mostram-se capazes de manter-se estáveis sob longos períodos de integração, surge a necessidade de se exercer mais controle sobre o processo de integração.

A principal fonte de erro mostra-se ser a norma e direção dos versores da base integrada. Propriedades oriundas da ortonormalidade da base foram usadas na dedução das equações. Portanto, é imprescindível o cuidado com o acúmulo de erros nos versores.

Dessa maneira, a inserção do algoritmo de ortonormalização de Gram-Schmidt na integração pode ser utilizada para garantir a ortogonalidade da base após cada passo de integração.

A ordem em que o algoritmo é aplicado deve ser aleatória para que não se privilegie nenhuma direção.

Para a implementação da ortonormalização as próprias funções *ode* do MALTAB poderiam ser editadas. No entanto, uma função específica para o programa foi implementada utilizando o algoritmo de Runge-Kutta de quarta ordem juntamente à rotina de Gram-Schmidt.

Além da vantagem pedagógica, o fato de Runge-Kutta integrar com passo temporal fixo trará vantagens posteriores na integração em tempo real. Uma vez que o algoritmo da integração é significativamente leve, o passo pode ser mantido refinado à ordem do centésimo de segundo sem custos computacionais maiores do que aqueles já impostos pelas funções *ode45* e *ode113*.

De fato, a nova função de integração/ortonormalização garante maior estabilidade e confiabilidade do resultado, conforme exemplificado abaixo.

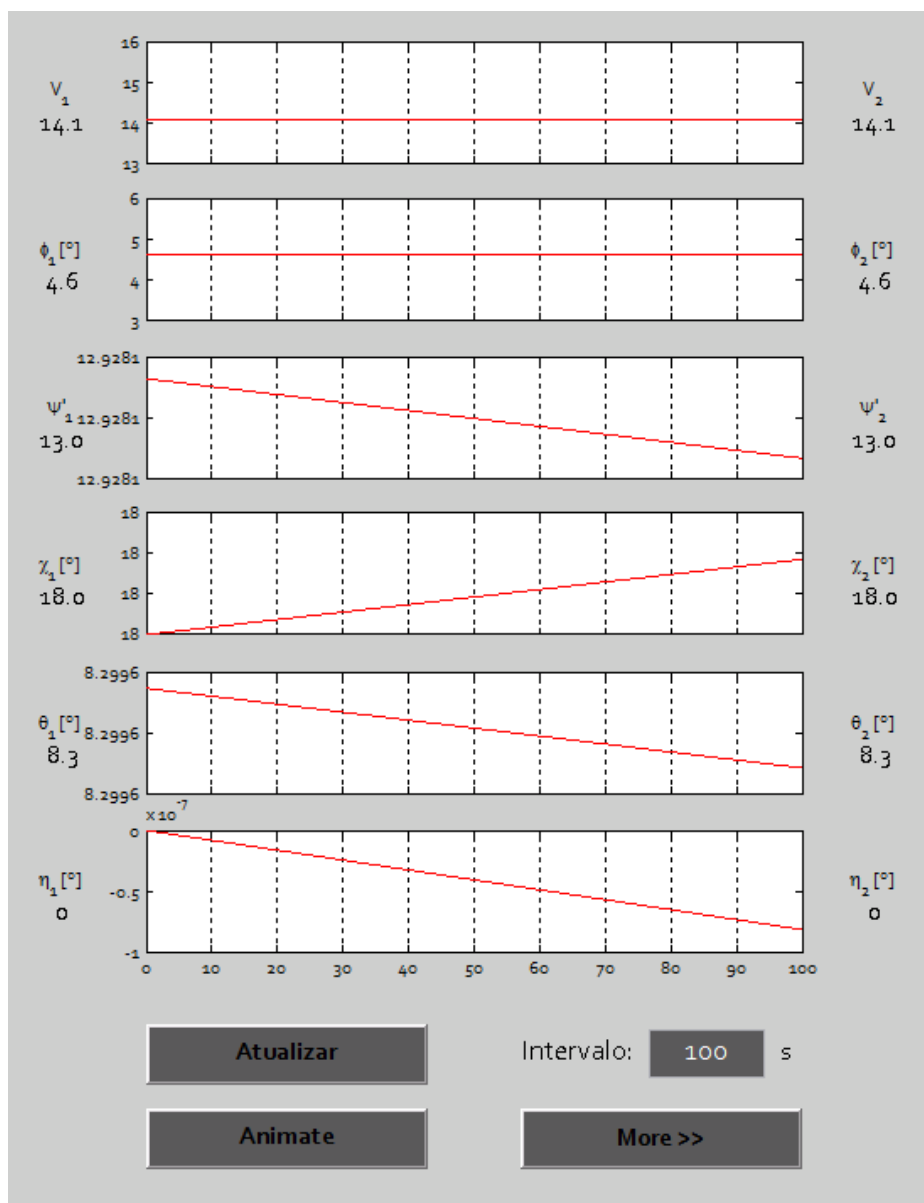


Figura 15.1 - Estabilidade numérica

16. Gimble-lock

A técnica das matrizes de rotações finitas não é infalível. Dependendo da ordem estabelecida é possível que não haja uma combinação de rotações para atingir a posição desejada. Para evitar esse problema há outras técnicas como o uso de quatérnos que não possuem tal deficiência.

Por outro lado, para o conjunto de rotações adotado no capítulo 13, o travamento, ou *Gimble-lock*, só ocorre para $\theta = \pm 90^\circ$. Uma vez que o domínio dos modelos de coeficientes aerodinâmicos exclui ângulos de ataque tão extremos quanto $\pm 90^\circ$, nunca deve-se chegar ao ponto do travamento.

Contudo, durante a execução do simulador é possível que se passe rapidamente por essa condição. Para esse caso, uma rotina lógica força a utilização dos dados dos ângulos de rotação do instante “imediatamente” anterior para a integração e geração dos gráficos até que a base numérica evolua suficientemente para que ela saia dessa condição.

Essa maneira de contornar o problema mostra-se eficiente pois impede falha do integrador devido a indeterminações e permite que o processo continue mesmo fatalmente passando sobre o “furo” no domínio.

17. Aprimoramentos do Modelo

Como discutido no capítulo 10, há a necessidade de que o modelo se aperfeiçoe. Neste estágio do projeto, questões operacionais cruciais já foram abordadas e resolvidas. Logo, direcionam-se os esforços para polir o programa e aumentar sua capacidade e versatilidade.

17.1. Auto-trim

A boa definição do ponto central do domínio de operação do profundor garante a pertinência da linearização de seu arrasto induzido. Para que esse processo não precise ser executado pelo usuário após qualquer alteração de configuração da aeronave, uma rotina que busca esse ponto foi implementada dentro da rotina de regimes de voo.

Dessa maneira, o valor de α_{L0p} é atualizado automaticamente todas as vezes que os regimes permanentes são calculados.

Isso é possível pois a localização de α_{L0p} é um processo de trimagem do voo 2D e possui solução analítica, pois trata-se de um polinômio do segundo grau.

Nos eixos denominados *Monitor* na função de regimes de voo é possível ver a comparação em vermelho e preto tracejado, respectivamente, dos valores de CD_{ht} adotados após a linearização dos que seriam obtidos pela utilização do arrasto induzido quadrático.

A variável *auto_trim* habilita a sub-rotina de trimagem. Caso deseje-se realizá-la manualmente, o valor zero deve ser atribuído a *auto_trim* pela linha de comando do *main*.

17.2. Stall da Asa

É possível importar dados de curvas quaisquer para o CL_w e CD_w . De tal forma que um *fit* polinomial forneça funções para esses coeficientes dentro do domínio $\pm 90^\circ$.

Ao incluir essa variação as simulações tornam-se mais fidedignas para situações de altas taxas de subida e manobras mais intensas.

17.3. Dutch-Roll

O acoplamento Guinada-Rolagem é um comportamento característico de aeronaves e até o momento tem sido ignorado.

Oriundos principalmente da asa, momentos de guinada e rolagem surgem alternadamente devido à uma manobra e a um perfil não simétrico de velocidade induzido por uma curva por exemplo, de tal forma que uma oscilação instável pode levar a queda da aeronave.

Naturalmente, o estudo desse comportamento pertence ao escopo do trabalho e deve ser modelado.

Para a quantificação da não-simetria do escoamento, dois coeficientes adimensionais foram propostos, de forma a indicar a intensidade da alteração da velocidade aparente nas duas “meias” asas.

Um coeficiente indica a velocidade induzida pela rolagem no sentido de alterar o ângulo efetivo de ataque ao longo da envergadura adimensionalizado pela velocidade aparente na raiz da asa, e o outro analogamente representa a velocidade aparente de promovida pelo movimento de guina no sentido de intensificar ou reduzir a velocidade aparente.

É imprescindível que tais velocidades induzidas pela rotação sejam calculadas em função das coordenadas da velocidade aparente total na base K''' . A base K''' é aquela que se alinha com o escoamento e o horizonte lateralmente inclinado da aeronave. Logo, ao utilizar essa base os coeficientes passam a representar perturbações nos ângulos efetivos de ataque, mantendo uma referência sólida do escoamento não perturbado, o que garante estabilidade do modelo e facilidade não obtenção de dados.

$$(17.1) \quad f_M := - \frac{\left(\Omega_{kiii} \&x \begin{bmatrix} 0 \\ \frac{bw}{2} \\ 0 \end{bmatrix} \right)}{V}$$

Se o vetor rotação for obtido a partir da base numérica, as rotações em θ e η devem ser “desfeitas” para chegar-se na base K”.

$$(17.2) \quad \Omega_{kiii} := \begin{bmatrix} \cos(\eta) \cos(\theta) \alpha \dot{\omega} - \sin(\eta) \cos(\theta) \beta \dot{\omega} - \sin(\theta) \gamma \dot{\omega} \\ \sin(\eta) \alpha \dot{\omega} + \cos(\eta) \beta \dot{\omega} \\ \sin(\theta) \cos(\eta) \alpha \dot{\omega} - \sin(\theta) \sin(\eta) \beta \dot{\omega} + \cos(\theta) \gamma \dot{\omega} \end{bmatrix}$$

E, nesse caso os coeficientes assumem a seguinte forma:

$$(17.3) \quad f_M := \begin{bmatrix} \frac{1}{2} \frac{(\sin(\theta) \cos(\eta) \alpha \dot{\omega} - \sin(\theta) \sin(\eta) \beta \dot{\omega} + \cos(\theta) \gamma \dot{\omega}) b_w}{V} \\ 0 \\ -\frac{1}{2} \frac{(\cos(\eta) \cos(\theta) \alpha \dot{\omega} - \sin(\eta) \cos(\theta) \beta \dot{\omega} - \sin(\theta) \gamma \dot{\omega}) b_w}{V} \end{bmatrix}$$

Por outro lado, na obtenção do regime permanente o vetor rotação é conhecido na base K. Portanto, as rotações em ψ , ϕ , χ devem ser aplicadas para a mudança de base.

$$(17.4) \quad \Omega_{kiii} := \begin{bmatrix} \sin(\phi) \psi \dot{\omega} \\ -\sin(\chi) \cos(\phi) \psi \dot{\omega} \\ \cos(\chi) \cos(\phi) \psi \dot{\omega} \end{bmatrix}$$

Lembrando da relação cinemática:

$$(17.5) \quad \psi \dot{\omega} := \frac{g \tan(\chi)}{V}$$

Os coeficientes têm nesse caso outra expressão, mas devem ser idênticos no regime permanente.

$$(17.6) \quad f_M := \begin{bmatrix} \frac{1}{2} \frac{\cos(\chi) \cos(\phi) g \tan(\chi) b_w}{V^2} \\ 0 \\ -\frac{1}{2} \frac{\sin(\phi) g \tan(\chi) b_w}{V^2} \end{bmatrix}$$

A partir dos coeficientes f_M , os coeficientes totais de momento de rolagem e guinada na asa devem ser calculados.

Para isso, utiliza-se a seguinte fórmula.

$$(17.7) \quad \begin{bmatrix} CMr_w \\ CM_y_w \end{bmatrix} := \begin{bmatrix} CM_{rr} & CM_{ry} \\ CM_{yr} & CM_{yy} \end{bmatrix} \cdot \begin{bmatrix} f_M[3] \\ f_M[1] \end{bmatrix} + \begin{bmatrix} CMr\delta_w \\ CM_y\delta_w \end{bmatrix} \cdot \delta_w$$

Onde a matriz 2X2 exerce o acoplamento e a ponderação. Momento de rolagem induzido por velocidade de rolagem (o que deve ser negativo, ou seja, restituidor e estável), assim como momento de guinada induzido por velocidade de guinada (também restituidor e estável). Em contrapartida, a diagonal secundária é positiva e exerce influência divergente.

Dessa forma, o comportamento *Dutch-Roll* é incluído no modelo. Exemplos da influência são claros em simulações e serão expostos na apresentação final do trabalho.

17.4. Momentos da Propulsão

Utilizar o desalinhamento da linha de ação das forças de propulsão do centro de massa era uma sofisticação do modelo desejada desde o início do projeto.

Contudo, tal versatilidade no modelo implica em termos dependentes do inverso da velocidade nas equações de momentos e foi por isso até então evitada.

Felizmente, o método de solução de busca vetorizada de φ , não é prejudicado por esse acoplamento algébrico. Diferentemente do método do capítulo 8, em que isso era proibitivo.

Logo, a possibilidade de alterar a direção da resultante de propulsão foi programada sem aumento de custo computacional, pois o próprio método solução por busca já tornara-se o gargalo da velocidade de processamento.

Essa modificação expandiu significativamente os horizontes de projeto, pois com essa “nova fonte” de momentos, otimizações para o uso da empenagem podem ser realizadas de forma reduzir sua atuação, diminuindo arrasto, ou até inverter seu regime de sustentação, passando a contribuir positivamente na ascensão da aeronave.

18. Real-time e Joystick

Para a finalização do trabalho e o comprimento total da proposta inicial, simulação em tempo real e com entrada dinâmica do usuário também devem ser opções oferecidas pelo programa.

18.1. Implementação

Um Joystick com entrada USB, semelhante ao utilizado em *video games* da *Sony* foi empregado como interface.

No sistema operacional *Windows Vista* tal dispositivo é *plug and play* e é automaticamente reconhecido pelo MATLAB.

Dessa forma, não houve dificuldade alguma de configuração e compatibilidade. Uma função do MATLAB lê quando o for solicitado os estados dos botões e alavancas do Joystick.

Inserindo esse comando no código de integração, permite-se a entrada dinâmica.

Além disso, os cálculos devem ser realizados em tempo real. Para isso, um *delay* regido pelo comando *tic toc* atrasa o avanço para o próximo *step* da integração e garante o sincronismo das contas com o “relógio”.

A atualização dos gráficos não precisa ser feita para todos os pontos da discretização, mas sim apenas para um divisor inteiro. Uma vez que é justamente a saída gráfica que consome tempo, pode-se manter o Δt de integração $\approx 1/32$ s, atualizando a tela apenas a cada $1/8$ s que os cálculos ainda andam mais rápido que o tempo real e a função *tic toc* é capaz de manter o “clock”.

Obviamente, computadores mais potentes que um notebook pessoal de 2GHz/2Gb não terão problemas em refinar ainda mais as contas.

18.2. Exemplos

O relógio na canto inferior direito da tela indica o instante em que a simulação se encontra. Se a indicação estiver em vermelho, os cálculos estão consumindo muito tempo e o

tempo real não pode ser mantido. Por outro lado, a cor preta indica a existência de folga e a manutenção do ritmo adequado.

O gráfico na base da tela mostra os comandos.

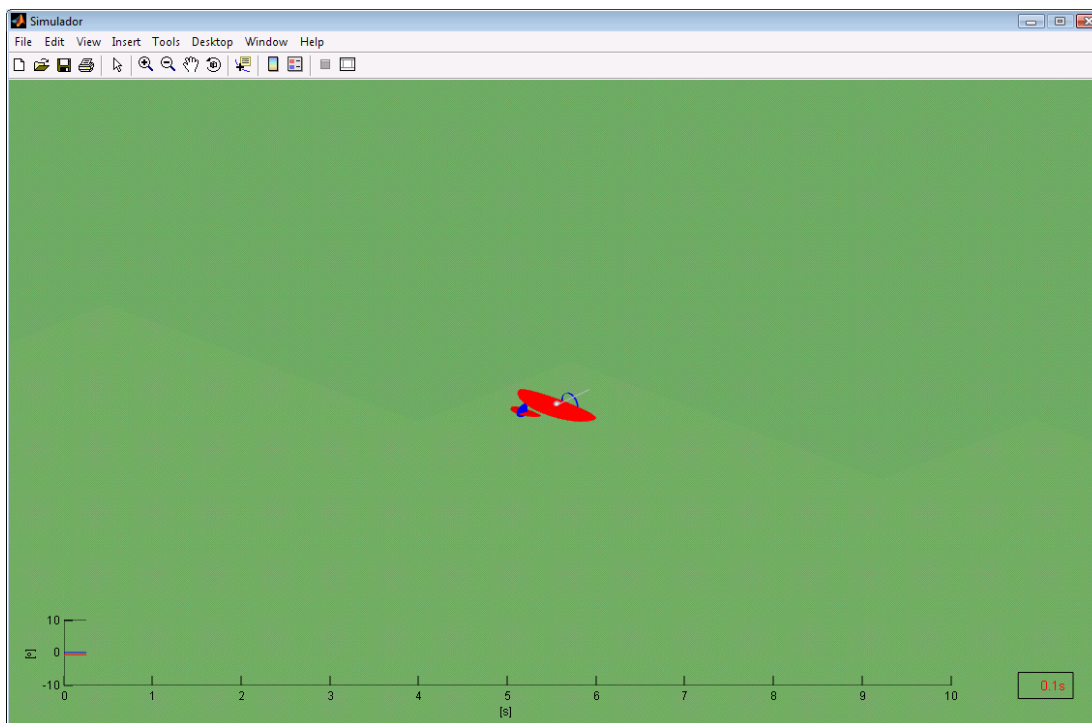


Figura 18.1 - Simulador

O CD encontrado no Anexo IV contém o programa e um arquivo *txt* com explicações para a execução.

Alguns exemplos foram preparados e podem ser carregados com o botão "Load" para sua exploração.

19. Conclusão

Diferentemente da abordagem tradicional, em que se utilizam coeficientes oriundos de linearizações das equações do movimento e um método de resolução da dinâmica por inversão de matrizes [8], o método aqui discutido preza por evitar resumir o sistema de equações a tal condição.

As derivadas que surgem pela forma tradicional são compostas pela influência de todas as superfícies e normalmente são obtidas experimentalmente. Caso haja grande disponibilidade de recursos para ensaios em túnel de vento, não há nenhum motivo pelo qual essa abordagem seja menosprezada.

Por outro lado, ensaios em que se leva toda a aeronave, parte dela ou um modelo em escala ao túnel de vento são sabidamente custosos e requerem aparelhagem específica. Além disso, dessa forma, perde-se uma parte significativa da sensibilidade de projeto, uma vez que não se avaliam influências independentemente, mas apenas o conjunto. Por exemplo, as derivadas de momento passam a mascarar todas as relações de braços inerentes da aeronave.

Uma visão mais pragmática avalia superfície sustentadora a superfície e propõe-se a compor o todo, conhecendo bem os elementos. Obviamente, para esse segunda abordagem efeitos de interferência no escoamento tornam-se empecilhos e devem ser cuidadosamente contornados.

Portanto, um equilíbrio entre dados experimentais ou simulações de interferência e um modelo mecânico de composição explícita de forças e momentos aerodinâmicos para cada superfície sustentadora deve ser encontrado pelo compromisso das duas diferentes visões.

A abordagem aqui proposta ataca o problema de uma forma particular. Espera-se que a metodologia desenvolvida tenha permitido ao leitor compreender o encadeamento lógico a ser empregado na estudo de trajetórias de uma aeronave, e que o método tenha sido elaborado de forma suficientemente genérica e bem argumentada para que um projetista da aeronaves possa basear-se neste trabalho e empregá-lo adequadamente.

Na vertente de segurança e conforto de vôo, uma ferramenta capaz de prever o comportamento da aeronave com acuracidade aumenta a capacidade de controle do sistema, tanto para um controlador quanto para uma definição prévia de trajetória.

Para um projeto de planador, por exemplo, em que se deseje maximizar autonomia de voo a ferramenta desenvolvida é altamente adequada.

É importante ressaltar que o desenvolvimento do trabalho seguiu diretrizes do estudo da mecânica de corpo rígido. Sem dúvida é esse o ponto de partida para a análise do voo. Obviamente, esse fenômeno pertence a um universo mais amplo que compreende a mecânica dos fluídos e sólidos deformáveis.

Logo, quanto à complexidade do modelo, atingiram-se dois limites, um imposto pela mecânica de corpo rígido, que elimina comportamentos aero-elásticos e outro pelo acoplamento fluído-mecânico que gera a necessidade da resolução do escoamento simultaneamente.

Um próximo passo para o aprofundamento no estudo da dinâmica de voo, além da consideração de deformações da aeronave, seria equacionar o escoamento (da asa pelo menos) e resolver todo sistema em paralelo.

Problemas operacionais podem surgir. Acoplamentos nessa formulação poderiam inviabilizar o isolamento das acelerações para a integração convencional. Além disso as técnicas de elementos finitos ou mesmo diferenças finitas seriam muitas “pesadas” e, portanto, não adequadas para a resolução do escoamento.

Especula-se que uma generalização da teoria de linhas de Sustentação de Prandtl para um escoamento potencial com a inclusão de termos temporais, auxiliado pelos estudos de vorticidade de Helmholtz será um grande avanço científico da área.

O formando propõe-se a continuar sua pesquisa no tema e espera que a Escola Politécnica, apesar de não possuir um departamento de aeronáutica, possa incentivar a continuação desse desenvolvimento.

20. Bibliografia

- [1] - Bruce R. Munson, Donald F. Young E Theodore H. Okiishi, "FUNDAMENTOS DA MECÂNICA DOS FLUIDOS", Editora Edgard Blücher
- [2] - Durrant, W.H. "Fluid Mechanics and Aircraft Design", Collection I to VII
- [3] - John D. Anderson, Jr., "Introduction to flight", McCraw-Hill Companies, Inc., 4-th ed., Boston, 1999
- [4] - Roskam, Jan, "Airplan Design Part - VI Preliminary Calculation of Aerodynamic, Thust and Power Characteristics"
- [5] - Raymer, Daniel P., "Aircraft Design" Fourth Edition, AIAA Education Series, New York, NY, 2006
- [6] - Nelson, Robert C. – Flight Stability and Automatic Control
- [7] - Miele, Angelo – Flight Mechanics Vol.1
- [8] - Phillips, Warren F. – Mechanics of Flight, 2004
- [9] - Thomson, Willian Tyrrell – Introduction to Space Dynamics, 1963
- [10] - Beer Johnston, "Vector Mechancis For Engineers: Statics And Dynamics"
- [11] - Keep Flying da EPUSP – SEA Aerodesign

ANEXO I - Desenvolvimento em Maple

```

Mkik := 
$$\begin{bmatrix} \cos(\phi) \cos(\psi) & \sin(\psi) \cos(\phi) & \sin(\phi) \\ -\sin(\psi) & \cos(\psi) & 0 \\ -\cos(\psi) \sin(\phi) & -\sin(\psi) \sin(\phi) & \cos(\phi) \end{bmatrix}$$

>

Mkki := Mkik%T

> check := simplify(Mkik.Mkki)

check := 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$


Mkiiki := 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\chi) & -\sin(\chi) \\ 0 & \sin(\chi) & \cos(\chi) \end{bmatrix}$$

>

Mkikii := Mkiiki%T

Mkiikii := 
$$\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

>

Mkiikiii := Mkiikii%T

> check := simplify(Mkiikiii.Mkiikii)

check := 
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$


> Mkiik := simplify(Mkiikiii.Mkiiki.Mkik)

Mkiik := 
$$\begin{bmatrix} \cos(\theta) \cos(\phi) \cos(\psi) - \sin(\theta) \sin(\chi) \sin(\psi) - \sin(\theta) \cos(\chi) \cos(\psi) \sin(\phi), \\ \cos(\theta) \sin(\psi) \cos(\phi) + \sin(\theta) \sin(\chi) \cos(\psi) - \sin(\theta) \cos(\chi) \sin(\psi) \sin(\phi), \\ \cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi) ], \\ [-\cos(\chi) \sin(\psi) + \sin(\chi) \cos(\psi) \sin(\phi), \cos(\chi) \cos(\psi) + \sin(\chi) \sin(\psi) \sin(\phi), \\ -\sin(\chi) \cos(\phi) ], \\ [-\sin(\theta) \cos(\phi) \cos(\psi) - \cos(\theta) \sin(\chi) \sin(\psi) - \cos(\theta) \cos(\chi) \cos(\psi) \sin(\phi), \\ -\sin(\theta) \sin(\psi) \cos(\phi) + \cos(\theta) \sin(\chi) \cos(\psi) - \cos(\theta) \cos(\chi) \sin(\psi) \sin(\phi), \\ -\sin(\theta) \sin(\phi) + \cos(\theta) \cos(\chi) \cos(\phi) ] \end{bmatrix}$$


> Mkiikiii := Mkiik%T

> Mkiiki := simplify(Mkiikiii.Mkiiki)

Mkiiki := 
$$\begin{bmatrix} \cos(\theta) & \sin(\theta) \sin(\chi) & \sin(\theta) \cos(\chi) \\ 0 & \cos(\chi) & -\sin(\chi) \\ -\sin(\theta) & \cos(\theta) \sin(\chi) & \cos(\theta) \cos(\chi) \end{bmatrix}$$


> Mkiikiii := Mkiiki%T

```

> *check* := *simplify* (*Mkiiik.Mkiii*)

$$check := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> *Mkiiik* := *simplify* (*Mkiki.Mkik*)

$$Mkiiik := \begin{bmatrix} \cos(\phi) \cos(\psi), \sin(\psi) \cos(\phi), \sin(\phi) \\ [-\cos(\chi) \sin(\psi) + \sin(\chi) \cos(\psi) \sin(\phi), \cos(\chi) \cos(\psi) + \sin(\chi) \sin(\psi) \sin(\phi), \\ -\sin(\chi) \cos(\phi)] \\ [-\sin(\chi) \sin(\psi) - \cos(\chi) \cos(\psi) \sin(\phi), \sin(\chi) \cos(\psi) - \cos(\chi) \sin(\psi) \sin(\phi), \\ \cos(\chi) \cos(\phi)] \end{bmatrix}$$

> *Mkiiikiii* := *simplify* (*Mkiiik.Mkiii*)

$$Mkiiikiii := \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$\Omega\psi k := \begin{bmatrix} 0 \\ 0 \\ \psi i \end{bmatrix}$$

