



UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
Engenharia Mecânica Aeronáutica

TRABALHO DE CONCLUSÃO DE CURSO
- MONOGRAFIA FINAL -

DIRETIVAS PARA PROJETOS DE SISTEMAS DE
AERONAVES REMOTAMENTE PILOTADAS NA
CONFIGURAÇÃO *TAIL-SITTER*

Aluno: Luiz Alberto Monnerat de Amorim
Orientador: Prof. Rogerio Frauendorf de Faria Coimbra
Coorientador: Prof. Marcelo Santiago de Sousa

Agradecimentos

Agradeço primeiramente a Deus por todas as oportunidades que me permitiram chegar até o desenvolvimento deste trabalho.

Agradeço a minha família, começando por minha esposa Carolina e meu filho Arthur, que são fontes de apoio e motivação. Aos meus pais, minha avó e meu tio, agradeço por apoiarem incansavelmente toda minha trajetória desde o ingresso no Colégio Naval.

Agradeço ao CF (EN) Alex pela proposta de trabalho que foi extremamente desafiadora e gratificante.

Agradeço aos professores Rogério, Marcelo e Yohan e ao CT (EN) Marcelo Botelho, pessoas que transmitiram conhecimentos fundamentais para o desenvolvimento deste trabalho.

*“Treinaremos firmemente nosso coração e nosso corpo,
para termos o espírito inabalável”.*

Masutatsu Oyama

Resumo

Este trabalho apresenta análises da implementação da configuração *tail-sitter* em sistemas de aeronaves remotamente pilotadas (SARP). Os principais objetivos são entender as principais diretivas de projeto para seleção de motores, asas e hélices, além de estudar o comportamento em voo por simulação numérica. Esse tipo de aeronave pode realizar pousos e decolagens na vertical (VTOL), o que é uma grande vantagem em operações militares, principalmente em contexto naval. Consequentemente, análises e sugestões de melhorias nesse tipo de configuração podem ser uma contribuição interessante. Para entender as diretivas básicas de projetos, o V-BAT 128, um SARP *tail-sitter* de emprego militar, e dois protótipos acadêmicos foram comparados. As simulações numéricas implementaram o modelo do protótipo acadêmico do V-BAT 128, em *softwares* de simulação. Os resultados indicaram que a teoria da quantidade de movimento e as diretivas de projeto publicadas por Raymer (2018) podem ser uma boa contribuição para um projeto de *tail-sitter*, mas o maior desafio é a implementação de controle para estabilização no voo pairado. As simulações em voo pairado mostraram melhorias com a utilização de controladores PID, mas melhorias ainda são necessárias para estabilizar completamente a aeronave. Outra grande contribuição do trabalho foi o cálculo do trim (condições de equilíbrio da aeronave) para algumas condições de voo, o que permitiu uma melhor compreensão da manobra de transição.

Palavras-chave: Simulação de voo; Projeto de aeronaves; Controle de voo; *Tail-sitter*; *SARP*.

Abstract

This paper presents analysis on the tail-sitter design implementation for UAVs. The main goals are to understand basic project guidelines to select engines, design wings, propellers, and study the behavior in flight by performing numerical simulation. This kind of aircraft can perform vertical take-off and landing (VTOL), which is a great advantage for military operations, especially in naval scenarios. Therefore, evaluation and improvement proposals for tail-sitters can be a good contribution. To understand the basic project guidelines, the V-BAT 128, a tail-sitter UAV designed for military use, and two academic prototypes were compared. The numerical simulation implemented the model of the academic prototype of the V-BAT 128, in simulation software. The results indicated that the basic momentum theory and project guidelines published by Raymer (2018) can be a great contribution to the project of a tail-sitter, but the biggest challenge is the implementation of control to achieve stability in hover flight. Dynamic simulations in hover flight showed some improvements using PID controllers, but further improvements are needed to achieve full stability. Another contribution of this work was the calculation of several trim conditions, which contributed for better understanding the transition maneuver.

Key words: Flight Simulation; Flight Control; Aircraft design; Tail-sitter; UAV.