>

$$\Omega\psi k := \begin{bmatrix} 0 \\ 0 \\ \psi i \end{bmatrix}$$

> $\Omega\psi kiii := Mkiiik.\Omega\psi k$

$$\Omega\psi kiii := \begin{bmatrix} (\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \psi i \\ -\sin(\chi) \cos(\phi) \psi i \\ (-\sin(\theta) \sin(\phi) + \cos(\theta) \cos(\chi) \cos(\phi)) \psi i \end{bmatrix}$$

$$Jkiiikiii := \begin{bmatrix} J1 & 0 & 0 \\ 0 & J2 & 0 \\ 0 & 0 & J3 \end{bmatrix}$$

>

$$Mkikponto := \psi i \cdot \begin{bmatrix} -\cos(\phi) \sin(\psi) & \cos(\psi) \cos(\phi) & 0 \\ -\cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) \sin(\phi) & -\cos(\psi) \sin(\phi) & 0 \end{bmatrix}$$

>

> *Mkiiikponto* := *Mkiiikii.Mkiki.Mkikponto*

> *Mkkiponto* := *Mkikponto*^{%T}

> *Mkiiiponto* := *Mkkiponto.Mkiki.Mkiiikiii*

> *check* := *simplify* (*Mkiiiponto*^{%T} - *Mkiiikponto*)

$$check := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

> *Hkiii* := *Jkiiikiii.* $\Omega\psi kiii$

$$Hkiii := \begin{bmatrix} J1 (\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \psi i \\ -J2 \sin(\chi) \cos(\phi) \psi i \\ J3 (-\sin(\theta) \sin(\phi) + \cos(\theta) \cos(\chi) \cos(\phi)) \psi i \end{bmatrix}$$

$$> \text{termo1} := \text{simplify}(Mkiiik.Mkiiiponto.Hkiii)$$

$$> \text{with}(LinearAlgebra) :$$

$$> \text{termo2} := \text{simplify}(\Omega \psi kiii \&x Hkiii)$$

$$\begin{aligned} \text{termo2} := & \left[\begin{aligned} & -\sin(\chi) \cos(\phi) \psi i^2 (\sin(\theta) \sin(\phi) - \cos(\theta) \cos(\chi) \cos(\phi)) (-J3 + J2), \\ & \psi i^2 (J1 - J3) (-\sin(\theta) \cos(\theta) + \sin(\theta) \cos(\theta) \cos(\phi)^2 - \sin(\phi) \cos(\chi) \cos(\phi) \\ & + 2 \cos(\phi) \cos(\chi) \cos(\theta)^2 \sin(\phi) + \cos(\phi)^2 \cos(\chi)^2 \cos(\theta) \sin(\theta)) \end{aligned} \right], \\ & \left[(\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \psi i^2 \sin(\chi) \cos(\phi) (-J2 + J1) \right] \end{aligned}$$

$$> \text{check} := \text{simplify}(\text{termo1} - \text{termo2})$$

$$\text{check} := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$> \text{dwkii} := \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{dwkii} := \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$> \text{lwki} := \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \&x \begin{pmatrix} Mkiiikii. \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{pmatrix}$$

$$\text{lwki} := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$> \text{Lwkiii} := \text{Lw} \cdot (Mkiiikii.lwki)$$

$$\text{Lwkiii} := \begin{bmatrix} \text{Lw} \sin(\theta) \\ 0 \\ \text{Lw} \cos(\theta) \end{bmatrix}$$

$$> \text{Dwkiii} := \text{Dw} \cdot (Mkiiikii.dwkii)$$

$$\text{Dwkiii} := \begin{bmatrix} -\text{Dw} \cos(\theta) \\ 0 \\ \text{Dw} \sin(\theta) \end{bmatrix}$$

$$> \text{Mwkiii} := \begin{bmatrix} -\text{Mrw} \\ -\text{Mpw} \\ \text{Myw} \end{bmatrix}$$

$$\text{Mwkiii} := \begin{bmatrix} -\text{Mrw} \\ -\text{Mpw} \\ \text{Myw} \end{bmatrix}$$

$$> \quad MLwkiii := combine \left(\begin{bmatrix} -lw \cdot \cos(\gamma w) \\ 0 \\ lw \cdot \sin(\gamma w) \end{bmatrix} \&x Lwkiii \right)$$

$$MLwkiii := \begin{bmatrix} 0 \\ lw \cdot Lw \cos(-\gamma w + \theta) \\ 0 \end{bmatrix}$$

$$> \quad MDwkiii := combine \left(\begin{bmatrix} -lw \cdot \cos(\gamma w) \\ 0 \\ lw \cdot \sin(\gamma w) \end{bmatrix} \&x Dwkiii \right)$$

$$MDwkiii := \begin{bmatrix} 0 \\ lw \cdot Dw \sin(-\gamma w + \theta) \\ 0 \end{bmatrix}$$

$$> \quad Mkiiikii := \begin{bmatrix} \cos(\epsilon) & 0 & \sin(\epsilon) \\ 0 & 1 & 0 \\ -\sin(\epsilon) & 0 & \cos(\epsilon) \end{bmatrix}$$

$$> \quad Mkiikiii := Mkiiikii^{9/T}$$

$$> \quad check := simplify(Mkiikiii.Mkiiikii)$$

$$check := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$> \quad dhtkiii := combine \left(Mkiiikii.Mkiiikiii, \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$dhtkiii := \begin{bmatrix} -\cos(-\theta + \epsilon) \\ 0 \\ -\sin(-\theta + \epsilon) \end{bmatrix}$$

$$> \quad lhtkiii := combine \left(Mkiiikii.Mkiiikiii, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$

$$lhtkiii := \begin{bmatrix} -\sin(-\theta + \epsilon) \\ 0 \\ \cos(-\theta + \epsilon) \end{bmatrix}$$

$$> \quad Lhtkiii := Lht \cdot lhtkiii$$

$$Lhtkiii := \begin{bmatrix} -Lht \sin(-\theta + \epsilon) \\ 0 \\ Lht \cos(-\theta + \epsilon) \end{bmatrix}$$

$$> \quad Dhtkiii := Dht \cdot dhtkiii$$

$$Dhtkiii := \begin{bmatrix} -Dht \cos(-\theta + \epsilon) \\ 0 \\ -Dht \sin(-\theta + \epsilon) \end{bmatrix}$$

$$> \quad Mhtkiii := \begin{bmatrix} -Mrht \\ -Mpht \\ Myht \end{bmatrix}$$

$$Mhtkiii := \begin{bmatrix} -Mrht \\ -Mpht \\ Myht \end{bmatrix}$$

$$> MLhtkiii := combine \left(\begin{bmatrix} -lht \cdot \cos(\gamma ht) \\ 0 \\ lht \cdot \sin(\gamma ht) \end{bmatrix} \&x Lhtkiii \right)$$

$$MLhtkiii := \begin{bmatrix} 0 \\ lht \ Lht \cos(\gamma ht - \theta + \varepsilon) \\ 0 \end{bmatrix}$$

$$> MDhtkiii := combine \left(\begin{bmatrix} -lht \cdot \cos(\gamma ht) \\ 0 \\ lht \cdot \sin(\gamma ht) \end{bmatrix} \&x Dhtkiii \right)$$

$$MDhtkiii := \begin{bmatrix} 0 \\ -lht \ Dht \sin(\gamma ht - \theta + \varepsilon) \\ 0 \end{bmatrix}$$

$$> lvtkiii := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$lvtkiii := \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$> dvtkiii := combine \left(Mkiiikii.Mkiiikiii, \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \right)$$

$$dvtkiii := \begin{bmatrix} -\cos(-\theta + \varepsilon) \\ 0 \\ -\sin(-\theta + \varepsilon) \end{bmatrix}$$

$$> Lvtkiii := Lvt \cdot lvtkiii$$

$$Lvtkiii := \begin{bmatrix} 0 \\ Lvt \\ 0 \end{bmatrix}$$

$$> Dvtkiii := Dvt \cdot dvtkiii$$

$$Dvtkiii := \begin{bmatrix} -Dvt \cos(-\theta + \varepsilon) \\ 0 \\ -Dvt \sin(-\theta + \varepsilon) \end{bmatrix}$$

$$> Mvtkiii := \begin{bmatrix} -Mrvt \\ -Myvt \\ Mpvt \end{bmatrix}$$

$$Mvtkiii := \begin{bmatrix} -Mrvt \\ -Myvt \\ Mpvt \end{bmatrix}$$

$$> MLvtkiii := combine \left(\begin{bmatrix} -lvt \cdot \cos(\gamma vt) \\ 0 \\ lvt \cdot \sin(\gamma vt) \end{bmatrix} \&x Lvtkiii \right)$$

$$MLvtkii := \begin{bmatrix} -lvt \sin(\gamma t) \ Lvt \\ 0 \\ -lvt \cos(\gamma t) \ Lvt \end{bmatrix}$$

$$> MDvtkiii := combine \left(\begin{bmatrix} -lvt \cdot \cos(\gamma t) \\ 0 \\ lvt \cdot \sin(\gamma t) \end{bmatrix} \&x Dvtkiii \right)$$

$$MDvtkiii := \begin{bmatrix} 0 \\ -lvt \ Dvt \sin(\gamma t - \theta + \varepsilon) \\ 0 \end{bmatrix}$$

$$> dfkii := \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$dfkii := \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$> lfkii := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$lfkii := \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$> Lfkiii := Lf \cdot (Mkiikii.lfkii)$$

$$Lfkiii := \begin{bmatrix} Lf \sin(\theta) \\ 0 \\ Lf \cos(\theta) \end{bmatrix}$$

$$> Dfkiii := Df \cdot (Mkiikii.dfki)$$

$$Dfkiii := \begin{bmatrix} -Df \cos(\theta) \\ 0 \\ Df \sin(\theta) \end{bmatrix}$$

$$> Mfkiii := \begin{bmatrix} -Mrf \\ -Mpf \\ Myf \end{bmatrix}$$

$$Mfkiii := \begin{bmatrix} -Mrf \\ -Mpf \\ Myf \end{bmatrix}$$

$$> MLfkiii := combine \left(\begin{bmatrix} -lf \cdot \cos(\gamma) \\ 0 \\ lf \cdot \sin(\gamma) \end{bmatrix} \&x Lfkiii \right)$$

$$MLfkiii := \begin{bmatrix} 0 \\ lf \ Lf \cos(-\gamma + \theta) \\ 0 \end{bmatrix}$$

$$> MDfkiii := combine \left(\begin{bmatrix} -lf \cdot \cos(\gamma) \\ 0 \\ lf \cdot \sin(\gamma) \end{bmatrix} \&x Dfkiii \right)$$

$$MDfkiii := \begin{bmatrix} 0 \\ lf \, Df \sin(-\gamma f + \theta) \\ 0 \end{bmatrix}$$

$$> \quad \varepsilon := 0$$

$$\varepsilon := 0$$

$$> \quad T := x \rightarrow a_2 x^2 + a_1 x + a_0$$

$$T := x \rightarrow a_2 x^2 + a_1 x + a_0$$

$$> \quad TMA_e := termo2$$

$$\begin{aligned} TMA_e := & \left[\left[-\sin(\chi) \cos(\phi) \, \psi t^2 (\sin(\theta) \sin(\phi) - \cos(\theta) \cos(\chi) \cos(\phi)) (-J3 + J2) \right], \right. \\ & \left[\psi t^2 (J1 - J3) (-\sin(\theta) \cos(\theta) + \sin(\theta) \cos(\theta) \cos(\phi)^2 - \sin(\phi) \cos(\chi) \cos(\phi) \right. \\ & \quad \left. + 2 \cos(\phi) \cos(\chi) \cos(\theta)^2 \sin(\phi) + \cos(\phi)^2 \cos(\chi)^2 \cos(\theta) \sin(\theta)) \right], \\ & \left. \left[(\cos(\theta) \sin(\phi) + \sin(\theta) \cos(\chi) \cos(\phi)) \, \psi t^2 \sin(\chi) \cos(\phi) (-J2 + J1) \right] \right] \end{aligned}$$

$$\begin{aligned} TMA_d := & collect(MLwkiii + MDwkiii + Mwkiii + MLhtkiii + MDhtkiii + Mhtkiii + MLvtkiii \\ & + MDvtkiii + Mvtkiii + MLfkiii + MDfkiii + Mfkiii, \{V^2, \rho\}) \\ > \end{aligned}$$

$$\begin{aligned} TMA_d := & \left[\left[-Mrw - Mrht - lvt \sin(\gamma t) \, Lvt - Mrvt - Mrf \right], \right. \\ & \left[lw \, Lw \cos(-\gamma w + \theta) + lw \, Dw \sin(-\gamma w + \theta) - Mpw + lht \, Lht \cos(-\gamma ht + \theta) \right. \\ & \quad \left. + lht \, Dht \sin(-\gamma ht + \theta) - Mpht + lvt \, Dvt \sin(-\gamma t + \theta) - Myvt + lf \, Lf \cos(-\gamma f + \theta) \right. \\ & \quad \left. + lf \, Df \sin(-\gamma f + \theta) - Mpf \right], \\ & \left. \left[Myw + Myht - lvt \cos(\gamma t) \, Lvt + Mpvt + Myf \right] \right] \end{aligned}$$

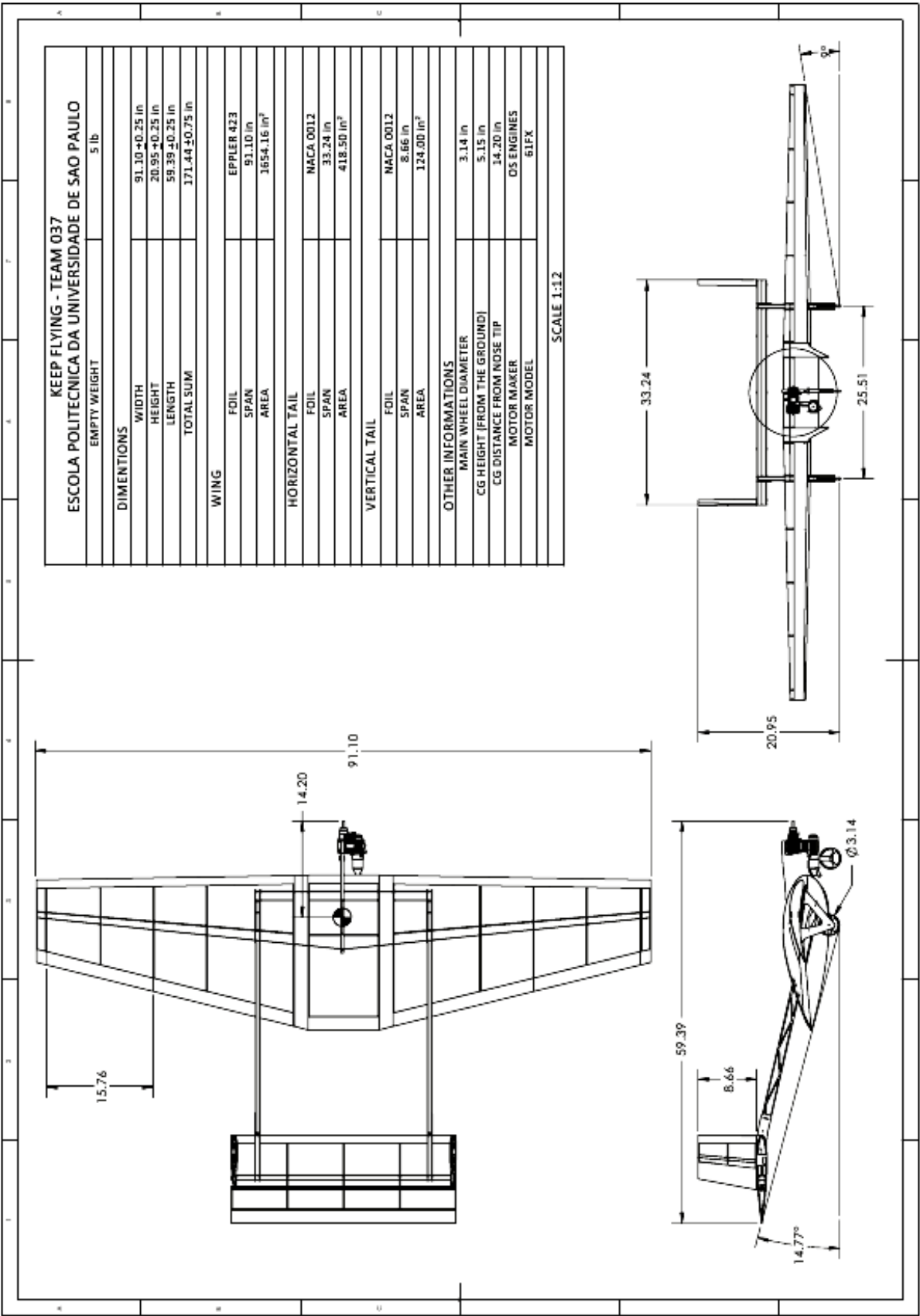
$$> \quad TMB_e := Mkiik. \begin{bmatrix} 0 \\ m \cdot \psi i \cdot V \cdot \cos(\phi) \\ 0 \end{bmatrix}$$

$$TMB_e := \begin{bmatrix} 0 \\ \cos(\chi) \, m \, \psi i \, V \cos(\phi) \\ \sin(\chi) \, m \, \psi i \, V \cos(\phi) \end{bmatrix}$$

$$\begin{aligned} TMB_d := & combine(Mkiikiii, (Lwkiii + Dwkiii + Lhtkiii + Dhtkiii + Lvtkiii + Dvtkiii + Lfkiii \\ & + Dfkiii)) + T \left(V \cdot combine \left(\begin{bmatrix} \cos(\gamma m) & 0 & \sin(\gamma m) \end{bmatrix} . Mkiikii. \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) \right) \cdot combine \left(Mkiikiii \right. \\ & \left. \cdot \begin{bmatrix} \cos(\gamma m) \\ 0 \\ \sin(\gamma m) \end{bmatrix} \right) + Mkiik. \begin{bmatrix} 0 \\ 0 \\ -m \cdot g \end{bmatrix} \\ > \end{aligned}$$

$$\begin{aligned} TMB_d := & \left[\left[-Dw - Dht - Dvt - Df + \left(a_2 V^2 \cos(\gamma m + \theta)^2 + a_1 V \cos(\gamma m + \theta) + a_0 \right) \cos(\gamma m \right. \right. \\ & \quad \left. \left. + \theta) - \sin(\phi) \, m \, g \right], \right. \\ & \left[Lvt + \sin(\chi) \cos(\phi) \, m \, g \right], \\ & \left[Lw + Lht + Lf + \left(a_2 V^2 \cos(\gamma m + \theta)^2 + a_1 V \cos(\gamma m + \theta) + a_0 \right) \sin(\gamma m + \theta) \right. \\ & \quad \left. - \cos(\chi) \cos(\phi) \, m \, g \right] \end{aligned}$$

ANEXO II - Planta



ANEXO III - Código em MATLAB

```
clc
close all
clear all

global T g I J K fps dps FPS DPS t trt e ert watcher Geom Aero Prop Iner Atmo colors fCLw fCDw stall mainframe setgeom setaero setpropiner setatmo setregvoo
setestant control desen anim simul intcurves RV1 RV2 viewzoom view3d b_simul solid auto_trim history help c_alphaw c_CLw c_CDw cmd
inicializa_constantes;

stall=input('Ativar Stall? Y/N [N]: ','s');
if size(stall,2)>1
    stall='N';
elseif isempty(stall) || all(stall~=['Y' 'N'])
    stall='N';
end
if strcmp(stall,'Y') stall=1;
else stall=0;
end
clc

mainframe=figure('Units','normalized','Renderer','OpenGL','Position',[0.01 0.05 0.63
0.9],'MenuBar','none','Resize','off','Name','Main','NumberTitle','off','DeleteFcn',{@quitprog});

uicontrol('Units','normalized','Style','pushbutton','String','Geometria','Position',[0.75 0.9 0.2
0.05],'Callback',{@callbacks,'setgeom'},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Aerodinâmica','Position',[0.75 0.83 0.2
0.05],'Callback',{@callbacks,'setaero'},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Propulsao e Inércia','Position',[0.75 0.76 0.2
0.05],'Callback',{@callbacks,'setpropiner'},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Atmosfera','Position',[0.75 0.69 0.2
0.05],'Callback',{@callbacks,'setatmo'},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Desenhar','Position',[0.75 0.62 0.2
0.05],'Callback',{@desenha,0},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
viewzoom=uicontrol('Units','normalized','Style','togglebutton','String','Zoom','Position',[0.75 0.55 0.2
0.05],'Callback',{@visual,1},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');
view3d=uicontrol('Units','normalized','Style','togglebutton','String','Rotate','Position',[0.75 0.48 0.2
0.05],'Callback',{@visual,2},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');
uicontrol('Units','normalized','Style','pushbutton','String','Regimes de Vão','Position',[0.75 0.41 0.2
0.05],'Callback',{@regvoo,0,0,0},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Estabilidade','Position',[0.75 0.34 0.2
0.05],'Callback',{@estant,0,0},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
b_simul=uicontrol('Units','normalized','Style','pushbutton','String','Simulador','Position',[0.75 0.27 0.2
0.05],'Callback',{@simulate},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');
uicontrol('Units','normalized','Style','pushbutton','String','Save','Position',[0.75 0.20 0.2 0.05],'Callback',{@saveprt},'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Load','Position',[0.75 0.13 0.2 0.05],'Callback',{@loadprt},'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
uicontrol('Units','normalized','Style','pushbutton','String','Quit','Position',[0.75 0.06 0.2 0.05],'Callback',{@quitprog},'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
desen=axes('Units','normalized','Position',[0.05 0.15 0.65 0.75],'Xlim',[-1 1],'Ylim',[-1 1],'Zlim',[-1 1],'CameraPosition',[1 1
1],'Color','none','XTickLabel',[],'YTickLabel',[],'ZTickLabel',[],'XTick',[],'YTick',[],'ZTick',[],'XColor',[0.8 0.8 0.8],'YColor',[0.8 0.8 0.8],'ZColor',[0.8
0.8 0.8]);
annotation('textbox',[0.05 0.062 0.1
0.04],'String','Command:', 'FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Left','VerticalAlignment','Middle');
cmd=uicontrol('Style','Edit','Units','Normalized','Position',[0.15 0.062 0.55 0.04],'String','', 'FontSize',10,'ForegroundColor',[0 0 0],'BackgroundColor',[1 1
1],'Fontname','Corbel','HorizontalAlignment','Left','Callback',{@cmdwindow});

help=[ 'stall=[1 ou 0] - Ativa/Desativa Stall '
'auto_trim=[1 ou 0] - Ativa/Desativa Trimagem auto. do profundo';
'figure - Abre janela para plotagem '
'history.* - Guarda saída do Simulador '
'plot(history.t,history.CLw, 'r') - Exemplo para traçar CLw(t) '
'help - Mostrar lista de ajuda '
'who - Listar todas as variaveis disponiveis '];

% setenv('TEMP','C:\TEMP');
% mcc -m main.m -T link:exe -I C:\Users\Mauricio\Work\TF\Programa -d C:\Users\Mauricio\Work\TF\Programa\Output -o DINVOOv1 -W WinMain -v
```

function regvoo(obj,event,atualiza,background,oncall)

```
% Regimes de Voo

global q I J K Geom Aero Prop Iner Atmo colors fCLW fCDW stall setregvoo setestant progress alphaL0ht RV1 RV2 mapavoo monitor button setzoom st1 st2 td tdA tdB
theta CLht CDht_lin CDht_n1 CLW CDW control auto_trim

if background==0

    if oncall==0
        if findobj('Name','Estabilidade Nao-linear')>0 set(setestant,'Tag','0'); end
    end
    if findobj('Name','Regimes de Voo')>0
        if oncall==0 set(setregvoo,'Visible','on','Tag','1');
        else figure(setregvoo);
        end
    else

        setregvoo=figure('Units','normalized','Position',[0.01 0.05 0.435 0.90],'MenuBar','none','Resize','off','Name','Regimes de
Voo','NumberTitle','off','Tag','1','DeleteFcn',{@tagestnt});

        mapavoo=axes('Units','normalized','Position',[0.1 0.58 0.8 0.37],'XLim',[-1 1],'YLim',[-1
1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);
        button=axes('Units','normalized','Position',[0.1 0.58 0.8 0.37],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1
1],'Color','none');
        monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1
1]);

        uicontrol('Units','normalized','Style','pushbutton','String','Atualizar','Position',[0.1 0.46 0.3
0.05],'Callback',{@regvoo,1,0,0},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
        setzoom=uicontrol('Units','normalized','Style','togglebutton','String','Zoom','Position',[0.6 0.46 0.3 0.05],'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');

        st1=uicontrol('Units','normalized','Style','radiobutton','String','Regime 1','Value',1,'Callback',{@select,1},'Position',[0.1 0.36 0.3
0.05],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');
        st2=uicontrol('Units','normalized','Style','radiobutton','String','Regime 2','Callback',{@select,2},'Position',[0.1 0.315 0.3
0.05],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');

        annotation('textbox',[0.235 0.435 0.15 0.01],'String','V [m/s]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
        annotation('textbox',[0.36 0.435 0.1 0.01],'String','\phi [circ]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
        annotation('textbox',[0.46 0.435 0.1 0.01],'String','R [m]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
        annotation('textbox',[0.57 0.435 0.1 0.01],'String','\delta_h_t
[circ]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
        annotation('textbox',[0.68 0.435 0.1 0.01],'String','\delta_w
[circ]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
        annotation('textbox',[0.79 0.435 0.1 0.01],'String','\delta_v_t
[circ]','FontSize',11,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');

        td(1)=uicontrol('Style','Text','Units','Normalized','Position',[0.26 0.375 0.1 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(2)=uicontrol('Style','Text','Units','Normalized','Position',[0.36 0.375 0.1 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(3)=uicontrol('Style','Text','Units','Normalized','Position',[0.46 0.375 0.1 0.022],'String',num2str(Inf,'%-.0f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(4)=uicontrol('Style','Text','Units','Normalized','Position',[0.56 0.375 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(5)=uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.375 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(6)=uicontrol('Style','Text','Units','Normalized','Position',[0.78 0.375 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');

        td(7)=uicontrol('Style','Text','Units','Normalized','Position',[0.26 0.33 0.1 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(8)=uicontrol('Style','Text','Units','Normalized','Position',[0.36 0.33 0.1 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(9)=uicontrol('Style','Text','Units','Normalized','Position',[0.46 0.33 0.1 0.022],'String',num2str(Inf,'%-.0f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(10)=uicontrol('Style','Text','Units','Normalized','Position',[0.56 0.33 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(11)=uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.33 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');
        td(12)=uicontrol('Style','Text','Units','Normalized','Position',[0.78 0.33 0.12 0.022],'String',num2str(0,'%-.1f'),'FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.7 0.7 0.7],'Fontname','Corbel','HorizontalAlignment','Center');

    end

end

% ----- %

try
if atualiza==1

    I=21;      % Impar
    J=20;      % Par
    K=50;      % Par

    chimin=-20*pi/180;
    chimax=20*pi/180;

    if stall

        [fCLW fCDW]=CLW_CDW;

        CLWmin=0.6;
        [CLWmax aux]=max(fCLW(0:pi/2/(J - 1):pi/2));
        thetamax=pi/2/(J - 1)*(aux - 1);

        aux=@(x) fCLW(x) - CLWmin;
        thetamin=fzero(aux,0) + Aero.alphaL0w*pi/180 + Iner.theta0*pi/180;
        thetamax=thetamax + Aero.alphaL0w*pi/180 + Iner.theta0*pi/180;

    else

        CLWmin=0.6;
        CLWmax=1.5;

        thetamin=CLWmin/Aero.CLalphaw + Aero.alphaL0w*pi/180 + Iner.theta0*pi/180;
        thetamax=CLWmax/Aero.CLalphaw + Aero.alphaL0w*pi/180 + Iner.theta0*pi/180;

    end

    phimin=-15*pi/180;
    phimax=15*pi/180;

    % ----- %

    chi=chimin:(chimax - chimin)/(I - 1):chimax;
    theta=thetamin:(thetamax - thetamin)/(J - 1):thetamax;
    phi=phimin:(phi_max - phimin)/(K - 1):phi_max;

    % ----- % Localizacao do alphaL0ht

    if auto_trim==1

        phi_trim=5*pi/180;
```

```

if stall==1
    CLw=fCLw(theta(round(J/2)) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
    CDw=fCDw(theta(round(J/2)) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
else
    CLw=Aero.CLalphaw*(theta(round(J/2)) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
    CDw=CLw^2/Aero.Arwp/pi/Aero.ew + Aero.CD0w;
end

CLf=Aero.CL0f + Aero.CLalphaf*(theta(round(J/2)) - Iner.theta0*pi/180);
CDF=Aero.CD0f + Aero.CDalphaf*(theta(round(J/2)) - Iner.theta0*pi/180);
Cmpf=Aero.Cmp0f + Aero.Cmpalphaf*(theta(round(J/2)) - Iner.theta0*pi/180);

e2=Prop.p*Atmo.rho/Prop.rho0*Prop.a2*cos(theta(round(J/2)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)^2;
e1=Prop.p*Atmo.rho/Prop.rho0*Prop.a1*cos(theta(round(J/2)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180);
e0=Prop.p*Atmo.rho/Prop.rho0*Prop.a0;

a_ht=e2*cos(theta(round(J/2)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)...
- 0.5*Atmo.rho*Geom.Sw*CDw...
- 0.5*Atmo.rho*Geom.Sht*Aero.CD0ht...
- 0.5*Atmo.rho*Geom.Svt*Aero.CD0vt...
- 0.5*Atmo.rho*Geom.Sf*CDF;
b_ht=e1*cos(theta(round(J/2)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180);
c_ht=e0*cos(theta(round(J/2)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - Iner.m0*g*sin(phi_trim);

V_medio=-(b_ht + (b_ht^2 - 4*a_ht*c_ht)^0.5)/2/a_ht;

a_ht=-(Aero.CLalpaht*pi/180)^2/pi/Aero.Arht/Aero.eht*sin(theta(round(J/2)) + Geom.gammaht*pi/180 - Iner.theta0*pi/180);
b_ht=Aero.CLalpaht*pi/180*cos(theta(round(J/2)) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
+ 2*(Aero.CLalpaht^2*theta(round(J/2))*pi/180 - Aero.CLalpaht^2*Iner.theta0*pi/180*pi/180)/pi/Aero.Arht/Aero.eht*sin(theta(round(J/2)) +
Geom.gammaht*pi/180 - Iner.theta0*pi/180);
c_ht=(- Geom.Sht*Aero.cht*Aero.Cmp0ht...
- Geom.lw*Geom.Sw*CLw*cos(theta(round(J/2)) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
- Geom.lw*Geom.Sw*CDw*sin(theta(round(J/2)) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
- Geom.Sw*Aero.cw*Aero.Cmp0w...
- Geom.lvt*Geom.Svt*Aero.CD0vt*sin(theta(round(J/2)) + Geom.gammavt*pi/180 - Iner.theta0*pi/180)...
- Geom.lf*Geom.Sf*CLf*cos(theta(round(J/2)) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
- Geom.lf*Geom.Sf*CDF*sin(theta(round(J/2)) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
- Geom.Sf*Aero.cf*Cmpf...
- 2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e2...
-2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e1/V_medio...
-2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e0/V_medio^2)/(Geom.lht*Geom.Sht)...
-Aero.CD0ht*sin(theta(round(J/2)) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
-Aero.CLalpaht*(theta(round(J/2)) - Iner.theta0*pi/180)*cos(theta(round(J/2)) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
-Aero.CLalpaht^2*(theta(round(J/2))^2 + (Iner.theta0*pi/180)^2 -
2*theta(round(J/2))*Iner.theta0*pi/180)/pi/Aero.Arht/Aero.eht*sin(theta(round(J/2)) + Geom.gammaht*pi/180 - Iner.theta0*pi/180);

Aero.alphaL0ht=-(b_ht + (b_ht^2 - 4*a_ht*c_ht)^0.5)/2/a_ht;
if findobj('Name','Aerodinamica')>0
    set(alphaL0ht,'String',num2str(round(Aero.alphaL0ht*10)/10));
end

end

% ----- %

v1=zeros(1,K);
v2=zeros(1,K);

V=zeros(1,J);
Phi=zeros(1,J);
R=zeros(1,J);
deltaht=zeros(1,J);
deltaw=zeros(1,J);
deltavt=zeros(1,J);

if findobj('Name','Waitbar')>0 close(progress); end
progress=waitbar(0,'Calculando...','Name','Waitbar');
for j=1:J

    if stall==1
        CLw(j)=fCLw(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
        CDw(j)=fCDw(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
    else
        CLw(j)=Aero.CLalphaw*(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
        CDw(j)=CLw(j)^2/Aero.Arwp/pi/Aero.ew + Aero.CD0w;
    end

    CL0ht=Aero.CLalpaht*(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180);
    CDi0ht=(Aero.CLalpaht*(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180))^2/pi/Aero.Arht/Aero.eht;
    CDideltaht=2*Aero.CLalpaht*Aero.Cldeltaht*(theta(j) - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180)/pi/Aero.Arht/Aero.eht;

    CLf=Aero.CL0f + Aero.CLalphaf*(theta(j) - Iner.theta0*pi/180);
    CDF=Aero.CD0f + Aero.CDalphaf*(theta(j) - Iner.theta0*pi/180);
    Cmpf=Aero.Cmp0f + Aero.Cmpalphaf*(theta(j) - Iner.theta0*pi/180);

    e2=Prop.p*Atmo.rho/Prop.rho0*Prop.a2*cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)^2;
    e1=Prop.p*Atmo.rho/Prop.rho0*Prop.a1*cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180);
    e0=Prop.p*Atmo.rho/Prop.rho0*Prop.a0;

    % QsdhtxV2=...
    % (- TbxV2...
    % - QxV2*Geom.lht*Geom.Sht*CL0ht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.lht*Geom.Sht*(Aero.CD0ht + CDi0ht)*sin(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.Sht*Aero.cht*Aero.Cmp0ht...
    % - QxV2*Geom.lw*Geom.Sw*CLw(j)*cos(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.lw*Geom.Sw*CDw(j)*sin(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.Sw*Aero.cw*Aero.Cmp0w...
    % - QxV2*Geom.lvt*Geom.Svt*Aero.CD0vt*sin(theta(j) + Geom.gammavt*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.lf*Geom.Sf*CLf*cos(theta(j) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.lf*Geom.Sf*CDF*sin(theta(j) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
    % - QxV2*Geom.Sf*Aero.cf*Cmpf...
    % - Geom.lt*sin(Geom.betat*pi/180)*(V4*e2 + V3*e1 + V2*e0)/...
    % (Geom.lht*Aero.Cldeltaht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) + Geom.lht*CDideltaht*sin(theta(j) + Geom.gammaht*pi/180 -
    Iner.theta0*pi/180) + Aero.cht*Aero.Cmpdeltaht);

    SdhtxV4_4=(-Geom.lht*Geom.Sht*CL0ht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
    - Geom.lht*Geom.Sht*(Aero.CD0ht + CDi0ht)*sin(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
    - Geom.Sht*Aero.cht*Aero.Cmp0ht...
    - Geom.lw*Geom.Sw*CLw(j)*cos(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
    - Geom.lw*Geom.Sw*CDw(j)*sin(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
    - Geom.Sw*Aero.cw*Aero.Cmp0w...
    - Geom.lvt*Geom.Svt*Aero.CD0vt*sin(theta(j) + Geom.gammavt*pi/180 - Iner.theta0*pi/180)...
    - Geom.lf*Geom.Sf*CLf*cos(theta(j) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
    - Geom.lf*Geom.Sf*CDF*sin(theta(j) + Geom.gammaf*pi/180 - Iner.theta0*pi/180)...
    - Geom.Sf*Aero.cf*Cmpf)/...
    (Geom.lht*Aero.Cldeltaht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) + Geom.lht*CDideltaht*sin(theta(j) + Geom.gammaht*pi/180 -
    Iner.theta0*pi/180) + Aero.cht*Aero.Cmpdeltaht)...
    - 2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e2/...
    (Geom.lht*Aero.Cldeltaht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) + Geom.lht*CDideltaht*sin(theta(j) + Geom.gammaht*pi/180 -
    Iner.theta0*pi/180) + Aero.cht*Aero.Cmpdeltaht);

    SdhtxV4_3=-2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e1/(Geom.lht*Aero.Cldeltaht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) +
    Geom.lht*CDideltaht*sin(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) + Aero.cht*Aero.Cmpdeltaht);

    SdhtxV4_2=-2/Atmo.rho*Geom.lt*sin(Geom.betat*pi/180)*e0/(Geom.lht*Aero.Cldeltaht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) +
    Geom.lht*CDideltaht*sin(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180) + Aero.cht*Aero.Cmpdeltaht);

    aux=0;

```

```

for i=1:I
    parfor (k=1:K)

        TbxV2=g^2*tan(chi(i))^2*(Iner.J10 - Iner.J30)*(2*cos(chi(i))*cos(phi(k))*cos(theta(j))^2*sin(phi(k)) - sin(theta(j))*cos(theta(j))*sin(phi(k))^2 + cos(chi(i))^2*cos(phi(k))^2*cos(theta(j))*sin(theta(j)) - sin(phi(k))*cos(chi(i))*cos(phi(k)));

        SdhtxV4_0=-2/Atmo.rho*TbxV2/(Geom.lht*Aero.CLdeltaht*cos(theta(j)) + Geom.gammat*pi/180 - Iner.theta0*pi/180) + Geom.lht*CDideltaht*sin(theta(j)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Aero.cht*Aero.CMpdeltaht);

        % Polinomio 1
        % 0=E*cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - Dw - Dht - Dvt - Df - W*sin(phi(k))

        b4=cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e2...
            - 0.5*Atmo.rho*Geom.Sw*CDw(j)...
            - 0.5*Atmo.rho*(Geom.Sht*(Aero.CD0ht + CDi0ht) + CDideltaht*SdhtxV4_4)...
            - 0.5*Atmo.rho*Geom.Svt*Aero.CD0vt...
            - 0.5*Atmo.rho*Geom.Sf*CDF;
        b3=cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e1...
            - 0.5*Atmo.rho*CDideltaht*SdhtxV4_3;
        b2=cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e0...
            - 0.5*Atmo.rho*CDideltaht*SdhtxV4_2...
            - Iner.m0*g*sin(phi(k));
        b1=0;
        b0=-0.5*Atmo.rho*CDideltaht*SdhtxV4_0;

        r=roots([b4 b3 b2 b1 b0]);
        for l=1:4
            if imag(r(l))~=0 r(l)=0; end
        end
        v1(k)=max(r);

        % Polinomio 2
        % 0=E*sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) + Lw + Lht + Lf - W*cos(phi(k))*sec(chi(i));

        c4=sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e2...
            + 0.5*Atmo.rho*Geom.Sw*CLw(j)...
            + 0.5*Atmo.rho*(Geom.Sht*CL0ht + Aero.CLdeltaht*SdhtxV4_4)...
            + 0.5*Atmo.rho*Geom.Sf*CLF;
        c3=sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e1...
            + 0.5*Atmo.rho*Aero.CLdeltaht*SdhtxV4_3;
        c2=sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*e0...
            + 0.5*Atmo.rho*Aero.CLdeltaht*SdhtxV4_2...
            - Iner.m0*g*cos(phi(k))*sec(chi(i));
        c1=0;
        c0=0.5*Atmo.rho*Aero.CLdeltaht*SdhtxV4_0;

        r=roots([c4 c3 c2 c1 c0]);
        for l=1:4
            if imag(r(l))~=0 r(l)=0; end
        end
        v2(k)=max(r);

    end
    [v k]=min(abs(v1 - v2));
    V(i,j)=(v1(k) + v2(k))/2;
    Phi(i,j)=phi(k);

    if chi(i)<0
        chiE(i)=chi(i)*180/pi;
        RE(i,j)=V(i,j)^2*cos(Phi(i,j))/g/tan(chi(i));
    elseif chi(i)>0
        if aux==0 aux=i - 1; end
        chiD(i - aux)=chi(i)*180/pi;
        RD(i - aux,j)=V(i,j)^2*cos(Phi(i,j))/g/tan(chi(i));
    end

    if chi(i)==0 R(i,j)=Inf;
    else R(i,j)=V(i,j)^2*cos(Phi(i,j))/g/tan(chi(i));
    end

    TaxV2=g^2*tan(chi(i))^2*(Iner.J10 - Iner.J20)*sin(chi(i))*cos(phi(k))*(sin(theta(j))*sin(phi(k)) - cos(theta(j))*cos(chi(i))*cos(phi(k)));
    TbxV2=g^2*tan(chi(i))^2*(Iner.J30 - Iner.J30)*(2*cos(chi(i))*cos(phi(k))*cos(theta(j))^2*sin(phi(k)) - sin(theta(j))*cos(theta(j))*sin(phi(k))^2 + cos(chi(i))^2*cos(phi(k))^2*cos(theta(j))*sin(theta(j)) - sin(phi(k))*cos(chi(i))*cos(phi(k)));
    TcxV2=g^2*tan(chi(i))^2*(Iner.J10 - Iner.J20)*sin(chi(i))*cos(phi(k))*(cos(theta(j))*sin(phi(k)) + sin(theta(j))*cos(chi(i))*cos(phi(k)));

    SdhtxV4_0=-2/Atmo.rho*TbxV2/(Geom.lht*Aero.CLdeltaht*cos(theta(j)) + Geom.gammat*pi/180 - Iner.theta0*pi/180) + Geom.lht*CDideltaht*sin(theta(j)) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Aero.cht*Aero.CMpdeltaht);

    CMrRwxV2=Aero.CMrRyw*0.5*cos(chi(i))*cos(phi(k))*g*tan(chi(i))*Geom.bw - Aero.CMrRrw*0.5*sin(phi(k))*g*tan(chi(i))*Geom.bw;
    CMrRwV2=Aero.CMrRyw*0.5*cos(chi(i))*cos(phi(k))*g*tan(chi(i))*Geom.bw - Aero.CMrRrw*0.5*sin(phi(k))*g*tan(chi(i))*Geom.bw;

    QScdwV2_0=(TcxV2*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180) + TaxV2*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180))/Aero.CMydeltaw*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180) - Aero.CMrdeltaw*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180));
    QScdwV2_2=(-0.5*Atmo.rho*Geom.Sw*Aero.cw*CMrRwxV2*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180) + Aero.CMydeltaw*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180) - Aero.CMrdeltaw*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180));

    LvtxV2_0=((TcxV2 - QScdwV2_0*Aero.CMydeltaw)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180) - (TaxV2 + QScdwV2_0*Aero.CMrdeltaw)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180))/Geom.lvt;
    LvtxV2_2=((QScdwV2_2*Aero.CMydeltaw - 0.5*Atmo.rho*Geom.Sw*Aero.cw*CMrRwxV2)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180) - (QScdwV2_2*Aero.CMrdeltaw + 0.5*Atmo.rho*Geom.Sw*Aero.cw*CMrRwV2)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180))/Geom.lvt;

    deltaht(i,j)=(SdhtxV4_4 + SdhtxV4_3/V(i,j) + SdhtxV4_2/V(i,j)^2 + SdhtxV4_0/V(i,j)^4)/Geom.Sht;
    deltaw(i,j)=2*(QScdwV2_0/V(i,j)^4 + QScdwV2_2/V(i,j)^2)/Atmo.rho/Geom.Sw/Aero.cw;
    deltavt(i,j)=2*(LvtxV2_0/V(i,j)^4 + LvtxV2_2/V(i,j)^2)/Atmo.rho/Geom.Svt/Aero.CLdeltavt;

    CLht(i,j)=CL0ht + Aero.CLdeltaht*deltaht(i,j);
    CDht_lin(i,j)=Aero.CD0ht + CDi0ht + CDideltaht*deltaht(i,j);
    CDht_nl(i,j)=Aero.CD0ht + (CL0ht + Aero.CLdeltaht*deltaht(i,j))^2/pi/Aero.Arht/Aero.eht;

    % ----- %

    E=e2*V(i,j)^2 + e1*V(i,j) + e0;
    W=Iner.m0*g;

    Lw=0.5*Atmo.rho*V(i,j)^2*Geom.Sw*CLw(j);
    Dw=0.5*Atmo.rho*V(i,j)^2*Geom.Sw*CDw(j);
    Mrw=0.5*Atmo.rho*V(i,j)^2*Geom.Sw*Aero.cw*(Aero.CMrdeltaw*deltaw(i,j) + CMrRwxV2/V(i,j)^2);
    Myw=0.5*Atmo.rho*V(i,j)^2*Geom.Sw*Aero.cw*(Aero.CMydeltaw*deltaw(i,j) + CMrRwV2/V(i,j)^2);
    Mpw=0.5*Atmo.rho*V(i,j)^2*Geom.Sw*Aero.cw*Aero.CMp0w;

    Lht=0.5*Atmo.rho*V(i,j)^2*Geom.Sht*(CL0ht + Aero.CLdeltaht*deltaht(i,j));
    Dht=0.5*Atmo.rho*V(i,j)^2*Geom.Sht*(Aero.CD0ht + CDi0ht + CDideltaht*deltaht(i,j));
    Mph=0.5*Atmo.rho*V(i,j)^2*Geom.Sht*Aero.cht*(Aero.CMp0ht + Aero.CMpdeltaht*deltaht(i,j));

    Lf=0.5*Atmo.rho*V(i,j)^2*Geom.Sf*CLF;
    Df=0.5*Atmo.rho*V(i,j)^2*Geom.Sf*CDF;
    Mpf=0.5*Atmo.rho*V(i,j)^2*Geom.Sf*Aero.cf*CMpf;

    Lvt=0.5*Atmo.rho*V(i,j)^2*Geom.Svt*Aero.CLdeltavt*deltavt(i,j);
    Dvt=0.5*Atmo.rho*V(i,j)^2*Geom.Svt*Aero.CD0vt;

    Ft(i,j)=E*cos(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - Dw - Dht - Dvt - Df - W*sin(Phi(i,j));
    Fr(i,j)=(E*sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) + Lw + Lht + Lf)*sin(chi(i));
    Fn(i,j)=(E*sin(theta(j) + Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) + Lw + Lht + Lf)*cos(chi(i)) - W*cos(Phi(i,j));

```

```

% Ta(i,j)=-g^2*tan(chi(i))^2/V(i,j)^2*(Iner.J30 - Iner.J20)*sin(chi(i))*cos(Phi(i,j))*(sin(theta(j))*sin(Phi(i,j)) -
cos(theta(j))*cos(chi(i))*cos(Phi(i,j)))...
% - Mrw...
% - Geom.lvt*sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*Lvt;
%
% Tb(i,j)=-g^2*tan(chi(i))^2/V(i,j)^2*(Iner.J10 - Iner.J30)*(2*cos(chi(i))*cos(Phi(i,j))*cos(theta(j))^2*sin(Phi(i,j)) -
sin(theta(j))*cos(theta(j))*sin(Phi(i,j))^2 + cos(chi(i))^2*cos(Phi(i,j))^2*cos(theta(j))*sin(theta(j)) - sin(Phi(i,j))*cos(chi(i))*cos(Phi(i,j)))...
% - Geom.lw*Lw*cos(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
% - Geom.lw*Dw*sin(theta(j) + Geom.gammaw*pi/180 - Iner.theta0*pi/180)...
% - Mpw...
% - Geom.lht*Lht*cos(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
% - Geom.lht*Dht*sin(theta(j) + Geom.gammaht*pi/180 - Iner.theta0*pi/180)...
% - Mph...
% - Geom.lvt*Dvt*sin(theta(j) + Geom.gammavt*pi/180 - Iner.theta0*pi/180)...
% - Geom.lf*Lf*cos(theta(j) + Geom.gammalf*pi/180 - Iner.theta0*pi/180)...
% - Geom.lf*Df*sin(theta(j) + Geom.gammalf*pi/180 - Iner.theta0*pi/180)...
% - Mpf...
% - E*Geom.lt*sin(Geom.betat*pi/180);
%
% Tc(i,j)=-g^2*tan(chi(i))^2/V(i,j)^2*(Iner.J10 - Iner.J20)*sin(chi(i))*cos(Phi(i,j))*(cos(theta(j))*sin(Phi(i,j)) +
sin(theta(j))*cos(chi(i))*cos(Phi(i,j)))...
% + Myw...
% + Geom.lvt*cos(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*Lvt;
%
% ----- %
end
waitbar(j/J,progress);

end
close(progress);

% ----- %

if RV1(9)==1
    RV1(1)=V(RV1(7),RV1(8));
    RV1(2)=Phi(RV1(7),RV1(8));
    RV1(3)=R(RV1(7),RV1(8));
    RV1(4)=deltaht(RV1(7),RV1(8));
    RV1(5)=deltaw(RV1(7),RV1(8));
    RV1(6)=deltavt(RV1(7),RV1(8));
    RV1(10)=chi(RV1(7));
    RV1(11)=theta(RV1(8));

    if findobj('Name','Estabilidad Nao-linear')>0
        set(tdA(1),'String',num2str(RV1(1),'%.1f'));
        set(tdA(2),'String',num2str(RV1(2)*180/pi,'%.1f'));
        set(tdA(3),'String',num2str(RV1(1)*cos(RV1(2))/RV1(3)*180/pi,'%.1f'));
        set(tdA(4),'String',num2str(RV1(10)*180/pi,'%.1f'));
        set(tdA(5),'String',num2str(RV1(11)*180/pi,'%.1f'));
        set(tdA(6),'String',num2str(RV1(12)*180/pi,'%.0f'));
        if findobj('Name','Entradas')>0
            set(tdA(7),'String',num2str(RV1(4)*180/pi,'%.1f'));
            set(tdA(8),'String',num2str(RV1(5)*180/pi,'%.1f'));
            set(tdA(9),'String',num2str(RV1(6)*180/pi,'%.1f'));
        end
    end

end

end

if RV2(9)==1
    RV2(1)=V(RV2(7),RV2(8));
    RV2(2)=Phi(RV2(7),RV2(8));
    RV2(3)=R(RV2(7),RV2(8));
    RV2(4)=deltaht(RV2(7),RV2(8));
    RV2(5)=deltaw(RV2(7),RV2(8));
    RV2(6)=deltavt(RV2(7),RV2(8));
    RV2(10)=chi(RV2(7));
    RV2(11)=theta(RV2(8));

    if findobj('Name','Estabilidad Nao-linear')>0
        set(tdB(1),'String',num2str(RV2(1),'%.1f'));
        set(tdB(2),'String',num2str(RV2(2)*180/pi,'%.1f'));
        set(tdB(3),'String',num2str(RV2(1)*cos(RV2(2))/RV2(3)*180/pi,'%.1f'));
        set(tdB(4),'String',num2str(RV2(10)*180/pi,'%.1f'));
        set(tdB(5),'String',num2str(RV2(11)*180/pi,'%.1f'));
        set(tdB(6),'String',num2str(RV2(12)*180/pi,'%.0f'));
        if findobj('Name','Entradas')>0
            set(tdB(7),'String',num2str(RV2(4)*180/pi,'%.1f'));
            set(tdB(8),'String',num2str(RV2(5)*180/pi,'%.1f'));
            set(tdB(9),'String',num2str(RV2(6)*180/pi,'%.1f'));
        end
    end

end

end

% ----- %

if background==0
    figure(setregvoo);

    if RV1(9)==0
        RV1=zeros(12,1);
        RV1(3)=Inf;
    end

    if RV2(9)==0
        RV2=zeros(12,1);
        RV2(3)=Inf;
    end

    set(td(1),'String',num2str(RV1(1),'%.1f'));
    set(td(2),'String',num2str(RV1(2)*180/pi,'%.1f'));
    set(td(3),'String',num2str(RV1(3),'%.0f'));
    set(td(4),'String',num2str(RV1(4)*180/pi,'%.1f'));
    set(td(5),'String',num2str(RV1(5)*180/pi,'%.1f'));
    set(td(6),'String',num2str(RV1(6)*180/pi,'%.1f'));

    set(td(7),'String',num2str(RV2(1),'%.1f'));
    set(td(8),'String',num2str(RV2(2)*180/pi,'%.1f'));
    set(td(9),'String',num2str(RV2(3),'%.0f'));
    set(td(10),'String',num2str(RV2(4)*180/pi,'%.1f'));
    set(td(11),'String',num2str(RV2(5)*180/pi,'%.1f'));
    set(td(12),'String',num2str(RV2(6)*180/pi,'%.1f'));

    V_levels=round(linspace(round(min(min(V))), round(max(max(V))), 9)*10)/10;
    phi_levels=round(linspace(round(min(min(Phi*180/pi))), round(max(max(Phi*180/pi))), 9)*10)/10;
    R_levels=round(logspace(log10(min(min(abs(RD))), log10(max(max(RD))), 8));

    delete (mapavoo);

```

```

mapavoo=axes('Units','normalized','Position',[0.1 0.58 0.8 0.37],'Drawmode','fast','FontSize',10,'XLim',[chi(1)*180/pi
chi(I)*180/pi],'YLim',[theta(1)*180/pi theta(J)*180/pi],'ColorOrder',colors,'FontName','Corbel');
hold on

xlabel('Angulo de inclinação lateral \chi [\circ]');
ylabel('\theta [\circ]');

[C contV]=contour(chi*180/pi,theta*180/pi,V,V_levels);
clabel(C,contV,'LabelSpacing',350,'FontSize',8,'FontName','Corbel','FontWeight','bold','Color',[0.2 0.2 0.2],'Rotation',0);
hh=get(contV,'Children');
aux=0;
for i=1:length(hh)
    if strcmp(get(hh(i),'Type'),'patch')==1
        if aux==0 aux=1; end
        set(hh(i),'LineStyle','--','CDataMapping','direct');
        data=get(hh(i),'CData');
        for j=1:length(data)
            data(j)=ceil((i - aux)/(length(hh) - aux)*size(colors,1)/4);
        end
        set(hh(i),'CData',data);
    else
        set(hh(i),'String',sprintf('V %s',get(hh(i),'String')));
    end
end

[C contphi]=contour(chi*180/pi,theta*180/pi,Phi*180/pi,phi_levels);
clabel(C,contphi,'LabelSpacing',400,'FontSize',8,'FontName','Corbel','FontWeight','bold','Color',[0.2 0.2 0.2],'Rotation',0);
hh=get(contphi,'Children');
aux=0;
for i=1:length(hh)
    if strcmp(get(hh(i),'Type'),'patch')==1
        if aux==0 aux=1; end
        set(hh(i),'LineStyle',':','CDataMapping','direct');
        data=get(hh(i),'CData');
        for j=1:length(data)
            data(j)=ceil((i - aux)/(length(hh) - aux)*size(colors,1)/4 + size(colors,1)/4);
        end
        set(hh(i),'CData',data);
    else
        set(hh(i),'String',sprintf('\phi %s',get(hh(i),'String')));
    end
end

[C contRE]=contour(chiE,theta*180/pi,RE,-R_levels);
clabel(C,contRE,'LabelSpacing',400,'FontSize',8,'FontName','Corbel','FontWeight','bold','Color',[0.2 0.2 0.2],'Rotation',0);
hh=get(contRE,'Children');
aux=0;
for i=1:length(hh)
    if strcmp(get(hh(i),'Type'),'patch')==1
        if aux==0 aux=1; end
        set(hh(i),'LineStyle',':','CDataMapping','direct');
        data=get(hh(i),'CData');
        for j=1:length(data)
            data(j)=ceil((i - aux)/(length(hh) - aux)*size(colors,1)/4 + size(colors,1)/2);
        end
        set(hh(i),'CData',data);
    else
        set(hh(i),'String',sprintf('R %s',get(hh(i),'String')));
    end
end

[C contRD]=contour(chiD,theta*180/pi,RD,R_levels);
clabel(C,contRD,'LabelSpacing',400,'FontSize',8,'FontName','Corbel','FontWeight','bold','Color',[0.2 0.2 0.2],'Rotation',0);
hh=get(contRD,'Children');
aux=0;
for i=1:length(hh)
    if strcmp(get(hh(i),'Type'),'patch')==1
        if aux==0 aux=1; end
        set(hh(i),'LineStyle',':','CDataMapping','direct');
        data=get(hh(i),'CData');
        for j=1:length(data)
            data(j)=ceil((i - aux)/(length(hh) - aux)*size(colors,1)/4 + 3*size(colors,1)/4);
        end
        set(hh(i),'CData',data);
    else
        set(hh(i),'String',sprintf('R %s',get(hh(i),'String')));
    end
end

ar1=text('Visible','off','String','+1','Color','r','FontSize',15,'FontName','Corbel');
if RV1(9)==1 set(ar1,'Visible','on','Position',[chi(RV1(7))*180/pi theta(RV1(8))*180/pi]); end
ar2=text('Visible','off','String','+2','Color','r','FontSize',15,'FontName','Corbel');
if RV2(9)==1 set(ar2,'Visible','on','Position',[chi(RV2(7))*180/pi theta(RV2(8))*180/pi]); end

% -----%

delete(monitor);
if RV1(9)==1 && get(st1,'Value')==1

    monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'Drawmode','fast','FontSize',8,'FontName','Corbel');
    hold on

    plot(theta*180/pi,CLw/10,'b');
    plot(theta*180/pi,CDw,'m');

    plot(theta*180/pi,CLht(RV1(7),:)/10,'g');
    plot(theta*180/pi,CDht_lin(RV1(7),:),'r');
    plot(theta*180/pi,CDht_n1(RV1(7),:),'k--');

    title('Monitor');
    xlabel('\theta [\circ]');
    ylabel('CL.e-1/CD - w:b/m ht:g/r');

elseif RV2(9)==1 && get(st2,'Value')==1

    monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'Drawmode','fast','FontSize',8,'FontName','Corbel');
    hold on

    plot(theta*180/pi,CLw/10,'b');
    plot(theta*180/pi,CDw,'m');

    plot(theta*180/pi,CLht(RV2(7),:)/10,'g');
    plot(theta*180/pi,CDht_lin(RV2(7),:),'r');
    plot(theta*180/pi,CDht_n1(RV2(7),:),'k--');

    title('Monitor');
    xlabel('\theta [\circ]');
    ylabel('CL.e-1/CD - w:b/m ht:g/r');

else monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1
1 1]);
end

% -----%

set(button,'ButtonDownFcn',{getpoint},'XLim',[1 I],'YLim',[1 J]);
set(setzoom,'Enable','on','Value',0,'Callback',{@actzoom});
axes(button);

```

```

end % background

end % atualiza==1
catch
    Error='Unable to calculate.'
end

%-----%

function getpoint(obj,event)

    global refresh

    point=abs(round(get(button,'CurrentPoint')));

    if get(stl,'Value')==1

        RV1(1)=V(point(1,1),point(1,2));
        RV1(2)=Phi(point(1,1),point(1,2));
        RV1(3)=R(point(1,1),point(1,2));
        RV1(4)=deltaht(point(1,1),point(1,2));
        RV1(5)=deltaw(point(1,1),point(1,2));
        RV1(6)=deltavt(point(1,1),point(1,2));
        RV1(7)=point(1,1);
        RV1(8)=point(1,2);
        RV1(9)=1;
        RV1(10)=chi(point(1,1));
        RV1(11)=theta(point(1,2));
        RV1(12)=0;

        set(td(1),'String',num2str(RV1(1),'%.1f'));
        set(td(2),'String',num2str(RV1(2)*180/pi,'%.1f'));
        set(td(3),'String',num2str(RV1(3),'%.0f'));
        set(td(4),'String',num2str(RV1(4)*180/pi,'%.1f'));
        set(td(5),'String',num2str(RV1(5)*180/pi,'%.1f'));
        set(td(6),'String',num2str(RV1(6)*180/pi,'%.1f'));

        set(ar1,'Visible','on','Position',[chi(point(1,1))*180/pi theta(point(1,2))*180/pi]);

        if findobj('Name','Estabilidade Nao-linear')>0

            set(tdA(1),'String',num2str(RV1(1),'%.1f'));
            set(tdA(2),'String',num2str(RV1(2)*180/pi,'%.1f'));
            set(tdA(3),'String',num2str(RV1(1)*cos(RV1(2))/RV1(3)*180/pi,'%.1f'));
            set(tdA(4),'String',num2str(RV1(10)*180/pi,'%.1f'));
            set(tdA(5),'String',num2str(RV1(11)*180/pi,'%.1f'));
            set(tdA(6),'String',num2str(RV1(12)*180/pi,'%.0f'));
            if findobj('Name','Entradas')>0
                set(tdA(7),'String',num2str(RV1(4)*180/pi,'%.1f'));
                set(tdA(8),'String',num2str(RV1(5)*180/pi,'%.1f'));
                set(tdA(9),'String',num2str(RV1(6)*180/pi,'%.1f'));
            end
        end

    else

        RV2(1)=V(point(1,1),point(1,2));
        RV2(2)=Phi(point(1,1),point(1,2));
        RV2(3)=R(point(1,1),point(1,2));
        RV2(4)=deltaht(point(1,1),point(1,2));
        RV2(5)=deltaw(point(1,1),point(1,2));
        RV2(6)=deltavt(point(1,1),point(1,2));
        RV2(7)=point(1,1);
        RV2(8)=point(1,2);
        RV2(9)=1;
        RV2(10)=chi(point(1,1));
        RV2(11)=theta(point(1,2));
        RV2(12)=0;

        set(td(7),'String',num2str(RV2(1),'%.1f'));
        set(td(8),'String',num2str(RV2(2)*180/pi,'%.1f'));
        set(td(9),'String',num2str(RV2(3),'%.0f'));
        set(td(10),'String',num2str(RV2(4)*180/pi,'%.1f'));
        set(td(11),'String',num2str(RV2(5)*180/pi,'%.1f'));
        set(td(12),'String',num2str(RV2(6)*180/pi,'%.1f'));

        set(ar2,'Visible','on','Position',[chi(point(1,1))*180/pi theta(point(1,2))*180/pi]);

        if findobj('Name','Estabilidade Nao-linear')>0

            set(tdB(1),'String',num2str(RV2(1),'%.1f'));
            set(tdB(2),'String',num2str(RV2(2)*180/pi,'%.1f'));
            set(tdB(3),'String',num2str(RV2(1)*cos(RV2(2))/RV2(3)*180/pi,'%.1f'));
            set(tdB(4),'String',num2str(RV2(10)*180/pi,'%.1f'));
            set(tdB(5),'String',num2str(RV2(11)*180/pi,'%.1f'));
            set(tdB(6),'String',num2str(RV2(12)*180/pi,'%.0f'));
            if findobj('Name','Entradas')>0
                set(tdB(7),'String',num2str(RV2(4)*180/pi,'%.1f'));
                set(tdB(8),'String',num2str(RV2(5)*180/pi,'%.1f'));
                set(tdB(9),'String',num2str(RV2(6)*180/pi,'%.1f'));
            end
        end

    end

delete(monitor);
monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'Drawmode','fast','FontSize',8,'FontName','Corbel');
hold on

plot(theta*180/pi,CLw/10,'b');
plot(theta*180/pi,CDw,'m');

plot(theta*180/pi,CLht(point(1,1),:)/10,'g');
plot(theta*180/pi,CDht_lin(point(1,1),:),'r');
plot(theta*180/pi,CDht_n1(point(1,1),:),'k--');

title('Monitor');
xlabel('\theta [circ]');
ylabel('CL.e-1/CD - w:b/m ht:g/r');

axes(button);

if RV1(9)==1 && RV2(9)==1
    if findobj('Name','Estabilidade Nao-linear')>0 set(refresh,'Enable','on'); end
end

end

%-----%

function actzoom(obj,event)

    axes(mapavoo);
    if get(setzoom,'Value')==1 zoom on
    else

```

```

zoom off
Xlim=get(mapavoo,'XLim');
Ylim=get(mapavoo,'YLim');

Xlim(1)=(Xlim(1) - chimin*180/pi)*(I - 1)/(chimax*180/pi - chimin*180/pi) + 1;
Xlim(2)=(Xlim(2) - chimin*180/pi)*(I - 1)/(chimax*180/pi - chimin*180/pi) + 1;
Ylim(1)=(Ylim(1) - theta(1)*180/pi)*(J - 1)/(theta(J)*180/pi - theta(1)*180/pi) + 1;
Ylim(2)=(Ylim(2) - theta(1)*180/pi)*(J - 1)/(theta(J)*180/pi - theta(1)*180/pi) + 1;

set(button,'XLim',Xlim);
set(button,'YLim',Ylim);
axes(button);
end

end

%-----%

function select(obj,event,which)

if which==1
    if get(st1,'Value')==1 set(st2,'Value',0);
    else set(st2,'Value',1);
    end
else
    if get(st2,'Value')==1 set(st1,'Value',0);
    else set(st1,'Value',1);
    end
end

delete (monitor);
if RV1(9)==1 && get(st1,'Value')==1

    monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'Drawmode','fast','FontSize',8,'FontName','Corbel');
    hold on

    plot(theta*180/pi,CLw/10,'b');
    plot(theta*180/pi,CDw,'m');

    plot(theta*180/pi,CLht(RV1(7),:)/10,'g');
    plot(theta*180/pi,CDht_lin(RV1(7),:),'r');
    plot(theta*180/pi,CDht_n1(RV1(7),:),'k--');

    title('Monitor');
    xlabel('\theta [\circ]');
    ylabel('CL.e-1/CD - w:b/m ht:g/r');

elseif RV2(9)==1 && get(st2,'Value')==1

    monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'Drawmode','fast','FontSize',8,'FontName','Corbel');
    hold on

    plot(theta*180/pi,CLw/10,'b');
    plot(theta*180/pi,CDw,'m');

    plot(theta*180/pi,CLht(RV2(7),:)/10,'g');
    plot(theta*180/pi,CDht_lin(RV2(7),:),'r');
    plot(theta*180/pi,CDht_n1(RV2(7),:),'k--');

    title('Monitor');
    xlabel('\theta [\circ]');
    ylabel('CL.e-1/CD - w:b/m ht:g/r');

else monitor=axes('Units','normalized','Position',[0.1 0.06 0.8 0.22],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1
1]);
end

end

end

%-----%

function tagestant(obj,event)

if findobj('Name','Estabilidade Nao-linear')>0 set(setestant,'Visible','on','Tag','1'); end

end

%-----%

end

```

```
function estant(obj,event,atualiza,callregvoo)
```

```
% Estabilidade
```

```
global T N t e watcher mode Geom Aero Prop Iner Atmo stall fCLw fCDw setregvoo setestant control RV1 RV2 estabV estabphi estabpsi estabchi estabtheta estabeta  
estabdeltaht estabdeltaaw estabdeltaTavt estabv estabw estabw refresh traj3d tDA tDB input1 input2 input3 input4 entr trans setentr tr chilast thetalast etalast
```

```
mode=0;
```

```
if findobj('Name','Regimes de V6o')>0 set(setregvoo,'Tag','0'); end  
if findobj('Name','Estabilidade Nao-linear')>0  
    set(setestant,'Visible','on','Tag','1');  
    if findobj('Name','Entradas')>0 set(control,'Visible','on'); end  
else
```

```
    setestant=figure('Units','normalized','Position',[0.01 0.05 0.435 0.90],'Menubar','none','Resize','off','Name','Estabilidade Nao-linear',  
    'NumberTitle','off','Tag','1','DeleteFcn',{@tagregvoo});
```

```
    estabV=axes('Units','normalized','Position',[0.15 0.87 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);  
    estabphi=axes('Units','normalized','Position',[0.15 0.74 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);  
    estabpsi=axes('Units','normalized','Position',[0.15 0.61 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);  
    estabchi=axes('Units','normalized','Position',[0.15 0.48 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);  
    estabtheta=axes('Units','normalized','Position',[0.15 0.35 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);  
    estabeta=axes('Units','normalized','Position',[0.15 0.22 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);
```

```
    refresh=uicontrol('Units','normalized','Style','pushbutton','String','Atualizar','Position',[0.15 0.11 0.3  
0.05],'Callback',{@estant,1,0},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35  
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');  
    traj3d=uicontrol('Units','normalized','Style','pushbutton','String','Animate','Position',[0.15 0.04 0.3  
0.05],'Callback',{@animate},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35  
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold','Enable','off');  
    setentr=uicontrol('Units','normalized','Style','togglebutton','String','More >>','Position',[0.55 0.04 0.3  
0.05],'Callback',{@opencontrol},'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35  
0.35],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');
```

```
    uicontrol('Style','Text','Units','Normalized','Position',[0.55 0.12 0.12 0.03],'String','Intervalo:','FontSize',13,'ForegroundColor',[0 0  
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');  
    intervalo=uicontrol('Style','Edit','Units','Normalized','Position',[0.685 0.115 0.125 0.04],'String',num2str(T),'FontSize',13,'ForegroundColor',[1 1  
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveinter,1});  
    uicontrol('Style','Text','Units','Normalized','Position',[0.815 0.118 0.03 0.03],'String','s','FontSize',12,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8  
0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
```

```
    annotation('textbox',[0.02 0.925 0.08 0.022],'String','V_1 [m/s]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.02 0.795 0.08 0.022],'String','\phi_1 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.02 0.665 0.08 0.022],'String','\psi_1  
[\circ/s]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.02 0.535 0.08 0.022],'String','\chi_1 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.02 0.405 0.08 0.022],'String','\theta_1  
[\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.02 0.275 0.08 0.022],'String','\eta_1 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.925 0.08 0.022],'String','V_2 [m/s]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.795 0.08 0.022],'String','\phi_2 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.665 0.08 0.022],'String','\psi_1^2  
[\circ/s]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.535 0.08 0.022],'String','\chi_2 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.405 0.08 0.022],'String','\theta_2  
[\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');  
    annotation('textbox',[0.885 0.275 0.08 0.022],'String','\eta_2 [\circ]','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
```

```
    tDA(1)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.89 0.08 0.022],'String',num2str(RV1(1),'%.1f'),'FontSize',11,'ForegroundColor',[0 0  
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDA(2)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.76 0.08 0.022],'String',num2str(RV1(2))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDA(3)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.63 0.08 0.022],'String',num2str(RV1(1))*cos(RV1(2))/RV1(3))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDA(4)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.50 0.08 0.022],'String',num2str(RV1(10))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDA(5)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.37 0.08 0.022],'String',num2str(RV1(11))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDA(6)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.24 0.08 0.022],'String',num2str(RV1(12))*180/pi,'%.  
.0f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
```

```
    tDB(1)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.89 0.08 0.022],'String',num2str(RV2(1),'%.1f'),'FontSize',11,'ForegroundColor',[0  
0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDB(2)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.76 0.08 0.022],'String',num2str(RV2(2))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDB(3)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.63 0.08 0.022],'String',num2str(RV2(1))*cos(RV2(2))/RV2(3))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDB(4)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.50 0.08 0.022],'String',num2str(RV2(10))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDB(5)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.37 0.08 0.022],'String',num2str(RV2(11))*180/pi,'%.  
.1f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');  
    tDB(6)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.24 0.08 0.022],'String',num2str(RV2(12))*180/pi,'%.  
.0f'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
```

```
end
```

```
%-----%
```

```
if callregvoo==1  
    if findobj('Name','Regimes de V6o')>0 regvoo(0,0,1,0,1);  
    else regvoo(0,0,1,1,0);  
    end  
end
```

```
if RV1(9)==1 && RV2(9)==1 set(refresh,'Enable','on'); end
```

```
%-----%
```

```
figure(setestant);
```

```
try
```

```
if atualiza==1
```

```
    phi0=RV1(2);  
    psi0=-atan(Atmo.v/(RV1(1)^2*cos(RV1(2))^2 - Atmo.v^2)^0.5);  
    chi0=RV1(10);  
    chilast=RV1(10);  
    theta0=RV1(11);  
    thetalast=RV1(11);  
    eta0=RV1(12);  
    etalast=RV1(12);
```

```
    a10=cos(eta0)*cos(theta0)*cos(phi0)*cos(psi0)-sin(psi0)*sin(eta0)*cos(chi0)-sin(psi0)*cos(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*cos(psi0)*sin(eta0)*sin(chi0)-  
    sin(phi0)*cos(psi0)*cos(eta0)*sin(theta0)*cos(chi0);  
    a20=cos(eta0)*cos(theta0)*cos(phi0)*sin(psi0)+cos(psi0)*sin(eta0)*cos(chi0)+cos(psi0)*cos(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*sin(psi0)*sin(eta0)*sin(chi0)-  
    sin(phi0)*sin(psi0)*cos(eta0)*sin(theta0)*cos(chi0);  
    a30=cos(eta0)*cos(theta0)*sin(phi0)-cos(phi0)*sin(eta0)*sin(chi0)+cos(phi0)*cos(eta0)*sin(theta0)*cos(chi0);
```

```
    b10=-sin(eta0)*cos(theta0)*cos(phi0)*cos(psi0)-  
    sin(psi0)*cos(eta0)*cos(chi0)+sin(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*cos(psi0)*cos(eta0)*sin(chi0)+sin(phi0)*cos(psi0)*sin(eta0)*sin(theta0)*cos(chi0);  
    b20=-sin(eta0)*cos(theta0)*cos(phi0)*sin(psi0)+cos(psi0)*cos(eta0)*cos(chi0)-  
    cos(psi0)*sin(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*sin(psi0)*cos(eta0)*sin(chi0)+sin(phi0)*sin(psi0)*sin(eta0)*sin(theta0)*cos(chi0);  
    b30=-sin(eta0)*cos(theta0)*sin(phi0)-cos(phi0)*cos(eta0)*sin(chi0)-cos(phi0)*sin(eta0)*sin(theta0)*cos(chi0);
```

```

c10=-sin(theta0)*cos(phi0)*cos(psi0)-cos(theta0)*sin(chi0)*sin(psi0)-cos(theta0)*cos(chi0)*sin(phi0)*cos(psi0);
c20=-sin(theta0)*cos(phi0)*sin(psi0)+cos(theta0)*sin(chi0)*cos(psi0)-cos(theta0)*cos(chi0)*sin(phi0)*sin(psi0);
c30=-sin(theta0)*sin(phi0)+cos(theta0)*cos(chi0)*cos(psi0);

alphadot0=a30*RV1(1)*cos(RV1(2))/RV1(3);
betadot0=b30*RV1(1)*cos(RV1(2))/RV1(3);
gammadot0=c30*RV1(1)*cos(RV1(2))/RV1(3);

if RV1(3)==Inf y0=5*Geom.bw;
else y0=-RV1(3);
end

e0=[RV1(1) psi0 phi0 alphadot0 betadot0 gammadot0 a10 a20 a30 b10 b20 b30 c10 c20 c30 0 y0 5*Geom.bw 0 0 0 Iner.m0 Iner.J10 Iner.J20 Iner.J30];

%-----%

if stall==1 [fCLW fCDW]=CLW_CDW; end
[t e]=rk4_rt(@equacoes,[0 T],e0);

N=size(t,2);
watcher=zeros(N,16);

for i=1:N [watcher(i,1) watcher(i,2) watcher(i,3) watcher(i,4) watcher(i,5) watcher(i,6) watcher(i,7) watcher(i,8) watcher(i,9) watcher(i,10) watcher(i,11)
watcher(i,12) watcher(i,13) watcher(i,14) watcher(i,15) watcher(i,16)]=entrada(t(i),e(i,:)); end

%-----%

delete(estabV);
estabV=axes('Units','normalized','Position',[0.15 0.87 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,e(:,1),'r');

%-----%

delete(estabphi);
estabphi=axes('Units','normalized','Position',[0.15 0.74 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,e(:,3)*180/pi,'r');

%-----%

delete(estabpsi);
estabpsi=axes('Units','normalized','Position',[0.15 0.61 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,7)*180/pi,'r');

%-----%

delete(estabchi);
estabchi=axes('Units','normalized','Position',[0.15 0.48 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,4)*180/pi,'r');

%-----%

delete(estabtheta);
estabtheta=axes('Units','normalized','Position',[0.15 0.35 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,5)*180/pi,'r');

%-----%

delete(estabeta);
estabeta=axes('Units','normalized','Position',[0.15 0.22 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,6)*180/pi,'r');

%-----%

set(traj3d,'Enable','on');

%-----%

if findobj('Name','Entradas')>0

figure(control);

%-----%

delete(estabdeltah);
estabdeltah=axes('Units','normalized','Position',[0.15 0.87 0.7
0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,1)*180/pi,'r');

%-----%

delete(estabdeltaw);
estabdeltaw=axes('Units','normalized','Position',[0.15 0.74 0.7
0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,2)*180/pi,'r');

%-----%

delete(estabdeltav);
estabdeltav=axes('Units','normalized','Position',[0.15 0.61 0.7
0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,3)*180/pi,'r');

%-----%

delete(estabu);
estabu=axes('Units','normalized','Position',[0.15 0.48 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,8),'r');

%-----%

delete(estabv);
estabv=axes('Units','normalized','Position',[0.15 0.35 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,9),'r');

%-----%

delete(estabw);
estabw=axes('Units','normalized','Position',[0.15 0.22 0.7 0.1],'Drawmode','fast','FontSize',8,'FontName','Corbel','Box','on','XTickLabel',[],'XGrid','on');
hold on
plot(t,watcher(:,10),'r');

%-----%

end % entradas

%-----%

```

```

end % atualiza
catch
    Error='Unable to calculate.'
end

%-----%

function tagregvoo(obj,event)

if findobj('Name','Entradas')>0 delete (control); end
if findobj('Name','Regimes de V6o')>0 set(setregvoo,'Visible','on','Tag','1'); end

end

%-----%

function select(obj,event,which)

switch which
case 1
    set(input1,'Value',1);
    set(input2,'Value',0);
    set(input3,'Value',0);
    set(input4,'Value',0);
    entr=1;
case 2
    set(input1,'Value',0);
    set(input2,'Value',1);
    set(input3,'Value',0);
    set(input4,'Value',0);
    entr=2;
case 3
    set(input1,'Value',0);
    set(input2,'Value',0);
    set(input3,'Value',1);
    set(input4,'Value',0);
    entr=3;
case 4
    set(input1,'Value',0);
    set(input2,'Value',0);
    set(input3,'Value',0);
    set(input4,'Value',1);
    entr=4;
end

end

%-----%

function saveinter(obj,event,what)

if what==1

    T=str2double(get(intervalo,'String'));
    if isinf(T) && (T > 0)
        set(refresh,'Enable','off');
    elseif isfinite(T) && (T > 0)
        if RV1(9)==1 && RV2(9)==1 set(refresh,'Enable','on'); end
    else
        set(refresh,'Enable','off');
        T=10;
        set(intervalo,'String',sprintf('%d',T));
    end

else

    trans(1)=str2double(get(tr(1),'String'));
    if ~isfinite(trans(1)) || (trans(1) < 0) || (trans(1) > trans(2))
        trans(1)=2;
        set(tr(1),'String',sprintf('%d',trans(1)));
    end

    trans(2)=str2double(get(tr(2),'String'));
    if ~isfinite(trans(2)) || (trans(2) < trans(1))
        trans(2)=4;
        set(tr(2),'String',sprintf('%d',trans(2)));
    end

    trans(3)=str2double(get(tr(3),'String'));
    if ~isfinite(trans(3)) || (trans(3) < 0) || (trans(3) > trans(4))
        trans(3)=4;
        set(tr(3),'String',sprintf('%d',trans(3)));
    end

    trans(4)=str2double(get(tr(4),'String'));
    if ~isfinite(trans(4)) || (trans(4) < trans(3))
        trans(4)=6;
        set(tr(4),'String',sprintf('%d',trans(4)));
    end

    trans(5)=2*trans(4) - trans(3);

end

end

%-----%

function opencontrol(obj,event)

if findobj('Name','Entradas')>0
    delete(control);
    set(setentr,'String','More >>','Value',0);
else
    control=figure('Units','normalized','Position',[0.46 0.05 0.435
0.90],'MenuBar','none','Resize','off','Name','Entradas','NumberTitle','off','DeleteFcn',{@opencontrol});
    set(setentr,'String','Less <<','Value',1);

    establdeltaht=axes('Units','normalized','Position',[0.15 0.87 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1
1 1]);
    establdeltaw=axes('Units','normalized','Position',[0.15 0.74 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1
1 1]);
    establdeltavt=axes('Units','normalized','Position',[0.15 0.61 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1
1 1]);
    estabu=axes('Units','normalized','Position',[0.15 0.48 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1
1]);
    estabv=axes('Units','normalized','Position',[0.15 0.35 0.7 0.1],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1 1]);
    estabw=axes('Units','normalized','Position',[0.15 0.22 0.7 0.1],'XTickLabel',[],'YTickLabel',[],'XTick',[],'YTick',[],'XColor',[1 1 1],'YColor',[1 1
1]);

    input1=uicontrol('Units','normalized','Style','radiobutton','String','Degrau','Callback',{@select,1},'Position',[0.43 0.13 0.2
0.03],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');
    input2=uicontrol('Units','normalized','Style','radiobutton','String','Rampa','Callback',{@select,2},'Position',[0.43 0.1 0.2
0.03],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');
    input3=uicontrol('Units','normalized','Style','radiobutton','String','Parabola','Callback',{@select,3},'Position',[0.43 0.07 0.2
0.03],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');

```

```

input4=uicontrol('Units','normalized','Style','radiobutton','String','Sigmoidal','Callback',{@select,4},'Position',[0.43 0.04 0.2
0.03],'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel');
switch entr
case 1
set(input1,'Value',1);
case 2
set(input2,'Value',1);
case 3
set(input3,'Value',1);
case 4
set(input4,'Value',1);
end

annotation('textbox',[0.02 0.925 0.08 0.022],'String','\delta_h_t_1
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.02 0.795 0.08 0.022],'String','\delta_w_1
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.02 0.665 0.08 0.022],'String','\delta_v_t_1
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');

tdA(7)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.89 0.08 0.022],'String',num2str(RV1(4))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
tdA(8)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.76 0.08 0.022],'String',num2str(RV1(5))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
tdA(9)=uicontrol('Style','Text','Units','Normalized','Position',[0.02 0.63 0.08 0.022],'String',num2str(RV1(6))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');

annotation('textbox',[0.885 0.925 0.08 0.022],'String','\delta_h_t_2
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.885 0.795 0.08 0.022],'String','\delta_w_2
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.885 0.665 0.08 0.022],'String','\delta_v_t_2
{\circ}','FontSize',9,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');

tdB(7)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.89 0.08 0.022],'String',num2str(RV2(4))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
tdB(8)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.76 0.08 0.022],'String',num2str(RV2(5))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
tdB(9)=uicontrol('Style','Text','Units','Normalized','Position',[0.885 0.63 0.08 0.022],'String',num2str(RV2(6))*180/pi,'%-
.lf'),'FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');

annotation('textbox',[0.02 0.535 0.08 0.022],'String','u [m/s]','FontSize',10,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.02 0.405 0.08 0.022],'String','v [m/s]','FontSize',10,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');
annotation('textbox',[0.02 0.275 0.08 0.022],'String','w [m/s]','FontSize',10,'EdgeColor','none','Fontname','Corbel','HorizontalAlignment','Center');

uicontrol('Style','Text','Units','Normalized','Position',[0.213 0.13 0.145 0.03],'String','Comandos','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
uicontrol('Style','Text','Units','Normalized','Position',[0.145 0.085 0.07 0.03],'String','Inicio:','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Right');
uicontrol('Style','Text','Units','Normalized','Position',[0.15 0.04 0.065 0.03],'String','Fim:','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Right');
tr(1)=uicontrol('Style','Edit','Units','Normalized','Position',[0.225 0.085 0.125 0.04],'String',num2str(trans(1)),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveinter,2});
tr(2)=uicontrol('Style','Edit','Units','Normalized','Position',[0.225 0.04 0.125 0.04],'String',num2str(trans(2)),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveinter,2});
uicontrol('Style','Text','Units','Normalized','Position',[0.35 0.085 0.03 0.03],'String','s','FontSize',12,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
uicontrol('Style','Text','Units','Normalized','Position',[0.35 0.04 0.03 0.03],'String','s','FontSize',12,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');

uicontrol('Style','Text','Units','Normalized','Position',[0.673 0.13 0.125 0.03],'String','Rajada','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
uicontrol('Style','Text','Units','Normalized','Position',[0.595 0.085 0.07 0.03],'String','Inicio:','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Right');
uicontrol('Style','Text','Units','Normalized','Position',[0.60 0.04 0.065 0.03],'String','Pico:','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Right');
tr(3)=uicontrol('Style','Edit','Units','Normalized','Position',[0.675 0.085 0.125 0.04],'String',num2str(trans(3)),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveinter,2});
tr(4)=uicontrol('Style','Edit','Units','Normalized','Position',[0.675 0.04 0.125 0.04],'String',num2str(trans(4)),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveinter,2});
uicontrol('Style','Text','Units','Normalized','Position',[0.8 0.085 0.03 0.03],'String','s','FontSize',12,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');
uicontrol('Style','Text','Units','Normalized','Position',[0.8 0.04 0.03 0.03],'String','s','FontSize',12,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center');

end

end

%-----
end

```

```

function [edot]=equacoes(t,e)
% Equacoes Diferenciais
%
% e(1)=V;
% e(2)=psi;
% e(3)=phi;
% e(4)=alphadot;
% e(5)=betadot;
% e(6)=gammadot;
% e(7)=a1;
% e(8)=a2;
% e(9)=a3;
% e(10)=b1;
% e(11)=b2;
% e(12)=b3;
% e(13)=c1;
% e(14)=c2;
% e(15)=c3;
% e(16)=x;
% e(17)=y;
% e(18)=z;
% e(19)=alpha;
% e(20)=beta;
% e(21)=gamma;
% e(22)=m;
% e(23)=J1;
% e(24)=J2;
% e(25)=J3;

global g Geom Aero Prop Iner Atmo fCLW fCDw stall

[deltaht deltaw deltavt chi theta eta psidot u v w udot vdot wdot]=entrada(t,e);

%-----§

if stall==1
    CLW=fCLW(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
    CDW=fCDW(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
else
    CLW=Aero.CLalphaw*(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
    CDW=CLW^2/Aero.ArW/pi/Aero.ew + Aero.CD0w;
end
CMrRw=(Aero.CMrRw*0.5*(sin(theta)*cos(eta)*e(4)-sin(theta)*sin(eta)*e(5)+cos(theta)*e(6))*Geom.bw - Aero.CMrRw*(1/2)*(cos(eta)*cos(theta)*e(4)-sin(eta)*cos(theta)*e(5)-sin(theta)*e(6))*Geom.bw)/e(1);
CMrRw=(Aero.CMrRw*0.5*(sin(theta)*cos(eta)*e(4)-sin(theta)*sin(eta)*e(5)+cos(theta)*e(6))*Geom.bw - Aero.CMrRw*(1/2)*(cos(eta)*cos(theta)*e(4)-sin(eta)*cos(theta)*e(5)-sin(theta)*e(6))*Geom.bw)/e(1);

CL0ht=Aero.CLalpaht*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180);
CDI0ht=(Aero.CLalpaht*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180))^2/pi/Aero.Arht/Aero.eht;
CDideltaht=2*Aero.CLalpaht*Aero.CLdeltaht*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180)/pi/Aero.Arht/Aero.eht;

CLf=Aero.CL0f + Aero.CLalphaf*(theta - Iner.theta0*pi/180);
CDF=Aero.CL0f + Aero.CDalpaf*(theta - Iner.theta0*pi/180);
CMPf=Aero.CMp0f + Aero.CMpalphaf*(theta - Iner.theta0*pi/180);

E=Prop.p*Atmo.rho/Prop.rho0*Prop.a2*(e(1)*(cos(eta)*cos(theta)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - sin(theta)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180))^2 + Prop.p*Atmo.rho/Prop.rho0*Prop.a1*e(1)*(cos(eta)*cos(theta)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - sin(theta)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)) + Prop.p*Atmo.rho/Prop.rho0*Prop.a0;
W=e(22)*g;

Lw=0.5*Atmo.rho*e(1)^2*Geom.Sw*CLw;
Dw=0.5*Atmo.rho*e(1)^2*Geom.Sw*CDw;
Mrw=0.5*Atmo.rho*e(1)^2*Geom.Sw*Aero.cw*(Aero.CMrdeltaht*deltaw + CMrRw);
Myw=0.5*Atmo.rho*e(1)^2*Geom.Sw*Aero.cw*(Aero.CMydeltaht*deltaw + CMyRw);
Mpw=0.5*Atmo.rho*e(1)^2*Geom.Sw*Aero.cw*Aero.CMp0w;

Lht=0.5*Atmo.rho*e(1)^2*Geom.Sht*(CL0ht + Aero.CLdeltaht*deltaht);
Dht=0.5*Atmo.rho*e(1)^2*Geom.Sht*(Aero.CD0ht + CDI0ht + CDideltaht*deltaht);
Mphht=0.5*Atmo.rho*e(1)^2*Geom.Sht*Aero.cht*(Aero.CMp0ht + Aero.CMpdeltaht*deltaht);

Lf=0.5*Atmo.rho*e(1)^2*Geom.Sf*CLf;
Df=0.5*Atmo.rho*e(1)^2*Geom.Sf*CDF;
Mpf=0.5*Atmo.rho*e(1)^2*Geom.Sf*Aero.cf*CMpf;

Lvt=0.5*Atmo.rho*e(1)^2*Geom.Svt*(Aero.CLalphavt*eta + Aero.CLdeltavt*deltavt);
Dvt=0.5*Atmo.rho*e(1)^2*Geom.Svt*Aero.CD0vt;

Ft=E*(cos(eta)*cos(theta)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - sin(theta)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)) - Dw - Dht - Dvt - Df - W*sin(e(3));
Fr=E*(cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*sin(eta)*cos(chi) + cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*sin(chi)*sin(eta)*sin(chi) + cos(theta)*sin(chi)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)) + (Lw + Lht + Lf)*((cos(eta)*sin(chi) + sin(eta)*sin(theta)*cos(chi))/(sin(theta)^2 + cos(eta)^2*cos(theta)^2^0.5);
Fn=E*(cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*cos(eta)*sin(theta)*cos(chi) + cos(theta)*cos(chi)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*sin(eta)*sin(chi) + (Lw + Lht + Lf)*((cos(eta)*cos(chi) - sin(eta)*sin(theta)*sin(chi))/(sin(theta)^2 + cos(eta)^2*cos(theta)^2^0.5) - W*cos(e(3));

% Verifica Regime Permanente
% Ft=0;
% Fr=e(22)*RV1(1)^2*cos(RV1(2))^2/RV1(3);
% Fn=0;

edot(1)=(Ft - Prop.p*Iner.dmdt*e(1))/e(22) - cos(e(3))*cos(e(2))*(Prop.p*Iner.dmdt/e(22)*u + udot) - cos(e(3))*sin(e(2))*(Prop.p*Iner.dmdt/e(22)*v + vdot) - sin(e(3))*(Prop.p*Iner.dmdt/e(22)*w + wdot);
if cos(e(3))==0
    edot(2)=0;
else
    edot(2)=(Fr/e(1)/cos(e(3))/e(22) + sin(e(2))*(Prop.p*Iner.dmdt/e(22)*u + udot) - cos(e(2))*(Prop.p*Iner.dmdt/e(22)*v + vdot);
end
edot(3)=Fn/e(1)/e(22) + sin(e(3))*cos(e(2))*(Prop.p*Iner.dmdt/e(22)*u + udot) + sin(e(3))*sin(e(2))*(Prop.p*Iner.dmdt/e(22)*v + vdot) - cos(e(3))*(Prop.p*Iner.dmdt/e(22)*w + wdot);

% Direcoes de Momentos complicadas
Ta=- Dw*Geom.lw*sin(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
- Dht*Geom.lht*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
- Lvt*Geom.lvt*sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*cos(eta)...
- Dvt*Geom.lvt*sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
- Df*Geom.lf*sin(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
- Mrw;

Tb=Lw*Geom.lw*(sin(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*sin(theta) - cos(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta))/(sin(theta)^2 + cos(eta)^2*cos(theta)^2^0.5...
+ Dw*Geom.lw*(-sin(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta) - cos(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*sin(eta))...
- Mpw...
+ Lht*Geom.lht*(sin(Geom.gammat*pi/180 - Iner.theta0*pi/180)*sin(theta) - cos(Geom.gammat*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta))/(sin(theta)^2 + cos(eta)^2*cos(theta)^2^0.5...
+ Dht*Geom.lht*(-sin(Geom.gammat*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta) - cos(Geom.gammat*pi/180 - Iner.theta0*pi/180)*sin(eta))...
- Mphht...
+ Lvt*Geom.lvt*sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*sin(eta)...
+ Dvt*Geom.lvt*(-sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta) - cos(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*sin(eta))...
+ Lf*Geom.lf*(sin(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*sin(theta) - cos(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta))/(sin(theta)^2 + cos(eta)^2*cos(theta)^2^0.5...
+ Df*Geom.lf*(-sin(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*cos(eta)*cos(theta) - cos(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*sin(eta))...
- Mpf...
- E*Geom.lt*sin(Geom.betat*pi/180);

Tc=Dw*Geom.lw*cos(Geom.gammaw*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...

```

```

+ Dht*Geom.lht*cos(Geom.gamahrt*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
+ Lvt*Geom.lvt*cos(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*cos(eta)...
+ Dvt*Geom.lvt*cos(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
+ Df*Geom.lf*cos(Geom.gammaf*pi/180 - Iner.theta0*pi/180)*sin(eta)*cos(theta)...
+ Myw;

% Direcoes de Momentos simplificadas
% Ta=-Mrw - Geom.lvt*sin(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*Lvt;
% Tb=-Geom.lw*Lw*cos(theta + Geom.gammaw*pi/180 - Iner.theta0*pi/180) - Geom.lw*Dw*sin(theta + Geom.gammaw*pi/180 - Iner.theta0*pi/180) - Mpw -
Geom.lht*Lht*cos(theta + Geom.gamahrt*pi/180 - Iner.theta0*pi/180) - Geom.lht*Dht*sin(theta + Geom.gamahrt*pi/180 - Iner.theta0*pi/180) - Mph -
Geom.lvt*Dvt*sin(theta + Geom.gammavt*pi/180 - Iner.theta0*pi/180) - Geom.lf*Lf*cos(theta + Geom.gammaf*pi/180 - Iner.theta0*pi/180) - Geom.lf*Df*sin(theta +
Geom.gammaf*pi/180 - Iner.theta0*pi/180) - Mpf - E*Geom.lt*sin(Geom.betat*pi/180);
% Tc=Myw + Geom.lvt*cos(Geom.gammavt*pi/180 - Iner.theta0*pi/180)*Lvt;

% Verifica Regime Permanente
% Ta=-e(24)*e(12)*edot(2)^2*e(7)*e(11) + e(24)*e(12)*edot(2)^2*e(8)*e(10) - e(25)*e(15)*edot(2)^2*e(7)*e(14) + e(25)*e(15)*edot(2)^2*e(8)*e(13);
% Tb=-e(23)*e(9)*edot(2)^2*e(8)*e(10) + e(23)*e(9)*edot(2)^2*e(7)*e(11) - e(25)*e(15)*edot(2)^2*e(10)*e(14) + e(25)*e(15)*edot(2)^2*e(11)*e(13);
% Tc=-e(23)*e(9)*edot(2)^2*e(8)*e(13) + e(23)*e(9)*edot(2)^2*e(7)*e(14) - e(24)*e(12)*edot(2)^2*e(11)*e(13) + e(24)*e(12)*edot(2)^2*e(10)*e(14);

edot(4)=(Ta - e(24)*e(5)*e(12)*e(7)*e(14)*e(6) + e(24)*e(5)*e(11)*e(7)*e(15)*e(6) - e(24)*e(5)*e(10)*e(8)*e(15)*e(6) + e(24)*e(5)*e(12)*e(8)*e(13)*e(6) -
e(24)*e(5)*e(11)*e(9)*e(13)*e(6) + e(24)*e(5)*e(10)*e(9)*e(14)*e(6) - e(25)*e(6)*e(15)*e(7)*e(11)*e(5) + e(25)*e(6)*e(14)*e(7)*e(12)*e(5) -
e(25)*e(6)*e(13)*e(8)*e(12)*e(5) + e(25)*e(6)*e(15)*e(8)*e(10)*e(5) - e(25)*e(6)*e(14)*e(9)*e(10)*e(5) + e(25)*e(6)*e(13)*e(9)*e(11)*e(5) -
Prop.p*Iner.dJ1dt*e(4))/e(23);
edot(5)=(Tb - e(23)*e(4)*e(10)*e(9)*e(14)*e(6) + e(23)*e(4)*e(10)*e(8)*e(15)*e(6) - e(23)*e(4)*e(11)*e(7)*e(15)*e(6) + e(23)*e(4)*e(11)*e(9)*e(13)*e(6) -
e(23)*e(4)*e(12)*e(8)*e(13)*e(6) + e(23)*e(4)*e(12)*e(7)*e(14)*e(6) - e(25)*e(6)*e(15)*e(10)*e(8)*e(4) + e(25)*e(6)*e(14)*e(10)*e(9)*e(4) -
e(25)*e(6)*e(13)*e(11)*e(9)*e(4) + e(25)*e(6)*e(15)*e(11)*e(7)*e(4) - e(25)*e(6)*e(14)*e(12)*e(7)*e(4) + e(25)*e(6)*e(13)*e(12)*e(8)*e(4) -
Prop.p*Iner.dJ2dt*e(5))/e(24);
edot(6)=(Tc - e(23)*e(4)*e(13)*e(9)*e(11)*e(5) + e(23)*e(4)*e(13)*e(8)*e(12)*e(5) - e(23)*e(4)*e(14)*e(7)*e(12)*e(5) + e(23)*e(4)*e(14)*e(9)*e(10)*e(5) -
e(23)*e(4)*e(15)*e(8)*e(10)*e(5) + e(23)*e(4)*e(15)*e(7)*e(11)*e(5) - e(24)*e(5)*e(13)*e(12)*e(8)*e(4) + e(24)*e(5)*e(13)*e(11)*e(9)*e(4) -
e(24)*e(5)*e(14)*e(10)*e(9)*e(4) + e(24)*e(5)*e(14)*e(12)*e(7)*e(4) - e(24)*e(5)*e(15)*e(11)*e(7)*e(4) + e(24)*e(5)*e(15)*e(10)*e(8)*e(4) -
Prop.p*Iner.dJ3dt*e(6))/e(25);

% Cálculo direto das derivas dos versores
% edot(7)=e(9)*e(11)*e(5) + e(9)*e(14)*e(6) - e(8)*e(12)*e(5) - e(8)*e(15)*e(6);
% edot(8)=e(7)*e(12)*e(5) + e(7)*e(15)*e(6) - e(9)*e(10)*e(5) - e(9)*e(13)*e(6);
% edot(9)=e(8)*e(10)*e(5) + e(8)*e(13)*e(6) - e(7)*e(11)*e(5) - e(7)*e(14)*e(6);
%
% edot(10)=e(12)*e(8)*e(4) + e(12)*e(14)*e(6) - e(11)*e(9)*e(4) - e(11)*e(15)*e(6);
% edot(11)=e(10)*e(9)*e(4) + e(10)*e(15)*e(6) - e(12)*e(7)*e(4) - e(12)*e(13)*e(6);
% edot(12)=e(11)*e(7)*e(4) + e(11)*e(13)*e(6) - e(10)*e(8)*e(4) - e(10)*e(14)*e(6);
%
% edot(13)=e(15)*e(8)*e(4) + e(15)*e(11)*e(5) - e(14)*e(9)*e(4) - e(14)*e(12)*e(5);
% edot(14)=e(13)*e(9)*e(4) + e(13)*e(12)*e(5) - e(15)*e(7)*e(4) - e(15)*e(10)*e(5);
% edot(15)=e(14)*e(7)*e(4) + e(14)*e(10)*e(5) - e(13)*e(8)*e(4) - e(13)*e(11)*e(5);

% Redução de erro numérico
edot(7)=e(8)*e(4)*e(10)*e(14) - e(8)*e(4)*e(11)*e(13) + e(11)*e(5)*e(10)*e(14) - e(11)^2*e(5)*e(13) + e(14)^2*e(6)*e(10) - e(14)*e(6)*e(11)*e(13) -
e(9)*e(4)*e(12)*e(11) + e(9)*e(4)*e(10)*e(15) - e(12)^2*e(5)*e(13) + e(12)*e(5)*e(10)*e(15) - e(15)*e(6)*e(12)*e(13) + e(15)^2*e(6)*e(10);
edot(8)=e(9)*e(4)*e(13)*e(11) - e(9)*e(4)*e(15)*e(12)*e(14) + e(12)*e(5)*e(11)*e(15) - e(12)^2*e(5)*e(14) + e(15)^2*e(6)*e(11) - e(15)*e(6)*e(12)*e(14) -
e(7)*e(4)*e(10)*e(14) + e(7)*e(4)*e(11)*e(13) - e(10)^2*e(5)*e(14) + e(10)*e(5)*e(11)*e(13) - e(13)*e(6)*e(10)*e(14) + e(13)^2*e(6)*e(11);
edot(9)=e(7)*e(4)*e(12)*e(13) - e(7)*e(4)*e(10)*e(15) + e(10)*e(5)*e(12)*e(13) - e(10)^2*e(5)*e(15) + e(13)^2*e(6)*e(12) - e(13)*e(6)*e(10)*e(15) -
e(8)*e(4)*e(11)*e(15) + e(8)*e(4)*e(12)*e(14) - e(11)^2*e(5)*e(15) + e(11)*e(5)*e(12)*e(14) - e(14)*e(6)*e(11)*e(15) + e(14)^2*e(6)*e(12);
edot(10)=e(8)^2*e(4)*e(13) - e(8)*e(4)*e(14)*e(7) + e(11)*e(5)*e(13)*e(8) - e(11)*e(5)*e(14)*e(7) + e(14)*e(6)*e(13)*e(8) - e(14)^2*e(6)*e(7) -
e(9)*e(4)*e(15)*e(7) + e(9)^2*e(4)*e(13) - e(12)*e(5)*e(15)*e(7) + e(12)*e(5)*e(13)*e(9) - e(15)^2*e(6)*e(7) + e(15)*e(6)*e(13)*e(9);
edot(11)=e(9)^2*e(4)*e(14) - e(9)*e(4)*e(15)*e(8) + e(12)*e(5)*e(14)*e(9) - e(12)*e(5)*e(15)*e(8) + e(15)*e(6)*e(14)*e(9) - e(15)^2*e(6)*e(8) -
e(7)*e(4)*e(13)*e(8) + e(7)^2*e(4)*e(14) - e(10)*e(5)*e(13)*e(8) + e(10)*e(5)*e(14)*e(7) - e(13)^2*e(6)*e(8) + e(13)*e(6)*e(14)*e(7);
edot(12)=e(7)^2*e(4)*e(15) - e(7)*e(4)*e(13)*e(9) + e(10)*e(5)*e(15)*e(7) - e(10)*e(5)*e(13)*e(9) + e(13)*e(6)*e(15)*e(7) - e(13)^2*e(6)*e(9) -
e(8)*e(4)*e(14)*e(9) + e(8)^2*e(4)*e(15) - e(11)*e(5)*e(14)*e(9) + e(11)*e(5)*e(15)*e(8) - e(14)^2*e(6)*e(9) + e(14)*e(6)*e(15)*e(8);

edot(13)=e(8)*e(4)*e(7)*e(11) - e(8)^2*e(4)*e(10) + e(11)^2*e(5)*e(7) - e(11)*e(5)*e(8)*e(10) + e(14)*e(6)*e(7)*e(11) - e(14)*e(6)*e(8)*e(10) -
e(9)^2*e(4)*e(10) + e(9)*e(4)*e(7)*e(12) - e(12)*e(5)*e(9)*e(10) + e(12)^2*e(5)*e(7) - e(15)*e(6)*e(9)*e(10) + e(15)*e(6)*e(7)*e(12);
edot(14)=e(9)*e(4)*e(8)*e(12) - e(9)^2*e(4)*e(11) + e(12)^2*e(5)*e(8) - e(12)*e(5)*e(9)*e(11) + e(15)*e(6)*e(8)*e(12) - e(15)*e(6)*e(9)*e(11) -
e(7)^2*e(4)*e(11) + e(7)*e(4)*e(8)*e(10) - e(10)*e(5)*e(7)*e(11) + e(10)^2*e(5)*e(8) - e(13)*e(6)*e(7)*e(11) + e(13)*e(6)*e(8)*e(10);
edot(15)=e(7)*e(4)*e(9)*e(10) - e(7)^2*e(4)*e(12) + e(10)^2*e(5)*e(9) - e(10)*e(5)*e(7)*e(12) + e(13)*e(6)*e(9)*e(10) - e(13)*e(6)*e(7)*e(12) -
e(8)^2*e(4)*e(12) + e(8)*e(4)*e(9)*e(11) - e(11)*e(5)*e(8)*e(12) + e(11)^2*e(5)*e(9) - e(14)*e(6)*e(8)*e(12) + e(14)*e(6)*e(9)*e(11);

% Exemplo de ruído numérico
% edot(4)=(Ta - e(23)*e(4)*e(7)*edot(7) - e(23)*e(4)*e(8)*edot(8) - e(23)*e(4)*e(9)*edot(8) - e(24)*e(5)*e(7)*edot(10) - e(24)*e(5)*e(8)*edot(11) -
e(24)*e(5)*e(9)*edot(12) - e(25)*e(6)*e(7)*edot(13) - e(25)*e(6)*e(8)*edot(14) - e(25)*e(6)*e(9)*edot(15) - Prop.p*Iner.dJ1dt*e(4))/e(23);
% edot(5)=(Tb - e(23)*e(4)*e(10)*edot(7) - e(23)*e(4)*e(11)*edot(8) - e(23)*e(4)*e(12)*edot(8) - e(24)*e(5)*e(10)*edot(10) - e(24)*e(5)*e(11)*edot(11) -
e(24)*e(5)*e(12)*edot(12) - e(25)*e(6)*e(10)*edot(13) - e(25)*e(6)*e(11)*edot(14) - e(25)*e(6)*e(12)*edot(15) - Prop.p*Iner.dJ2dt*e(5))/e(24);
% edot(6)=(Tc - e(23)*e(4)*e(13)*edot(7) - e(23)*e(4)*e(14)*edot(8) - e(23)*e(4)*e(15)*edot(8) - e(24)*e(5)*e(13)*edot(10) - e(24)*e(5)*e(14)*edot(11) -
e(24)*e(5)*e(15)*edot(12) - e(25)*e(6)*e(13)*edot(13) - e(25)*e(6)*e(14)*edot(14) - e(25)*e(6)*e(15)*edot(15) - Prop.p*Iner.dJ3dt*e(6))/e(25);

edot(16)=e(1)*cos(e(3))*cos(e(2)) + u;
edot(17)=e(1)*cos(e(3))*sin(e(2)) + v;
edot(18)=e(1)*sin(e(3)) + w;

edot(19)=e(4);
edot(20)=e(5);
edot(21)=e(6);

edot(22)=Prop.p*Iner.dmdt;
edot(23)=Prop.p*Iner.dJ1dt;
edot(24)=Prop.p*Iner.dJ2dt;
edot(25)=Prop.p*Iner.dJ3dt;

%-----%
end

```

```

function simulate(obj,event)
% Simulador

global T trt ert Geom Aero Iner Atmo fCLw fCDw stall RV1 solid joy mode simul virtual seeflight realtime history chilast thetalast etalast

if findobj('Name','Simulador')>0 delete(simul); end

try

joy=vrjoystick(1);
mode=1;

simul=figure('Units','normalized','Renderer','OpenGL','Position',[0.01 0.05 0.885
0.839],'Resize','off','Name','Simulador','NumberTitle','off','DeleteFcn',{@deljoy});
realtime=annotation('textbox',[0.92 0.05 0.05 0.04],'Units','normalized','HorizontalAlignment','Right');

%-----%

seeflight=axes('Units','normalized','Position',[0.05 0.07 0.81 0.1],'DrawMode','normal','FontSize',9,'Color','none','XLim',[0 10],'YLim',[-
max((abs(RV1(4))+abs(Aero.Dht)) (abs(RV1(5))+abs(Aero.Dw)) (abs(RV1(6))+abs(Aero.Dvt)))]*180/pi max((abs(RV1(4))+abs(Aero.Dht)) (abs(RV1(5))+abs(Aero.Dw))
(abs(RV1(6))+abs(Aero.Dvt)))]*180/pi));
xlabel('s');
ylabel('\circ');

history=[];
history.size=0;

%-----%

solid.ang=3;
ambient=0.5;
diffuse=0.7;
specular=0.9;

X1=-Geom.bw*100;
X2=Geom.bw*100;
Y1=-Geom.bw*100;
Y2=Geom.bw*100;
Z1=-10;
Z2=Geom.bw*100;
virtual=axes('Units','normalized','Position',[0.05 0.05 0.9 0.9],'DrawMode','normal','XLim',[X1 X2],'YLim',[Y1 Y2],'ZLim',[Z1
Z2],'Projection','orthographic','CameraPosition',[X1 Y1 Z2],'Color',[0.6 0.8 1]);
axis equal
hold on
xlabel('x [m]');
ylabel('y [m]');
zlabel('z [m]');

K=100;
[X_g Y_g Z_g]=peaks(K);
X_g=X1:(X2 - X1)/(K - 1):X2;
Y_g=Y1:(Y2 - Y1)/(K - 1):Y2;
surface(X_g,Y_g,Z_g/10,'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
colormap(summer);

light('Position',[X1 Y1 Z2],'Style','infinite');

%-----%

phi0=RV1(2);
psi0=-atan(Atmo.v/(RV1(1)^2*cos(RV1(2))^2 - Atmo.v^2)^0.5);
chi0=RV1(10);
chilast=RV1(10);
theta0=RV1(11);
thetalast=RV1(11);
eta0=RV1(12);
etalast=RV1(12);

a10=cos(eta0)*cos(theta0)*cos(phi0)*cos(psi0)-sin(psi0)*sin(eta0)*cos(chi0)-sin(psi0)*cos(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*cos(psi0)*sin(eta0)*sin(chi0)-
sin(phi0)*cos(psi0)*cos(eta0)*sin(theta0)*cos(chi0);
a20=cos(eta0)*cos(theta0)*cos(phi0)*sin(psi0)+cos(psi0)*sin(eta0)*cos(chi0)+cos(psi0)*cos(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*sin(psi0)*sin(eta0)*sin(chi0)-
sin(phi0)*sin(psi0)*cos(eta0)*sin(theta0)*cos(chi0);
a30=cos(eta0)*cos(theta0)*sin(phi0)-cos(phi0)*sin(eta0)*sin(chi0)+cos(phi0)*cos(eta0)*sin(theta0)*cos(chi0);

b10=-sin(eta0)*cos(theta0)*cos(phi0)*cos(psi0)-
sin(psi0)*cos(eta0)*cos(chi0)+sin(psi0)*sin(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*cos(psi0)*cos(eta0)*sin(chi0)+sin(phi0)*cos(psi0)*sin(eta0)*sin(theta0)*cos(chi0);
b20=-sin(eta0)*cos(theta0)*cos(phi0)*sin(psi0)+cos(psi0)*cos(eta0)*cos(chi0)-
cos(psi0)*sin(eta0)*sin(theta0)*sin(chi0)+sin(phi0)*sin(psi0)*cos(eta0)*sin(chi0)+sin(phi0)*sin(psi0)*sin(eta0)*sin(theta0)*cos(chi0);
b30=-sin(eta0)*cos(theta0)*sin(phi0)-cos(phi0)*cos(eta0)*sin(chi0)-cos(phi0)*sin(eta0)*sin(theta0)*cos(chi0);

c10=-sin(theta0)*cos(phi0)*cos(psi0)-cos(theta0)*sin(chi0)*sin(psi0)-cos(theta0)*cos(chi0)*sin(phi0)*cos(psi0);
c20=-sin(theta0)*cos(phi0)*sin(psi0)+cos(theta0)*sin(chi0)*cos(psi0)-cos(theta0)*cos(chi0)*sin(phi0)*sin(psi0);
c30=-sin(theta0)*sin(phi0)+cos(theta0)*cos(chi0)*cos(phi0);

alphanadot0=a30*RV1(1)*cos(RV1(2))/RV1(3);
betadot0=b30*RV1(1)*cos(RV1(2))/RV1(3);
gammadot0=c30*RV1(1)*cos(RV1(2))/RV1(3);

e0=[RV1(1) psi0 phi0 alphanadot0 betadot0 gammadot0 a10 a20 a30 b10 b20 b30 c10 c20 c30 0 0 40*Geom.bw 0 0 0 Iner.m0 Iner.J10 Iner.J20 Iner.J30];

%-----%

set(virtual,'CameraTarget',[e0(16) e0(17) e0(18)],'CameraViewAngle',solid.ang);

if stall==1 [fCLw fCDw]=CLw_CDw; end
[trt ert]=rk4_rt(@equacoes,[0 T],e0);

%-----%

catch
Error='No joystick plugged.'
end

function deljoy(obj,event)

close(joy);

end

end

```

```

function [tout yout]=rk4_rt(ode,tspan,y0)
% Runge-Kutta 4

global FPS DPS fps dps Geom Aero Prop Iner Atmo mode virtual solid realtime seeflight history fCLw fCDw stall

if DPS<FPS || rem(DPS,FPS)~=0 error('Error: Discretizacao'); end
if size(tspan,2)<2
    error('Error: Input t span');
elseif size(tspan,2)==2
    if isfinite(tspan(2))==1

        intinf=0;

        ti=tspan(1);
        tf=tspan(2);

        nout=round(FPS*(tf - ti)) + 1;
        neval=(nout - 1)*DPS/FPS + 1;

        dps=(neval - 1)/(tf - ti);
        fps=(nout - 1)/(tf - ti);

        h=(tf - ti)/(neval - 1);
        dt=(tf - ti)/(nout - 1);

        tout=ti:(tf - ti)/(nout - 1):tf;
        yout=zeros(nout,size(y0,2));

    else

        intinf=1;

        ti=tspan(1);
        tf=Inf;

        dps=DPS;
        fps=FPS;

        h=1/DPS;
        dt=1/FPS;

        tout=ti;
        yout=y0;

    end
else
    error('Error: t span must be = [ti tf]');
end

% ----- %

if mode==1

    axes(virtual);

    ambient=0.5;
    diffuse=0.7;
    specular=0.9;

    s3d_cg=patch('Faces',solid.f_cg,'Vertices',solid.v_cg,'FaceColor',[0.6 0.6
0.6],'EdgeColor','none','Facelighting','gouraud','AmbientStrength',diffuse,'SpecularStrength',specular);
    s3d_w=patch('Faces',solid.f_w,'Vertices',solid.v_w,'FaceColor',[1 0
0],'EdgeColor','none','Facelighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    s3d_ht=patch('Faces',solid.f_ht,'Vertices',solid.v_ht,'FaceColor',[1 0
0],'EdgeColor','none','Facelighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    s3d_vt=patch('Faces',solid.f_vt,'Vertices',solid.v_vt,'FaceColor',[0 0
1],'EdgeColor','none','Facelighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    s3d_t=patch('Faces',solid.f_t,'Vertices',solid.v_t,'FaceColor',[0 0
1],'EdgeColor','none','Facelighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    s3d_vel=line([0 1], [0 1], [0 1],'Color',[0.7 0.7 0.7]);

    V_cg=zeros(size(solid.v_cg));
    V_w=zeros(size(solid.v_w));
    V_ht=zeros(size(solid.v_ht));
    V_vt=zeros(size(solid.v_vt));
    V_t=zeros(size(solid.v_t));

    % ----- %

    n=15;
    [X_wd Y_wd Z_wd]=sphere(n);
    points=surf2patch(X_wd,Y_wd,Z_wd);

    X_wd=1.5*Geom.bw*points.vertices(:,1);
    Y_wd=1.5*Geom.bw*points.vertices(:,2);
    Z_wd=1.5*Geom.bw*points.vertices(:,3);
    U_wd=ones(size(points.vertices,1),1);
    V_wd=ones(size(points.vertices,1),1);
    W_wd=ones(size(points.vertices,1),1);

    wind3d=quiver3(X_wd,Y_wd,Z_wd,U_wd,V_wd,W_wd,0,'Color',[1 1 1]);

    % ----- %

    n=20;
    v_bum=zeros(n,3);
    V_bum=zeros(n,3);
    F_bum=zeros(n,3);
    C_bum=zeros(n,3);
    for j=1:n
        v_bum(j,1)=cos(2*pi*rand)*cos(2*pi*rand);
        v_bum(j,2)=cos(2*pi*rand)*sin(2*pi*rand);
        v_bum(j,3)=sin(2*pi*rand);
        C_bum(j,:)=[rand rand/4 0];
    end
    for j=1:round(1.5*n)
        F_bum(j,:)=[ceil(n*rand) ceil(n*rand) ceil(n*rand)];
    end
    buumm=patch('Faces',F_bum,'Vertices',v_bum,'FaceVertexCData',C_bum,'EdgeColor','none','FaceColor','flat','Visible','off');

end

% ----- %

try

    s=1;
    t=ti;
    ts=ti;
    y=y0;
    tic;
    while t<(tf + h)

        % Runge-Kutta

        k1=h*ode(t,y);
        k2=h*ode(t + h/2,y + k1/2);

```

```

k3=h*ode(t + h/2,y + k2/2);
k4=h*ode(t + h,y + k3);

y=y + (k1 + 2*k2 + 2*k3 + k4)/6;

% Grand-Schmidt

order=randperm(3);

norma=(y(7 + 3*(order(1) - 1))^2 + y(8 + 3*(order(1) - 1))^2 + y(9 + 3*(order(1) - 1))^2)^0.5;
y(7 + 3*(order(1) - 1))=y(7 + 3*(order(1) - 1))/norma;
y(8 + 3*(order(1) - 1))=y(8 + 3*(order(1) - 1))/norma;
y(9 + 3*(order(1) - 1))=y(9 + 3*(order(1) - 1))/norma;

y(7 + 3*(order(2) - 1))=y(7 + 3*(order(2) - 1)) - (y(7 + 3*(order(2) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(2) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(2) - 1))*y(9 + 3*(order(1) - 1)))/norma;
y(8 + 3*(order(2) - 1))=y(8 + 3*(order(2) - 1)) - (y(7 + 3*(order(2) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(2) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(2) - 1))*y(9 + 3*(order(1) - 1)))/norma;
y(9 + 3*(order(2) - 1))=y(9 + 3*(order(2) - 1)) - (y(7 + 3*(order(2) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(2) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(2) - 1))*y(9 + 3*(order(1) - 1)))/norma;

norma=(y(7 + 3*(order(2) - 1))^2 + y(8 + 3*(order(2) - 1))^2 + y(9 + 3*(order(2) - 1))^2)^0.5;
y(7 + 3*(order(2) - 1))=y(7 + 3*(order(2) - 1))/norma;
y(8 + 3*(order(2) - 1))=y(8 + 3*(order(2) - 1))/norma;
y(9 + 3*(order(2) - 1))=y(9 + 3*(order(2) - 1))/norma;

y(7 + 3*(order(3) - 1))=y(7 + 3*(order(3) - 1)) - (y(7 + 3*(order(3) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(3) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(3) - 1))*y(9 + 3*(order(1) - 1)))/norma;
y(8 + 3*(order(3) - 1))=y(8 + 3*(order(3) - 1)) - (y(7 + 3*(order(3) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(3) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(3) - 1))*y(9 + 3*(order(1) - 1)))/norma;
y(9 + 3*(order(3) - 1))=y(9 + 3*(order(3) - 1)) - (y(7 + 3*(order(3) - 1))*y(7 + 3*(order(1) - 1))+y(8 + 3*(order(3) - 1))*y(8 + 3*(order(1) - 1))+y(9 +
3*(order(3) - 1))*y(9 + 3*(order(1) - 1)))/norma;

norma=(y(7 + 3*(order(3) - 1))^2 + y(8 + 3*(order(3) - 1))^2 + y(9 + 3*(order(3) - 1))^2)^0.5;
y(7 + 3*(order(3) - 1))=y(7 + 3*(order(3) - 1))/norma;
y(8 + 3*(order(3) - 1))=y(8 + 3*(order(3) - 1))/norma;
y(9 + 3*(order(3) - 1))=y(9 + 3*(order(3) - 1))/norma;

% Save yout

if abs(t - ts)<h/2

    if all(isfinite(y))==0
        Error='Divergência na simulação.'
        break
    end

    if intinf==0
        yout(s,:)=y;
        tout(s)=t;
        s=s+1;
    end
    ts=ts+dt;

    if mode==1 % Dynamic-Input

        [deltaht deltaw deltavt chi theta eta psidot u v w udot vdot wdot xdot ydot zdot]=entrada(t,y);

        % ----- %

        history.t(history.size + 1)=t;

        history.V(history.size + 1)=y(1);
        history.psi(history.size + 1)=y(2);
        history.phi(history.size + 1)=y(3);
        history.chi(history.size + 1)=chi;
        history.theta(history.size + 1)=theta;
        history.eta(history.size + 1)=eta;

        history.alphadot(history.size + 1)=y(4);
        history.betadot(history.size + 1)=y(5);
        history.gammadot(history.size + 1)=y(6);

        history.x(history.size + 1)=y(16);
        history.y(history.size + 1)=y(17);
        history.z(history.size + 1)=y(18);

        history.m(history.size + 1)=y(22);
        history.J1(history.size + 1)=y(23);
        history.J2(history.size + 1)=y(24);
        history.J3(history.size + 1)=y(25);

        history.u(history.size + 1)=u;
        history.v(history.size + 1)=v;
        history.w(history.size + 1)=w;

        history.p(history.size + 1)=Prop.p;
        history.deltaht(history.size + 1)=deltaht*180/pi;
        history.deltavt(history.size + 1)=deltavt*180/pi;
        history.deltaw(history.size + 1)=deltaw*180/pi;

        if stall==1
            history.CLw(history.size + 1)=fCLw(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
            history.CDw(history.size + 1)=fCDw(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
        else
            history.CLw(history.size + 1)=Aero.CLalphaw*(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
            history.CDw(history.size + 1)=history.CLw(history.size + 1)^2/Aero.Arwp/pi/Aero.ew + Aero.CD0w;
        end
        history.CMpw(history.size + 1)=Aero.CMp0w;
        history.CMrw(history.size + 1)=Aero.CMrdeltaw*deltaw + (Aero.CMrRyw*0.5*(sin(theta)*cos(eta)*y(4)-sin(theta)*sin(eta)*y(5)+cos(theta)*y(6))*Geom.bw
- Aero.CMrRw*(1/2)*(cos(eta)*cos(theta)*y(4)-sin(eta)*cos(theta)*y(5)-sin(theta)*y(6))*Geom.bw)/y(1);
        history.CMyw(history.size + 1)=Aero.CMydeltaw*deltaw + (Aero.CMyRyw*0.5*(sin(theta)*cos(eta)*y(4)-sin(theta)*sin(eta)*y(5)+cos(theta)*y(6))*Geom.bw
- Aero.CMyRw*(1/2)*(cos(eta)*cos(theta)*y(4)-sin(eta)*cos(theta)*y(5)-sin(theta)*y(6))*Geom.bw)/y(1);

        history.CLht(history.size + 1)=Aero.CLalphah*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180) + Aero.CLdeltah*t*deltah;
        history.CDht(history.size + 1)=Aero.CD0ht + (Aero.CLalphah*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180))^2/pi/Aero.Arht/Aero.eht +
2*Aero.CLalphah*Aero.CLdeltah*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180)/pi/Aero.Arht/Aero.eht*deltah;
        history.CMph(history.size + 1)=Aero.CMp0ht + Aero.CMpdelht*deltah;

        history.CLf(history.size + 1)=Aero.CL0f + Aero.CLalphaf*(theta - Iner.theta0*pi/180);
        history.CDf(history.size + 1)=Aero.CD0f + Aero.CDalphaf*(theta - Iner.theta0*pi/180);
        history.CMpf(history.size + 1)=Aero.CMp0f + Aero.CMpalphaf*(theta - Iner.theta0*pi/180);

        history.CLvt(history.size + 1)=Aero.CLalphavt*eta + Aero.CLdeltavt*deltavt;
        history.CDvt(history.size + 1)=Aero.CD0vt;

        history.size=history.size + 1;

        % ----- %

        axes(seeflight);
        line(history.t,history.deltaht,'Color','r');
        line(history.t,history.deltavt,'Color','g');
        line(history.t,history.deltaw,'Color','b');
        line(history.t,history.p*max(get(seeflight,'YLim')), 'Color','k');

```

```

if t>5 set(seeflight,'XLim',[t - 5) (t + 5)); end

F(1,1)=cos(eta)*cos(theta)*cos(y(3))*cos(y(2))-sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*sin(y(2))-(-
sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(y(3))*cos(y(2));
F(1,2)=-sin(eta)*cos(theta)*cos(y(3))*cos(y(2))-(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*sin(y(2))-(-cos(eta)*sin(chi)-
sin(eta)*sin(theta)*cos(chi))*sin(y(3))*cos(y(2));
F(1,3)=-sin(theta)*cos(y(3))*cos(y(2))-cos(theta)*sin(chi)*sin(y(2))-cos(theta)*cos(chi)*sin(y(3))*cos(y(2));
F(2,1)=cos(eta)*cos(theta)*cos(y(3))*sin(y(2))+(sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*cos(y(2))-(-
sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(y(3))*sin(y(2));
F(2,2)=-sin(eta)*cos(theta)*cos(y(3))*sin(y(2))+(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*cos(y(2))-(-cos(eta)*sin(chi)-
sin(eta)*sin(theta)*cos(chi))*sin(y(3))*sin(y(2));
F(2,3)=-sin(theta)*cos(y(3))*sin(y(2))+cos(theta)*sin(chi)*cos(y(2))-cos(theta)*cos(chi)*sin(y(3))*sin(y(2));
F(3,1)=cos(eta)*cos(theta)*sin(y(3))+(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*cos(y(3));
F(3,2)=-sin(eta)*cos(theta)*sin(y(3))+(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*cos(y(3));
F(3,3)=-sin(theta)*sin(y(3))+cos(theta)*cos(chi)*cos(y(3));

for j=1:size(solid.v_cg,1) V_cg(j,:)=solid.v_cg(j,:) + [y(16) y(17) y(18)]; end
for j=1:size(solid.v_w,1) V_w(j,:)=(F*solid.v_w(j,:))' + [y(16) y(17) y(18)]; end
for j=1:size(solid.v_ht,1) V_ht(j,:)=(F*solid.v_ht(j,:))' + [y(16) y(17) y(18)]; end
for j=1:size(solid.v_vt,1) V_vt(j,:)=(F*solid.v_vt(j,:))' + [y(16) y(17) y(18)]; end
for j=1:size(solid.v_t,1) V_t(j,:)=(F*solid.v_t(j,:))' + [y(16) y(17) y(18)]; end

set(s3d_cg,'Vertices',V_cg);
set(s3d_w,'Vertices',V_w);
set(s3d_ht,'Vertices',V_ht);
set(s3d_vt,'Vertices',V_vt);
set(s3d_t,'Vertices',V_t);

norma=(xdot^2 + ydot^2 + zdot^2)^0.5;
set(s3d_vel,'XData',[y(16) y(16) + xdot/norma],'YData',[y(17) y(17) + ydot/norma],'ZData',[y(18) y(18) + zdot/norma]);

set(wind3d,'XData',[X_wd + y(16)],'YData',[Y_wd + y(17)],'ZData',[Z_wd + y(18)],'UData',[U_wd*u],'VData',[V_wd*v],'WData',[W_wd*w]);

set(virtual,'CameraTarget',[y(16) y(17) y(18)],'CameraViewAngle',solid.ang);

if abs(y(16))>95*Geom.bw || abs(y(17))>95*Geom.bw || y(18)<5*Geom.bw

    delete(s3d_cg);
    delete(s3d_w);
    delete(s3d_ht);
    delete(s3d_vt);
    delete(s3d_t);
    delete(s3d_vel);
    delete(wind3d);

    dx=0.3;
    nx=round(fps*dx) + 1;
    tx=t:dx/(nx - 1):(t + dx);
    for k=1:nx

        for j=1:size(V_bum,1) V_bum(j,:)=(tx(k) - t)*Geom.bw*v_bum(j,:)/dx + [y(16) y(17) y(18)]; end
        set(buumm,'Vertices',V_bum,'Visible','on');
        drawnow;
        while (tx(k) - toc - ti)>0 end % Real-Time

    end
    break

end

drawnow;
if (t - toc - ti)<0 set(realtime,'Color',[1 0 0]);
else set(realtime,'Color',[0 0 0]);
end
while (t - toc - ti)>0 end % Real-Time
set(realtime,'String',sprintf('%s.ifs',round(t*10)/10));

end

end

t=t+h;

end
if intinf==0 tout(nout)=tf; end

catch
    Error='Figure closed during simulation.'
end
end
end

```

```

function [deltaht deltaw deltavt chi theta eta psidot u v w udot vdot wdot xdot
ydot zdot]=entrada(t,e)
% Calcula valores intermediarios
%
% e(1)=V;
% e(2)=psi;
% e(3)=phi;
% e(4)=alphadot;
% e(5)=betadot;
% e(6)=gammadot;
% e(7)=e(7);
% e(8)=a(8);
% e(9)=a3;
% e(10)=b1;
% e(11)=b2;
% e(12)=b3;
% e(13)=c1;
% e(14)=c2;
% e(15)=c3;
% e(16)=x;
% e(17)=y;
% e(18)=z;
% e(19)=alpha;
% e(20)=beta;
% e(21)=gamma;
% e(22)=m;
% e(23)=J1;
% e(24)=J2;
% e(25)=J3;

global dps Geom Aero Prop Iner Atmo RV1 RV2 entr trans mode joy chilast thetalast etalast
eps=10;

if mode==1

    [ax but pov]=read(joy);

    deltaht=RV1(4) + ax(2)*Aero.Dht;
    deltaw=RV1(5) + ax(1)*Aero.Dw;
    deltavt=RV1(6) + ax(3)*Aero.Dvt;

    if but(1)==1
        Prop.p=Prop.p + 1/dps/4;
        if Prop.p>1 Prop.p=1; end
    elseif but(3)==1
        Prop.p=Prop.p - 1/dps/4;
        if Prop.p<0 Prop.p=0; end
    end
else
    switch entr
    case 1

        deltaht=RV1(4);
        if t>=trans(1) deltaht=RV2(4); end

        deltaw=RV1(5);
        if t>=trans(1) deltaw=RV2(5); end

        deltavt=RV1(6);
        if t>=trans(1) deltavt=RV2(6); end

    case 2

        deltaht=RV1(4);
        if t>=trans(1) deltaht=deltaht + (RV2(4) - RV1(4))*(t - trans(1))/(trans(2) - trans(1)); end
        if t>=trans(2) deltaht=RV2(4); end

        deltaw=RV1(5);
        if t>=trans(1) deltaw=deltaht + (RV2(5) - RV1(5))*(t - trans(1))/(trans(2) - trans(1)); end
        if t>=trans(2) deltaw=RV2(5); end

        deltavt=RV1(6);
        if t>=trans(1) deltavt=deltaht + (RV2(6) - RV1(6))*(t - trans(1))/(trans(2) - trans(1)); end
        if t>=trans(2) deltavt=RV2(6); end

    case 3

        deltaht=RV1(4);
        if t>=trans(1) deltaht=deltaht + (RV2(4) - RV1(4))/(trans(2) - trans(1))^2*(t - trans(1))^2; end
        if t>=trans(2) deltaht=RV2(4); end

        deltaw=RV1(5);
        if t>=trans(1) deltaw=deltaht + (RV2(5) - RV1(5))/(trans(2) - trans(1))^2*(t - trans(1))^2; end
        if t>=trans(2) deltaw=RV2(5); end

        deltavt=RV1(6);
        if t>=trans(1) deltavt=deltaht + (RV2(6) - RV1(6))/(trans(2) - trans(1))^2*(t - trans(1))^2; end
        if t>=trans(2) deltavt=RV2(6); end

    case 4

        deltaht=RV1(4);
        if t>=trans(1) deltaht=deltaht + (RV2(4) - RV1(4))/(1 + exp(eps*((t - trans(1))/(trans(1) - trans(2)) + 1/2))); end
        if t>=trans(2) deltaht=RV2(4); end

        deltaw=RV1(5);
        if t>=trans(1) deltaw=deltaht + (RV2(5) - RV1(5))/(1 + exp(eps*((t - trans(1))/(trans(1) - trans(2)) + 1/2))); end
        if t>=trans(2) deltaw=RV2(5); end

        deltavt=RV1(6);
        if t>=trans(1) deltavt=deltaht + (RV2(6) - RV1(6))/(1 + exp(eps*((t - trans(1))/(trans(1) - trans(2)) + 1/2))); end
        if t>=trans(2) deltavt=RV2(6); end

    end

end

% ----- %

u=Atmo.u;
udot=0;
if t>=trans(3)
    u=Atmo.u + Atmo.du/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)));
    udot=-Atmo.du/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)))^2*exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2))*eps/(trans(3) - trans(4));
end
if t>=trans(4)
    u=Atmo.u + Atmo.du/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)));
    udot=Atmo.du/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)))^2*exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2))*eps/(trans(4) -
trans(5));
end
if t>=trans(5)
    u=Atmo.u;
    udot=0;
end

```

```

end

v=Atmo.v;
vdot=0;
if t>=trans(3)
    v=Atmo.v + Atmo.dv/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)));
    vdot=-Atmo.dv/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)))^2*exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2))*eps/(trans(3) - trans(4));
end
if t>=trans(4)
    v=Atmo.v + Atmo.dv/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)));
    vdot=Atmo.dv/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)))^2*exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2))*eps/(trans(4) - trans(5));
end
if t>=trans(5)
    v=Atmo.v;
    vdot=0;
end

w=Atmo.w;
wdot=0;
if t>=trans(3)
    w=Atmo.w + Atmo.dw/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)));
    wdot=-Atmo.dw/(1 + exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2)))^2*exp(eps*((t - trans(3))/(trans(3) - trans(4)) + 1/2))*eps/(trans(3) - trans(4));
end
if t>=trans(4)
    w=Atmo.w + Atmo.dw/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)));
    wdot=Atmo.dw/(1 + exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2)))^2*exp(-eps*((t - trans(4))/(trans(4) - trans(5)) + 1/2))*eps/(trans(4) - trans(5));
end
if t>=trans(5)
    w=Atmo.w;
    wdot=0;
end

% ----- %

if abs(cos(e(3))*cos(e(2))*e(13) + cos(e(3))*sin(e(2))*e(14) + sin(e(3))*e(15))>1
    theta=thetalast;
else
    theta=-asin(cos(e(3))*cos(e(2))*e(13) + cos(e(3))*sin(e(2))*e(14) + sin(e(3))*e(15));
end
thetalast=theta;

if (-sin(e(3))*cos(e(2))*e(13) - sin(e(3))*sin(e(2))*e(14) + cos(e(3))*e(15))==0
    if (-sin(e(2))*e(13) + cos(e(2))*e(14))<0 chi=-pi/2;
    elseif (-sin(e(2))*e(13) + cos(e(2))*e(14))==0 chi=chilast;
    else chi=pi/2;
end
elseif (-sin(e(3))*cos(e(2))*e(13) - sin(e(3))*sin(e(2))*e(14) + cos(e(3))*e(15))<0
    chi=pi - atan((-sin(e(2))*e(13) + cos(e(2))*e(14))/(-sin(e(3))*cos(e(2))*e(13) - sin(e(3))*sin(e(2))*e(14) + cos(e(3))*e(15)));
else
    chi=atan((-sin(e(2))*e(13) + cos(e(2))*e(14))/(-sin(e(3))*cos(e(2))*e(13) - sin(e(3))*sin(e(2))*e(14) + cos(e(3))*e(15)));
end
chilast=chi;

if (cos(e(3))*cos(e(2))*e(7) + cos(e(3))*sin(e(2))*e(8) + sin(e(3))*e(9))==0
    if (cos(e(3))*cos(e(2))*e(10) + cos(e(3))*sin(e(2))*e(11) + sin(e(3))*e(12))<0 eta=pi/2;
    elseif (cos(e(3))*cos(e(2))*e(10) + cos(e(3))*sin(e(2))*e(11) + sin(e(3))*e(12))==0 eta=etalast;
    else eta=-pi/2;
end
elseif (cos(e(3))*cos(e(2))*e(7) + cos(e(3))*sin(e(2))*e(8) + sin(e(3))*e(9))<0
    eta=pi - atan((cos(e(3))*cos(e(2))*e(10) + cos(e(3))*sin(e(2))*e(11) + sin(e(3))*e(12))/(cos(e(3))*cos(e(2))*e(7) + cos(e(3))*sin(e(2))*e(8) + sin(e(3))*e(9)));
else
    eta=atan((cos(e(3))*cos(e(2))*e(10) + cos(e(3))*sin(e(2))*e(11) + sin(e(3))*e(12))/(cos(e(3))*cos(e(2))*e(7) + cos(e(3))*sin(e(2))*e(8) + sin(e(3))*e(9)));
end
etalast=eta;

% ----- %

E=Prop.p*Atmo.rho/Prop.rho*Prop.a2*(e(1)*(cos(eta)*cos(theta)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - sin(theta)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180))^2 + Prop.p*Atmo.rho/Prop.rho*Prop.a1*e(1)*(cos(eta)*cos(theta)*cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180) - sin(theta)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)) + Prop.p*Atmo.rho/Prop.rho*Prop.a0;

CLw=Aero.CLalphaw*(theta - Iner.theta0*pi/180 - Aero.alphaL0w*pi/180);
CL0ht=Aero.CLalphaht*(theta - Iner.theta0*pi/180 - Aero.alphaL0ht*pi/180);
CLf=Aero.CL0f + Aero.CLalphaf*(theta - Iner.theta0*pi/180);

Lw=0.5*Atmo.rho*e(1)^2*Geom.Sw*CLw;
Lht=0.5*Atmo.rho*e(1)^2*Geom.Sht*(CL0ht + Aero.CLdeltaht*deltaht);
Lf=0.5*Atmo.rho*e(1)^2*Geom.Sf*CLf;

Fr=E*(cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*sin(eta)*cos(chi) + cos(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)*cos(eta)*sin(theta)*sin(chi) + cos(theta)*sin(chi)*sin(Geom.gammat*pi/180 - Iner.theta0*pi/180 + Geom.betat*pi/180)) + (Lw + Lht + Lf)*((cos(eta)*sin(chi) + sin(eta)*sin(theta)*cos(chi))/(sin(theta)^2 + cos(eta)^2*cos(theta)^0.5);

if cos(e(3))==0
    psidot=0;
else
    psidot=Fr/e(1)/cos(e(3))/e(22) + sin(e(2))*(Prop.p*Iner.dmdt/e(22)*u + udot - cos(e(2))*(Prop.p*Iner.dmdt/e(22)*v + vdot);
end

% ----- %

xdot=e(1)*cos(e(3))*cos(e(2)) + u;
ydot=e(1)*cos(e(3))*sin(e(2)) + v;
zdot=e(1)*sin(e(3)) + w;

end

```

function animate(obj,event)

```
global N T fps t e watcher Geom Aero Prop Iner Atmo RV1 RV2 anim solid

desenha(0,0,1);

if findobj('Name','Trajetória 3D')>0 delete(anim); end
anim=figure('Units','normalized','Renderer','OpenGL','Position',[0.01 0.05 0.885 0.839],'Resize','off','Name','Trajetória 3D','NumberTitle','off','KeyPressFcn',{@play});

draw='normal';
type='gouraud';

limits=max([max(e(:,16)) - min(e(:,16)); max(e(:,17)) - min(e(:,17)); max(e(:,18))]);
X1=(max(e(:,16)) + min(e(:,16)))/2 - limits;
X2=(max(e(:,16)) + min(e(:,16)))/2 + limits;
Y1=(max(e(:,17)) + min(e(:,17)))/2 - limits;
Y2=(max(e(:,17)) + min(e(:,17)))/2 + limits;
Z1=-10;
Z2=2*max(e(:,18));

traj=axes('Units','pixels','DrawMode','normal','XLim',[X1 X2],'YLim',[Y1 Y2],'ZLim',[Z1 Z2],'Projection','orthographic','CameraPosition',[X1 Y1 Z2],'Color',[0.6 0.8 1]);
axis equal
hold on
xlabel('x [m]');
ylabel('y [m]');
zlabel('z [m]');

light('Position',get(gca,'CameraPosition'),'Style','infinite');

cronom=annotation('textbox',[0.92 0.05 0.05 0.04],'Units','normalized','HorizontalAlignment','Right');

ang=1;
scale=Geom.bw/5;
ambient=0.5;
diffuse=0.7;
specular=0.9;

% ----- %

K=100;
[X_g Y_g Z_g]=peaks(K);
X_g=X1:(X2 - X1)/(K - 1):X2;
Y_g=Y1:(Y2 - Y1)/(K - 1):Y2;
surface(X_g,Y_g,Z_g,'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
colormap(summer);

% line([X1 X2],[Y1 Y1],[0 0],'Color',[0 0 0]);
% line([X1 X2],[Y2 Y2],[0 0],'Color',[0 0 0]);
% line([X1 X1],[Y1 Y2],[0 0],'Color',[0 0 0]);
% line([X2 X2],[Y1 Y2],[0 0],'Color',[0 0 0]);

% ----- %

J=15;
[X_wd Y_wd Z_wd]=sphere(J);
windpoints=surf2patch(X_wd, Y_wd, Z_wd);

X_wd=1.5*Geom.bw*windpoints.vertices(:,1);
Y_wd=1.5*Geom.bw*windpoints.vertices(:,2);
Z_wd=1.5*Geom.bw*windpoints.vertices(:,3);
U_wd=ones(size(windpoints.vertices,1),1);
V_wd=ones(size(windpoints.vertices,1),1);
W_wd=ones(size(windpoints.vertices,1),1);

wind3d=quiver3(X_wd,Y_wd,Z_wd,U_wd,V_wd,W_wd,0,'Color',[1 1 1]);

% ----- %

try

for i=1:N

%     versor_a=line([e(i,16) - scale*e(i,7), e(i,16) + scale*e(i,7)], [e(i,17) - scale*e(i,8), e(i,17) + scale*e(i,8)], [e(i,18) - scale*e(i,9), e(i,18) + scale*e(i,9)], 'Color',[0.3 0.3 0.3]);
%     versor_b=line([e(i,16) - scale*e(i,10), e(i,16) + scale*e(i,10)], [e(i,17) - scale*e(i,11), e(i,17) + scale*e(i,11)], [e(i,18) - scale*e(i,12), e(i,18) + scale*e(i,12)], 'Color',[1 0 0]);
%     versor_c=line([e(i,16) - scale*e(i,13), e(i,16) + scale*e(i,13)], [e(i,17) - scale*e(i,14), e(i,17) + scale*e(i,14)], [e(i,18) - scale*e(i,15), e(i,18) + scale*e(i,15)], 'Color',[0 0 1]);

norma=(watcher(i,14)^2 + watcher(i,15)^2 + watcher(i,16)^2)^0.5;
versor_q=line([e(i,16), e(i,16) + watcher(i,14)/norma], [e(i,17), e(i,17) + watcher(i,15)/norma], [e(i,18), e(i,18) + watcher(i,16)/norma], 'Color',[0.7 0.7 0.7]);

psi=e(i,2);
phi=e(i,3);
chi=watcher(i,4);
theta=watcher(i,5);
eta=watcher(i,6);

G(1,1)=cos(eta)*cos(theta)*cos(phi)*cos(psi)-sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*sin(psi)-(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
G(1,2)=-sin(eta)*cos(theta)*cos(phi)*cos(psi)-cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*sin(psi)-(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
G(1,3)=-sin(eta)*cos(phi)*cos(psi)-cos(theta)*sin(chi)*sin(psi)+cos(theta)*cos(chi)*sin(phi)*cos(psi);
G(2,1)=cos(eta)*cos(theta)*cos(phi)*sin(psi)+sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*cos(psi)-(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
G(2,2)=-sin(eta)*cos(theta)*cos(phi)*sin(psi)+cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*cos(psi)-(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
G(2,3)=-sin(eta)*cos(phi)*sin(psi)+cos(theta)*sin(chi)*cos(psi)-cos(theta)*cos(chi)*sin(phi)*sin(psi);
G(3,1)=cos(eta)*cos(theta)*sin(phi)+(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*cos(phi);
G(3,2)=-sin(eta)*cos(theta)*sin(phi)+(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*cos(phi);
G(3,3)=-sin(eta)*sin(phi)+cos(theta)*cos(chi)*cos(phi);

for j=1:size(solid.v_cg,1) V_cg(j,:)=solid.v_cg(j,:) + [e(i,16) e(i,17) e(i,18)]; end
for j=1:size(solid.v_w,1) V_w(j,:)=(G*solid.v_w(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
for j=1:size(solid.v_ht,1) V_ht(j,:)=(G*solid.v_ht(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
for j=1:size(solid.v_vt,1) V_vt(j,:)=(G*solid.v_vt(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
for j=1:size(solid.v_t,1) V_t(j,:)=(G*solid.v_t(j,:))' + [e(i,16) e(i,17) e(i,18)]; end

cg=patch('Faces',solid.f_cg,'Vertices',V_cg,'FaceColor',[0.6 0.6 0.6],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
w=patch('Faces',solid.f_w,'Vertices',V_w,'FaceColor',[1 0 0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
ht=patch('Faces',solid.f_ht,'Vertices',V_ht,'FaceColor',[1 0 0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
v=patch('Faces',solid.f_vt,'Vertices',V_vt,'FaceColor',[0 0 1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
tm=patch('Faces',solid.f_t,'Vertices',V_t,'FaceColor',[0 0 1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);

set(wind3d,'XData',(X_wd + e(i,16)), 'YData',(Y_wd + e(i,17)), 'ZData',(Z_wd + e(i,18)), 'UData',(U_wd*watcher(i,8)), 'VData',(V_wd*watcher(i,9)), 'WData',(W_wd*watcher(i,10)));

set(cronom,'String',sprintf('%1.1fs',round(t(i)*10)/10));
```

```

set(traj,'CameraTarget',[e(i,16) e(i,17) e(i,18)],'CameraViewAngle',ang);

M(i)=getframe;

% delete(versor_a);
% delete(versor_b);
% delete(versor_c);

delete(versor_q);

delete(cg);
delete(w);
delete(ht);
delete(vt);
delete(tm);

end
pos=get(gca,'Position');

for i=2:(N - 1)

    versor_a=line([e(i,16) - scale*e(i,7), e(i,16) + scale*e(i,7)], [e(i,17) - scale*e(i,8), e(i,17) + scale*e(i,8)], [e(i,18) - scale*e(i,9), e(i,18) + scale*e(i,9)], 'Color',[0.3 0.3 0.3]);
    versor_b=line([e(i,16) - scale*e(i,10), e(i,16) + scale*e(i,10)], [e(i,17) - scale*e(i,11), e(i,17) + scale*e(i,11)], [e(i,18) - scale*e(i,12), e(i,18) + scale*e(i,12)], 'Color',[1 0 0]);
    versor_c=line([e(i,16) - scale*e(i,13), e(i,16) + scale*e(i,13)], [e(i,17) - scale*e(i,14), e(i,17) + scale*e(i,14)], [e(i,18) - scale*e(i,15), e(i,18) + scale*e(i,15)], 'Color',[0 0 1]);
    versor_q=line([e(i,16), e(i,16) + watcher(i,14)], [e(i,17), e(i,17) + watcher(i,15)], [e(i,18), e(i,18) + watcher(i,16)], 'Color',[0.7 0.7 0.7]);

end

for i=[1,N]

    psi=e(i,2);
    phi=e(i,3);
    chi=watcher(i,4);
    theta=watcher(i,5);
    eta=watcher(i,6);

    G(1,1)=cos(eta)*cos(theta)*cos(phi)*cos(psi)-(sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*sin(psi)-(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
    G(1,2)=-sin(eta)*cos(theta)*cos(phi)*cos(psi)-(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*sin(psi)-(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
    G(1,3)=-sin(eta)*cos(theta)*cos(phi)*cos(psi)-cos(theta)*sin(chi)*sin(psi)-cos(theta)*cos(chi)*sin(phi)*cos(psi);
    G(2,1)=cos(eta)*cos(theta)*cos(phi)*sin(psi)+(sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*cos(psi)-(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
    G(2,2)=-sin(eta)*cos(theta)*cos(phi)*sin(psi)+(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*cos(psi)-(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
    G(2,3)=-sin(eta)*cos(theta)*cos(phi)*sin(psi)+cos(theta)*sin(chi)*sin(psi)-cos(theta)*cos(chi)*sin(phi)*sin(psi);
    G(3,1)=cos(eta)*cos(theta)*sin(phi)*(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*cos(psi);
    G(3,2)=-sin(eta)*cos(theta)*sin(phi)*(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*cos(psi);
    G(3,3)=-sin(eta)*sin(phi)*cos(theta)*cos(chi)*cos(psi);

    for j=1:size(solid.v_cg,1) V_cg(j,:)=solid.v_cg(j,:) + [e(i,16) e(i,17) e(i,18)]; end
    for j=1:size(solid.v_w,1) V_w(j,:)=(G*solid.v_w(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
    for j=1:size(solid.v_ht,1) V_ht(j,:)=(G*solid.v_ht(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
    for j=1:size(solid.v_vt,1) V_vt(j,:)=(G*solid.v_vt(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
    for j=1:size(solid.v_t,1) V_t(j,:)=(G*solid.v_t(j,:))' + [e(i,16) e(i,17) e(i,18)]; end

    cg=patch('Faces',solid.f_cg,'Vertices',V_cg,'FaceColor',[0.6 0.6 0.6], 'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);
    w=patch('Faces',solid.f_w,'Vertices',V_w,'FaceColor',[1 0 0], 'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);
    ht=patch('Faces',solid.f_ht,'Vertices',V_ht,'FaceColor',[1 0 0], 'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);
    vt=patch('Faces',solid.f_vt,'Vertices',V_vt,'FaceColor',[0 0 1], 'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);
    tm=patch('Faces',solid.f_t,'Vertices',V_t,'FaceColor',[0 0 0], 'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);

end

catch
    Error='Figure closed during animation.'
end

% movie2avi(M,'movie','fps',fps,'quality',10,'compression','Indeo5');

% ----- %

function play(obj,event)

    if event.Character=='p'

        delete(traj);
        traj=axes('Units','pixels','Position',[0 0 1 1], 'DrawMode','normal', 'XLim',[X1 X2], 'YLim',[Y1 Y2], 'ZLim',[Z1 Z2], 'Projection','orthographic', 'CameraPosition',[X1 Y1 Z2], 'Color',[0.6 0.8 1]);
        axis equal
        hold on
        xlabel('x [m]');
        ylabel('y [m]');
        zlabel('z [m]');

        light('Position',get(gca,'CameraPosition'),'Style','infinite');

        movie(M,1,fps,[0 0 0]);

        set(traj,'CameraTarget',[e(N,16) e(N,17) e(N,18)], 'CameraViewAngle',ang, 'Position',pos);

        surface(X_g,Y_g,Z_g,'EdgeColor','none', 'FaceLighting','gouraud', 'AmbientStrength',ambient, 'DiffuseStrength',diffuse, 'SpecularStrength',specular);
        colormap(summer);

        line([X1 X2],[Y1 Y1],[0 0], 'Color',[0 0 0]);
        line([X1 X2],[Y2 Y2],[0 0], 'Color',[0 0 0]);
        line([X1 X1],[Y1 Y2],[0 0], 'Color',[0 0 0]);
        line([X2 X2],[Y1 Y2],[0 0], 'Color',[0 0 0]);

        for i=2:(N - 1)

            versor_a=line([e(i,16) - scale*e(i,7), e(i,16) + scale*e(i,7)], [e(i,17) - scale*e(i,8), e(i,17) + scale*e(i,8)], [e(i,18) - scale*e(i,9), e(i,18) + scale*e(i,9)], 'Color',[0.3 0.3 0.3]);
            versor_b=line([e(i,16) - scale*e(i,10), e(i,16) + scale*e(i,10)], [e(i,17) - scale*e(i,11), e(i,17) + scale*e(i,11)], [e(i,18) - scale*e(i,12), e(i,18) + scale*e(i,12)], 'Color',[1 0 0]);
            versor_c=line([e(i,16) - scale*e(i,13), e(i,16) + scale*e(i,13)], [e(i,17) - scale*e(i,14), e(i,17) + scale*e(i,14)], [e(i,18) - scale*e(i,15), e(i,18) + scale*e(i,15)], 'Color',[0 0 1]);
            versor_q=line([e(i,16), e(i,16) + watcher(i,14)], [e(i,17), e(i,17) + watcher(i,15)], [e(i,18), e(i,18) + watcher(i,16)], 'Color',[0.7 0.7 0.7]);

        end

        for i=[1,N]

            psi=e(i,2);
            phi=e(i,3);
            chi=watcher(i,4);
            theta=watcher(i,5);
            eta=watcher(i,6);

```

```

        G(1,1)=cos(eta)*cos(theta)*cos(phi)*cos(psi)-(sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*sin(psi)-(-
sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
        G(1,2)=-sin(eta)*cos(theta)*cos(phi)*cos(psi)-(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*sin(psi)-(-cos(eta)*sin(chi)-
sin(eta)*sin(theta)*cos(chi))*sin(phi)*cos(psi);
        G(1,3)=-sin(theta)*cos(phi)*cos(psi)-cos(theta)*sin(chi)*sin(psi)-cos(theta)*cos(chi)*sin(phi)*cos(psi);
        G(2,1)=cos(eta)*cos(theta)*cos(phi)*sin(psi)+(sin(eta)*cos(chi)+cos(eta)*sin(theta)*sin(chi))*cos(psi)-(-
sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
        G(2,2)=-sin(eta)*cos(theta)*cos(phi)*sin(psi)+(cos(eta)*cos(chi)-sin(eta)*sin(theta)*sin(chi))*cos(psi)-(-cos(eta)*sin(chi)-
sin(eta)*sin(theta)*cos(chi))*sin(phi)*sin(psi);
        G(2,3)=-sin(theta)*cos(phi)*sin(psi)+cos(theta)*sin(chi)*cos(psi)-cos(theta)*cos(chi)*sin(phi)*sin(psi);
        G(3,1)=cos(eta)*cos(theta)*sin(phi)+(-sin(eta)*sin(chi)+cos(eta)*sin(theta)*cos(chi))*cos(phi);
        G(3,2)=-sin(eta)*cos(theta)*sin(phi)+(-cos(eta)*sin(chi)-sin(eta)*sin(theta)*cos(chi))*cos(phi);
        G(3,3)=-sin(theta)*sin(phi)+cos(theta)*cos(chi)*cos(phi);

        for j=1:size(solid.v_cg,1) V_cg(j,:)=(G*solid.v_cg(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
        for j=1:size(solid.v_w,1) V_w(j,:)=(G*solid.v_w(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
        for j=1:size(solid.v_ht,1) V_ht(j,:)=(G*solid.v_ht(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
        for j=1:size(solid.v_vt,1) V_vt(j,:)=(G*solid.v_vt(j,:))' + [e(i,16) e(i,17) e(i,18)]; end
        for j=1:size(solid.v_t,1) V_t(j,:)=(G*solid.v_t(j,:))' + [e(i,16) e(i,17) e(i,18)]; end

        cg=patch('Faces',solid.f_cg,'Vertices',V_cg,'FaceColor',[0.6 0.6
0.6],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
        w=patch('Faces',solid.f_w,'Vertices',V_w,'FaceColor',[1 0
0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
        ht=patch('Faces',solid.f_ht,'Vertices',V_ht,'FaceColor',[1 0
0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
        vt=patch('Faces',solid.f_vt,'Vertices',V_vt,'FaceColor',[0 0
1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
        tm=patch('Faces',solid.f_t,'Vertices',V_t,'FaceColor',[0 0
1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);

        end

        end

end

% ----- %

end

```

function desenha(obj,event,background)

```
% Esboca o Aviao

global Geom Aero Prop Iner Atmo RV1 mainframe desen viewzoom view3d solid b_simul

try

n=10;

ambient=0.3;
diffuse=0.6;
specular=0.9;

x1=-Geom.bw/25;
x2=Geom.bw/25;
y1=-Geom.bw/25;
y2=Geom.bw/25;
z1=-Geom.bw/25;
z2=Geom.bw/25;
[X,Y,Z]=meshgrid(x1:(x2 - x1)/(n - 1):x2,y1:(y2 - y1)/(n - 1):y2,z1:(z2 - z1)/(n - 1):z2);
for i=1:n
    for j=1:n
        for k=1:n
            V(i,j,k)=(X(i,j,k)^2/x1^2 + Y(i,j,k)^2/y1^2 + Z(i,j,k)^2/z1^2)^0.5;
        end
    end
end
[solid.f_cg,solid.v_cg]=isosurface(X,Y,Z,V,1);

x1=-2*Geom.Sw/pi/Geom.bw;
x2=2*Geom.Sw/pi/Geom.bw;
y1=-Geom.bw/2;
y2=Geom.bw/2;
z1=-Geom.bw/50;
z2=Geom.bw/50;
[X,Y,Z]=meshgrid(x1:(x2 - x1)/(n - 1):x2,y1:(y2 - y1)/(n - 1):y2,z1:(z2 - z1)/(n - 1):z2);
for i=1:n
    for j=1:n
        for k=1:n
            V(i,j,k)=(X(i,j,k)^2/x1^2 + Y(i,j,k)^2/y1^2 + Z(i,j,k)^2/z1^2)^0.5;
        end
    end
end
[solid.f_w,solid.v_w]=isosurface(X,Y,Z,V,1);
for i=1:size(solid.v_w,1) solid.v_w(i,1)=solid.v_w(i,1) + (1 - solid.v_w(i,2)^2/y1^2)^0.5*x1/2; end

x1=-2*Geom.Sht/pi/Geom.bht;
x2=2*Geom.Sht/pi/Geom.bht;
y1=-Geom.bht/2;
y2=Geom.bht/2;
z1=-Geom.bht/50;
z2=Geom.bht/50;
[X,Y,Z]=meshgrid(x1:(x2 - x1)/(n - 1):x2,y1:(y2 - y1)/(n - 1):y2,z1:(z2 - z1)/(n - 1):z2);
for i=1:n
    for j=1:n
        for k=1:n
            V(i,j,k)=(X(i,j,k)^2/x1^2 + Y(i,j,k)^2/y1^2 + Z(i,j,k)^2/z1^2)^0.5;
        end
    end
end
[solid.f_ht,solid.v_ht]=isosurface(X,Y,Z,V,1);
for i=1:size(solid.v_ht,1) solid.v_ht(i,1)=solid.v_ht(i,1) + (1 - solid.v_ht(i,2)^2/y1^2)^0.5*x1/2; end

x1=-2*Geom.Svt/pi/Geom.bvt;
x2=2*Geom.Svt/pi/Geom.bvt;
y1=-Geom.bvt/25;
y2=Geom.bvt/25;
z1=-Geom.bvt/2;
z2=Geom.bvt/2;
[X,Y,Z]=meshgrid(x1:(x2 - x1)/(n - 1):x2,y1:(y2 - y1)/(n - 1):y2,z1:(z2 - z1)/(n - 1):z2);
for i=1:n
    for j=1:n
        for k=1:n
            V(i,j,k)=(X(i,j,k)^2/x1^2 + Y(i,j,k)^2/y1^2 + Z(i,j,k)^2/z1^2)^0.5;
        end
    end
end
[solid.f_vt,solid.v_vt]=isosurface(X,Y,Z,V,1);
for i=1:size(solid.v_vt,1) solid.v_vt(i,1)=solid.v_vt(i,1) + (1 - solid.v_vt(i,3)^2/z1^2)^0.5*x1/2; end

x1=-Geom.bw/100;
x2=Geom.bw/100;
y1=-Geom.bw/10;
y2=Geom.bw/10;
z1=-Geom.bw/10;
z2=Geom.bw/10;
[X,Y,Z]=meshgrid(x1:(x2 - x1)/(n - 1):x2,y1:(y2 - y1)/(n - 1):y2,z1:(z2 - z1)/(n - 1):z2);
for i=1:n
    for j=1:n
        for k=1:n
            V(i,j,k)=(Y(i,j,k)^2/y1^2 + Z(i,j,k)^2/z1^2)^0.5;
        end
    end
end
[solid.f_t,solid.v_t]=isosurface(X,Y,Z,V,1);

theta=-Aero.alpha0w*pi/180;
for i=1:size(solid.v_w,1)
    solid.v_w(i,1)=solid.v_w(i,1)*cos(theta) - solid.v_w(i,3)*sin(theta);
    solid.v_w(i,3)=solid.v_w(i,1)*sin(theta) + solid.v_w(i,3)*cos(theta);
end

l=Geom.lw;
gamma=Geom.gammaw*pi/180;
for i=1:size(solid.v_w,1)
    solid.v_w(i,1)=solid.v_w(i,1) + l*cos(gamma);
    solid.v_w(i,3)=solid.v_w(i,3) + l*sin(gamma);
end

theta=-Aero.alpha0ht*pi/180;
for i=1:size(solid.v_ht,1)
    solid.v_ht(i,1)=solid.v_ht(i,1)*cos(theta) - solid.v_ht(i,3)*sin(theta);
    solid.v_ht(i,3)=solid.v_ht(i,1)*sin(theta) + solid.v_ht(i,3)*cos(theta);
end

l=Geom.lht;
gamma=Geom.gammaht*pi/180;
for i=1:size(solid.v_ht,1)
    solid.v_ht(i,1)=solid.v_ht(i,1) + l*cos(gamma);
    solid.v_ht(i,3)=solid.v_ht(i,3) + l*sin(gamma);
end

l=Geom.lvt;
gamma=Geom.gammavt*pi/180;
for i=1:size(solid.v_vt,1)
    solid.v_vt(i,1)=solid.v_vt(i,1) + l*cos(gamma);
    solid.v_vt(i,3)=solid.v_vt(i,3) + l*sin(gamma);
end
```

```

end

theta=Geom.gammat*pi/180 + Geom.betat*pi/180;
for i=1:size(solid.v_t,1)
    solid.v_t(i,1)=solid.v_t(i,1)*cos(theta) - solid.v_t(i,3)*sin(theta);
    solid.v_t(i,3)=solid.v_t(i,1)*sin(theta) + solid.v_t(i,3)*cos(theta);
end

l=Geom.lt;
gamma=Geom.gammat*pi/180;
for i=1:size(solid.v_t,1)
    solid.v_t(i,1)=solid.v_t(i,1) + l*cos(gamma);
    solid.v_t(i,3)=solid.v_t(i,3) + l*sin(gamma);
end

theta=-Iner.theta0*pi/180;
for i=1:size(solid.v_w,1)
    solid.v_w(i,1)=solid.v_w(i,1)*cos(theta) - solid.v_w(i,3)*sin(theta);
    solid.v_w(i,3)=solid.v_w(i,1)*sin(theta) + solid.v_w(i,3)*cos(theta);
end
for i=1:size(solid.v_ht,1)
    solid.v_ht(i,1)=solid.v_ht(i,1)*cos(theta) - solid.v_ht(i,3)*sin(theta);
    solid.v_ht(i,3)=solid.v_ht(i,1)*sin(theta) + solid.v_ht(i,3)*cos(theta);
end
for i=1:size(solid.v_vt,1)
    solid.v_vt(i,1)=solid.v_vt(i,1)*cos(theta) - solid.v_vt(i,3)*sin(theta);
    solid.v_vt(i,3)=solid.v_vt(i,1)*sin(theta) + solid.v_vt(i,3)*cos(theta);
end
for i=1:size(solid.v_t,1)
    solid.v_t(i,1)=solid.v_t(i,1)*cos(theta) - solid.v_t(i,3)*sin(theta);
    solid.v_t(i,3)=solid.v_t(i,1)*sin(theta) + solid.v_t(i,3)*cos(theta);
end

% ----- %

if background==0

    set(viewzoom,'Enable','on','Value',0);
    set(view3d,'Enable','on','Value',0);
    if RV1(9)==1 set(b_simul,'Enable','on'); end

    figure(mainframe);
    delete(desen);
    desen=axes('Units','normalized','Position',[0.05 0.15 0.65 0.75],'XLim',[-Geom.bw/3 Geom.bw/3],'YLim',[-Geom.bw/3 Geom.bw/3],'ZLim',[-Geom.bw/6
Geom.bw/6],'CameraPosition',[Geom.bw/3 Geom.bw/3
Geom.bw/6],'Color','none','XTickLabel',[],'YTickLabel',[],'ZTickLabel',[],'XTick',[],'YTick',[],'ZTick',[],'XColor',[0.8 0.8 0.8],'YColor',[0.8 0.8
0.8],'ZColor',[0.8 0.8 0.8]);
    axis equal
    hold on
    zoom off
    rotate3d off

    patch('Faces',solid.f_cg,'Vertices',solid.v_cg,'FaceColor',[0.6 0.6
0.6],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    patch('Faces',solid.f_w,'Vertices',solid.v_w,'FaceColor',[1 0
0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    patch('Faces',solid.f_ht,'Vertices',solid.v_ht,'FaceColor',[1 0
0],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    patch('Faces',solid.f_vt,'Vertices',solid.v_vt,'FaceColor',[0 0
1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    patch('Faces',solid.f_t,'Vertices',solid.v_t,'FaceColor',[0 0
1],'EdgeColor','none','FaceLighting','gouraud','AmbientStrength',ambient,'DiffuseStrength',diffuse,'SpecularStrength',specular);
    light('Position',get(gca,'CameraPosition'),'Style','infinite');

end
catch
    Error='Unable to draw.'
end

% ----- %

end

```

[illegible]

[illegible]


```

theta0 = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.30 0.25
0.04],'String',num2str(Inner.theta0),'FontSize',11,'ForegroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@savepropiner});
rho0 = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.26 0.25 0.04],'String',num2str(Prop.rho0),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@savepropiner});

uicontrol('Style','Text','Units','Normalized','Position',[0.1 0.29 0.25 0.04],'String','theta0','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.1 0.25 0.25 0.04],'String','rho0','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

uicontrol('Style','Text','Units','Normalized','Position',[0.673 0.295 0.3 0.04],'String','0','FontSize',8,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.673 0.25 0.3 0.04],'String','kg/m^3','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

%-----%

case {'setatmo'}

setatmo=figure('Units','normalized','Position',[0.655 0.05 0.16 0.90],'Menubar','none','Resize','off','Name','Atmosfera','NumberTitle','off');

%-----%

uicontrol('Style','Text','Units','Normalized','Position',[0.25 0.93 0.50 0.04],'String','Vento','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');

u = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.9 0.25 0.04],'String',num2str(Atmo.u),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});
v = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.86 0.25 0.04],'String',num2str(Atmo.v),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});
w = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.82 0.25 0.04],'String',num2str(Atmo.w),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});

uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.89 0.1 0.04],'String','u','FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8
0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.85 0.1 0.04],'String','v','FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8
0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.81 0.1 0.04],'String','w','FontSize',11,'ForegroundColor',[0 0 0],'BackgroundColor',[0.8
0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.89 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.85 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.81 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

%-----%

uicontrol('Style','Text','Units','Normalized','Position',[0.25 0.77 0.50 0.04],'String','Rajada','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');

du = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.74 0.25 0.04],'String',num2str(Atmo.du),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});
dv = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.70 0.25 0.04],'String',num2str(Atmo.dv),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});
dw = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.66 0.25 0.04],'String',num2str(Atmo.dw),'FontSize',11,'ForegroundColor',[1 1
1],'BackgroundColor',[0.35 0.35 0.35],'Fontname','Corbel','Callback',{@saveatmo,0,0});

uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.73 0.1 0.04],'String','du','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.69 0.1 0.04],'String','dv','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.65 0.1 0.04],'String','dw','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.73 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.69 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.65 0.15 0.04],'String','m/s','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

%-----%

uicontrol('Style','Text','Units','Normalized','Position',[0.15 0.61 0.7 0.04],'String','Altitude/Densidade','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Center','FontWeight','Bold');

rho = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.58 0.25
0.04],'String',num2str(round(Atmo.rho*1000)/1000),'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','Callback',{@saveatmo,1,1});
alt = uicontrol('Style','Edit','Units','Normalized','Position',[0.375 0.54 0.25
0.04],'String',num2str(round(Atmo.alt)),'FontSize',11,'ForegroundColor',[1 1 1],'BackgroundColor',[0.35 0.35
0.35],'Fontname','Corbel','Callback',{@saveatmo,1,2});

uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.57 0.1 0.04],'String','rho','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.2 0.53 0.1 0.04],'String','Alt','FontSize',11,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.57 0.2 0.04],'String','kg/m^3','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');
uicontrol('Style','Text','Units','Normalized','Position',[0.67 0.53 0.2 0.04],'String','m','FontSize',10,'ForegroundColor',[0 0
0],'BackgroundColor',[0.8 0.8 0.8],'Fontname','Corbel','HorizontalAlignment','Left');

%-----%

end

%-----%

function savegeom(obj,event)

Geom.bw=str2double(get(bw,'String'));
Geom.Sw=str2double(get(Sw,'String'));
Geom.lw=str2double(get(lw,'String'));
Geom.gammaw=str2double(get(gammaw,'String'));

Geom.bht=str2double(get(bht,'String'));
Geom.Sht=str2double(get(Sht,'String'));
Geom.lht=str2double(get(lht,'String'));
Geom.gammaht=str2double(get(gammaht,'String'));

Geom.bvt=str2double(get(bvt,'String'));
Geom.Svt=str2double(get(Svt,'String'));
Geom.lvt=str2double(get(lvt,'String'));
Geom.gammavt=str2double(get(gammavt,'String'));

Geom.Sf=str2double(get(Sf,'String'));
Geom.lf=str2double(get(lf,'String'));
Geom.gammaf=str2double(get(gammaf,'String'));

Geom.lt=str2double(get(lt,'String'));
Geom.gammat=str2double(get(gammat,'String'));
Geom.betat=str2double(get(betat,'String'));

try
Aero.Arw=Geom.bw^2/Geom.Sw;

```

```

Aero.Arht=Geom.bht^2/Geom.Sht;
Aero.Arvt=Geom.bvt^2/Geom.Svt;

set(b_simul,'Enable','off');
if findobj('Name','Estabilidade Nao-linear','Tag','1')>0
    if RV1(9)==1 && RV2(9)==1 estant(0,0,1,1);
    else estant(0,0,0,1);
    end
elseif findobj('Name','Regimes de Vôo','Tag','1')>0 regvoo(0,0,1,0,0);
else
    RV1(9)=0;
    RV2(9)=0;
    desenha(0,0,0);
end
catch
    Error='Bad Input.'
end

end

%-----%

function saveaero(obj,event)

    Aero.CLalphaw=str2double(get(CLalphaw,'String'));
    Aero.alphaL0w=str2double(get(alphaL0w,'String'));
    Aero.CD0w=str2double(get(CD0w,'String'));
    Aero.Arww=str2double(get(Arww,'String'));
    Aero.eww=str2double(get(eww,'String'));
    Aero.cww=str2double(get(cww,'String'));
    Aero.CMp0w=str2double(get(CMp0w,'String'));
    Aero.CMydeltaw=str2double(get(CMydeltaw,'String'));
    Aero.CMrdeltaw=str2double(get(CMrdeltaw,'String'));
    Aero.CMrRyw=str2double(get(CMrRyw,'String'));
    Aero.CMrRrw=str2double(get(CMrRrw,'String'));
    Aero.CMyRyw=str2double(get(CMyRyw,'String'));
    Aero.CMyRrw=str2double(get(CMyRrw,'String'));

    Aero.CLalphaht=str2double(get(CLalphaht,'String'));
    Aero.alphaL0ht=str2double(get(alphaL0ht,'String'));
    Aero.CD0ht=str2double(get(CD0ht,'String'));
    Aero.CLdeltaht=str2double(get(CLdeltaht,'String'));
    Aero.Arht=str2double(get(Arht,'String'));
    Aero.eht=str2double(get(eht,'String'));
    Aero.cht=str2double(get(cht,'String'));
    Aero.CMp0ht=str2double(get(CMp0ht,'String'));
    Aero.CMpdeltaht=str2double(get(CMpdeltaht,'String'));

    Aero.CL0f=str2double(get(CL0f,'String'));
    Aero.CLalphaf=str2double(get(CLalphaf,'String'));
    Aero.CD0f=str2double(get(CD0f,'String'));
    Aero.CDalphaf=str2double(get(CDalphaf,'String'));
    Aero.cf=str2double(get(cf,'String'));
    Aero.CMp0f=str2double(get(CMp0f,'String'));
    Aero.CMpalphaf=str2double(get(CMpalphaf,'String'));

    Aero.Arvt=str2double(get(Arvt,'String'));
    Aero.CLalphavt=str2double(get(CLalphavt,'String'));
    Aero.CLdeltavt=str2double(get(CLdeltavt,'String'));
    Aero.CD0vt=str2double(get(CD0vt,'String'));

    try
        Geom.bw=(Aero.Arww*Geom.Sw)^0.5;
        Geom.bht=(Aero.Arht*Geom.Sht)^0.5;
        Geom.bvt=(Aero.Arvt*Geom.Svt)^0.5;

    set(b_simul,'Enable','off');
    if findobj('Name','Estabilidade Nao-linear','Tag','1')>0
        if RV1(9)==1 && RV2(9)==1 estant(0,0,1,1);
        else estant(0,0,0,1);
        end
    elseif findobj('Name','Regimes de Vôo','Tag','1')>0 regvoo(0,0,1,0,0);
    else
        RV1(9)=0;
        RV2(9)=0;
        desenha(0,0,0);
    end
    catch
        Error='Bad Input.'
    end

end

%-----%

function savepropiner(obj,event)

    Prop.a0=str2double(get(a0,'String'));
    Prop.a1=str2double(get(a1,'String'));
    Prop.a2=str2double(get(a2,'String'));

    Prop.p=str2double(get(p,'String'));

    Iner.m0=str2double(get(m0,'String'));
    Iner.J10=str2double(get(J10,'String'));
    Iner.J20=str2double(get(J20,'String'));
    Iner.J30=str2double(get(J30,'String'));

    Iner.dmdt=str2double(get(dmdt,'String'));
    Iner.dJ1dt=str2double(get(dJ1dt,'String'));
    Iner.dJ2dt=str2double(get(dJ2dt,'String'));
    Iner.dJ3dt=str2double(get(dJ3dt,'String'));

    Iner.theta0=str2double(get(theta0,'String'));
    Prop.rho0=str2double(get(rho0,'String'));

    try
        set(b_simul,'Enable','off');
    if findobj('Name','Estabilidade Nao-linear','Tag','1')>0
        if RV1(9)==1 && RV2(9)==1 estant(0,0,1,1);
        else estant(0,0,0,1);
        end
    elseif findobj('Name','Regimes de Vôo','Tag','1')>0 regvoo(0,0,1,0,0);
    else
        RV1(9)=0;
        RV2(9)=0;
        desenha(0,0,0);
    end
    catch
        Error='Bad Input.'
    end

end

%-----%

function saveatmo(obj,event,act,altden)

```

```

Atmo.u=str2double(get(u,'String'));
Atmo.v=str2double(get(v,'String'));
Atmo.w=str2double(get(w,'String'));

Atmo.du=str2double(get(du,'String'));
Atmo.dv=str2double(get(dv,'String'));
Atmo.dw=str2double(get(dw,'String'));

try
if altden==1
    Atmo.rho=str2double(get(rho,'String'));
    Atmo.alt=(1 - (Atmo.rho/1.225)^(1/4.256))/0.0000225577;
    set(alt,'String',num2str(round(Atmo.alt)));
end
if altden==2
    Atmo.alt=str2double(get(alt,'String'));
    Atmo.rho=1.225*(1 - 0.0000225577*Atmo.alt)^4.256;
    set(rho,'String',num2str(round(Atmo.rho*1000)/1000));
end

set(b_simul,'Enable','off');
if findobj('Name','Estabilidade Nao-linear','Tag','1')>0
    if RV1(9)==1 && RV2(9)==1 estant(0,0,act,act);
    else estant(0,0,0,act);
    end
elseif findobj('Name','Regimes de Vôo','Tag','1')>0 regvoo(0,0,act,0,0);
else
    RV1(9)=0;
    RV2(9)=0;
    desenha(0,0,0);
end
catch
    Error='Bad Input.'
end

end

%-----%
end

```

```
function cmdwindow(obj,event)
```

```
% Salva todas as constantes
```

```
global T g I J K fps dps FPS DPS t trt e ert watcher Geom Aero Prop Iner Atmo colors fCLw fCDw stall mainframe setgeom setaero setpropiner setatmo setregvoo  
setestant control desen anim simul RV1 RV2 solid auto_trim history help c_alphaw c_CLw c_CDw intcurves cmd
```

```
cmdline=get(cmd,'String');  
if strcmp(cmdline,'figure');  
    if findobj('Name','History')>0 close(intcurves); end  
    intcurves=figure('Name','History','NumberTitle','off');  
else  
    if findobj('Name','History')>0 figure(intcurves); end  
    try  
        eval(cmdline)  
    catch  
        Error=sprintf('Invalid Command: '%s'',cmdline)  
    end  
end  
set(cmd,'String','');  
end
```

```
function saveprt(obj,event)
```

```
% Salva todas as constantes
```

```
global T g FPS DPS Geom Aero Prop Iner Atmo colors RV1 RV2 entr trans auto_trim history mainframe
```

```
name=input('File name? ','s');  
if isempty(name) name='Constantes.mat'; end  
save(name,'T','g','FPS','DPS','Geom','Aero','Prop','Iner','Atmo','colors','RV1','RV2','entr','trans','auto_trim','history');  
figure(mainframe);  
end
```

```
function loadprt(obj,event)
```

```
% Salva todas as constantes
```

```
global T N FPS DPS Geom Aero Prop Iner Atmo colors mainframe setgeom setaero setpropiner setatmo setregvoo setestant control anim simul RV1 RV2 entr trans  
auto_trim history
```

```
name=input('File name? ','s');  
if isempty(name) name='Constantes.mat'; end  
load(name,'T','g','FPS','DPS','Geom','Aero','Prop','Iner','Atmo','colors','RV1','RV2','entr','trans','auto_trim','history');  
  
if findobj('Name','Geometria')>0 close(setgeom); end  
if findobj('Name','Aerodinamica')>0 close(setaero); end  
if findobj('Name','Propulsao e Inercia')>0 close(setpropiner); end  
if findobj('Name','Atmosfera')>0 close(setatmo); end  
if findobj('Name','Regimes de Voo')>0 close(setregvoo); end  
if findobj('Name','Estabilidade Nao-linear')>0 close(setestant); end  
if findobj('Name','Trajetória 3D')>0 close(anim); end  
if findobj('Name','Entradas')>0 close(control); end  
if findobj('Name','Simulador')>0 close(simul); end  
  
figure(mainframe);  
end
```

```
function [fCLw fCDw]=CLw_CDw
```

```
% Atencao: CLw(0)=0 CDw(0)=CD0w  
% As funcoes sempre serao chamadas na forma: f(alpha - alphaL0)  
% NACA4418: Pontos - 'c_alphaw' 'c_CLw' 'c_CDw'
```

```
global c_alphaw c_CLw c_CDw
```

```
c_alphaw=[-1.745329252  
-1.658062789  
-1.570796327  
-1.483529864  
-1.396263402  
-1.308996939  
-1.221730476  
-1.134464014  
-1.047197551  
-0.959931089  
-0.872664626  
-0.785398163  
-0.698131701  
-0.610865238  
-0.523598776  
-0.436332313  
-0.34906585  
-0.261799388  
-0.174532925  
-0.087266463  
0  
0.087266463  
0.174532925  
0.261799388  
0.34906585  
0.436332313  
0.523598776  
0.610865238  
0.698131701  
0.785398163  
0.872664626  
0.959931089  
1.047197551  
1.134464014  
1.221730476  
1.308996939  
1.396263402  
1.483529864  
1.570796327  
1.658062789  
1.745329252];
```

```

c_CLW=[0.6
0.3
0
-0.3
-0.6
-0.75
-0.9
-1
-1.1
-1.15
-1.15
-1.15
-1.1
-1.15
-1.35
-1.5
-1.35
-1.1
-0.75
-0.38
0
0.38
0.75
1.1
1.35
1.5
1.35
1.15
1.1
1.15
1.15
1.15
1.1
1
0.9
0.75
0.6
0.3
0
-0.3
-0.6];

c_CDw=[2.05
2.07
2.07
2.05
2
1.95
1.85
1.8
1.65
1.5
1.32
1.15
1
0.75
0.5
0.25
0.15
0.1
0.05
0.03
0.02
0.03
0.05
0.1
0.15
0.25
0.5
0.75
1
1.15
1.32
1.5
1.65
1.8
1.85
1.95
2
2.05
2.07
2.07
2.05];

coefCLW=fit(c_alphaw,c_CLW,'poly9');
fCLW=@(x) feval(coefCLW,x);

coefCDw=fit(c_alphaw,c_CDw,'poly8');
fCDw=@(x) feval(coefCDw,x);

% fCDw=@(x) feval(coefCLW,x)^2/Aero.Arwing/pi/Aero.ew + Aero.CD0w;

end

```

```

function visual(obj,event,which)
% Ativa zoom e rotate3d

global desen viewzoom view3d

axes(desen);

if which==1

    set(view3d,'Value',0);
    rotate3d off

    if get(viewzoom,'Value')==1
        zoom on
    else
        zoom off
    end

elseif which==2

    set(viewzoom,'Value',0);
    zoom off

    if get(view3d,'Value')==1
        rotate3d on
    else
        rotate3d off
    end

end

end

function quitprog(obj,event)
% Fecha todas as janelas

global mainframe setgeom setaero setpropiner setatmo setregvoo setestant anim simul

delete(findobj('Type','figure'));

end

function inicializa_constantes
% Constantes => Modelos

global T g FPS DPS Geom Aero Prop Iner Atmo colors RV1 RV2 entr trans auto_trim
% ----- %

T=10;
FPS=8;
DPS=24;           % Atencao: Número inteiro de vezes de FPS

g=9.8;

entr=1;

trans=[2 4 4 6 8];

auto_trim=1;      % Para localizacao do alphaL0ht

% ----- %

Geom.bw=2.3;
Geom.Sw=1;
Geom.lw=0.0316;
Geom.gammaw=18;

Geom.bht=0.8;
Geom.Sht=0.27;
Geom.lht=0.91;
Geom.gammaht=168;

Geom.bvt=0.3;
Geom.Svt=0.08;
Geom.lvt=1;
Geom.gammavt=165;

Geom.Sf=0;
Geom.lf=0;
Geom.gammaf=0;

Geom.lt=0.4;
Geom.gammat=15;
Geom.betat=15;

% ----- %

Aero.CLalphaw=4.04;
Aero.alphaL0w=-11.63;
Aero.CD0w=0.02;
Aero.ArW=Geom.bw^2/Geom.Sw;
Aero.ew=0.88;
Aero.cw=0.47;
Aero.CMp0w=-0.24;
Aero.CMydeltaw=-0.5;
Aero.CMrdeltaw=1;

Aero.CMrRyw=0.1;
Aero.CMrRrw=-1;
Aero.CMyRyw=-1;
Aero.CMyRrw=0.05;

Aero.CLalphaht=1.53;
Aero.alphaL0ht=15;
Aero.CD0ht=0.02;
Aero.CLdeltaht=-1.5;
Aero.Arht=Geom.bht^2/Geom.Sht;
Aero.eht=0.33;
Aero.cht=0.329;
Aero.CMp0ht=0;
Aero.CMpdeltaht=0;

Aero.CL0f=0;
Aero.CLalphaf=0;
Aero.CD0f=0;
Aero.CDalphaf=0;
Aero.cf=0;
Aero.CMp0f=0;
Aero.CMpalphaf=0;

```

```

Aero.Arvt=Geom.bvt^2/Geom.Svt;
Aero.CLalphavt=1;
Aero.CLdeltavt=-1.5;
Aero.CD0vt=0.03;

% ----- %

Aero.Dht=5*pi/180;
Aero.Dvt=-10*pi/180;
Aero.Dw=-5*pi/180;

% ----- %

Prop.a2=-0.02052;
Prop.a1=-0.4153;
Prop.a0=39.39;

Prop.rho0=1.088;
Prop.p=1;

Iner.m0=15;
Iner.J10=0.3;
Iner.J20=0.4;
Iner.J30=0.5;

Iner.dmdt=0;
Iner.dJ1dt=0;
Iner.dJ2dt=0;
Iner.dJ3dt=0;

Iner.theta0=0;

% ----- %

Atmo.u=0;
Atmo.v=0;
Atmo.w=0;

Atmo.du=0;
Atmo.dv=0;
Atmo.dw=0;

Atmo.rho=1.1088;
Atmo.alt=1026;

% ----- %

RV1=zeros(12,1);
RV2=zeros(12,1);
RV1(3)=Inf;
RV2(3)=Inf;

% ----- %

colors=[0      0      0      0.5625;
0      0      0.6250;
0      0      0.6875;
0      0      0.7500;
0      0      0.8125;
0      0      0.8750;
0      0      0.9375;
0      0      1.0000;
0      0.1667      1.0000;
0      0.3333      1.0000;
0      0.5000      1.0000;
0      0.6667      1.0000;
0      0.8333      1.0000;
0      1.0000      1.0000;
0      1.0000      0.8930;
0      1.0000      0.7860;
0      1.0000      0.6790;
0      1.0000      0.5720;
0      1.0000      0.4650;
0      1.0000      0.3580;
0      1.0000      0.2510;
0.0342      1.0000      0.2282;
0.0684      1.0000      0.2053;
0.1027      1.0000      0.1825;
0.1369      1.0000      0.1597;
0.1711      1.0000      0.1369;
0.2053      1.0000      0.1141;
0.2396      1.0000      0.0913;
0.2738      1.0000      0.0684;
0.3080      1.0000      0.0456;
0.3422      1.0000      0.0228;
0.3765      1.0000      0;
0.4544      1.0000      0;
0.5324      1.0000      0;
0.6103      1.0000      0;
0.6882      1.0000      0;
0.7662      1.0000      0;
0.8441      1.0000      0;
0.9221      1.0000      0;
1.0000      1.0000      0;
1.0000      0.9279      0;
1.0000      0.8558      0;
1.0000      0.7837      0;
1.0000      0.7116      0;
1.0000      0.6395      0;
1.0000      0.5674      0;
1.0000      0.4953      0;
1.0000      0.4232      0;
1.0000      0.3511      0;
1.0000      0.2790      0;
1.0000      0.2069      0;
1.0000      0.1348      0;
1.0000      0.0627      0;
1.0000      0      0;
0.9500      0      0;
0.9000      0      0;
0.8500      0      0;
0.8000      0      0;
0.7500      0      0;
0.7000      0      0;
0.6500      0      0;
0.6000      0      0;
0.5500      0      0;
0.5000      0      0];

% ----- %

end

```

ANEXO IV - CD