Lista de Figuras

Figura 1.1 – Convair XFY-1 POGO (lado esquerdo) e Ryan X-13 Vertijet (lado direito).....	2
Figura 1.2 – SNECMA C.450 Coléoptère.....	2
Figura 1.3 – Boeing Isitu ScanEagle.....	3
Figura 1.4 – FlightTech FT-100.....	4
Figura 2.1 – Sistema de eixos do corpo.....	11
Figura 3.1 – Esboço do V-BAT 128.....	16
Figura 3.2 – Estimativa de eficiência aerodinâmica (L/D) máxima.....	17
Figura 3.3 – Vista superior do V-BAT.....	18
Figura 3.4 – Aerofólio NACA 6412.....	19
Figura 3.5 – Curva de C_L x α do aerofólio NACA 6412.....	20
Figura 3.6 – Efeito do alongamento no coeficiente de sustentação.....	20
Figura 3.7 – Eficiência em hélices de 2 pás em função da razão de avanço.....	21
Figura 3.8 – Curvas de potência requerida e disponível.....	22
Figura 3.9 – Diagrama V-n.....	22
Figura 4.1 – Comparação dos resultados do modelo do motor.....	26
Figura 4.2 – Valores de tração no motor quando com velocidade $u = 5$ m/s.....	26
Figura 4.3 – Resultado com as adaptações feitas na modelagem do motor.....	27
Figura 4.4 – Gráfico de C_m x α	28
Figura 4.5 – Resultados de simulação com valor de $C_{y\beta}$ positivo.....	28
Figura 4.6 – Resultados de simulação com valor de $C_{y\beta}$ positivo.....	29
Figura 4.7 – Modelo de simulação.....	31
Figura 4.8 – Ângulos de Euler na vertical.....	32
Figura 4.9 – Controlador de atitude por ângulo de Euler na vertical.....	33
Figura 5.1 – Dados de trim na manobra de transição.....	35
Figura 5.2 – Dados obtidos em voo com o protótipo do V-BAT.....	38
Figura 5.3 – Ângulos de atitude durante simulação de voo pairado sem controlador.....	40
Figura 5.4 – Simulação de voo reto e nivelado.....	41
Figura 5.5 – Resposta a um comando de aileron.....	41

Figura 5.6 – Vista superior da simulação com comando de aileron.....	42
Figura 5.7 – Resposta a um comando de profundor.....	43
Figura 5.8 – Simulação em voo pairado com controladores de atitude.....	44
Figura 5.9 – Oscilações na simulação em voo pairado com controladores de atitude.....	45

Lista de Tabelas

Tabela 2.1 – Dados do modelo de simulação.....	12
Tabela 3.1 – Comparação de massa e potência.....	15
Tabela 3.2 – Dados oficiais sobre o V-BAT 128.....	15
Tabela 3.3 – Informações sobre o motor TOA 288.....	16
Tabela 3.4 – Dados geométricos do V-BAT 128.....	18
Tabela 3.5 – Dados de desempenho calculados para o V-BAT 128.....	19
Tabela 4.1 – Valores fixados no trim pairado.....	30
Tabela 4.2 – Valores fixados no trim de cruzeiro.....	30
Tabela 5.1 – Condições de equilíbrio no voo pairado.....	34
Tabela 5.2 – Condições de equilíbrio no voo de cruzeiro.....	34
Tabela 5.3 – Condições de equilíbrio para velocidades de 0 a 5 m/s.....	36
Tabela 5.4 – Condições de equilíbrio para velocidades de 6 a 10 m/s.....	36
Tabela 5.5 – Condições de equilíbrio para velocidades de 11 a 15 m/s.....	36
Tabela 5.6 – Condições de equilíbrio para velocidades de 16 a 20 m/s.....	37
Tabela 5.7 – Forças e momentos gerados por atuação nos comandos.....	38
Tabela 5.8 – Forças e momentos gerados por ventos no voo pairado.....	39
Tabela 5.9 – Forças e momentos gerados por ventos no voo de cruzeiro.....	39
Tabela 5.10 – Ganhos dos controladores.....	46
Tabela 5.11 – Tabela de comparação de desempenho.....	46

Lista de Siglas e Abreviaturas

FObj – Função Objetivo

PID – Proporcional Integrativo Derivativo

MB – Marinha do Brasil

MTOW – *Maximum Take-Off Weight* (Peso máximo de decolagem)

RPM – Rotações por Minuto

SARP – Sistema de Aeronave Remotamente Pilotada

VTOL – *Vertical Take-off and Landing* (pouso e decolagem vertical)

Lista de Símbolos

g	Aceleração da gravidade (m/s)
h	Altitude (m)
θ	Ângulo de arfagem ($^{\circ}$)
α	Ângulo de ataque ($^{\circ}$)
β	Ângulo de derrapagem ($^{\circ}$)
ψ	Ângulo de proa ($^{\circ}$)
ϕ	Ângulo de rolamento ($^{\circ}$)
S	Área de referência (m ²)
E	Autonomia (h)
C_m	Coeficiente de momento
C_{m0}	Coeficiente de momento para ângulo de ataque nulo
C_L	Coeficiente de sustentação
δ	Comando
l_{vane}	Comprimento de superfície de controle (m)
k_{aug}	Constante de ajuste
C_{bhp}	Consumo de combustível (lb/hp/h)
sfc	Consumo específico de combustível (1/h)
c	Corda (m)
c_p	Corda de pá de hélice (m)
ρ	Densidade do ar (kg/m ³)
$C_{y\beta}$	Derivada de estabilidade que relaciona ângulo de derrapagem e força lateral
$d_{vane,cp}$	Distância de superfície de controle até o centro de pressão (m)
l_{duct}	Distância do centro de gravidade até a entrada do duto (m)
η_p	Eficiência de propulsor
e	Erro de controlador
F	Força (N)
D	Força de arrasto (N)
L	Força de sustentação (N)
$F_{L,vi}$	Força de sustentação na superfície de controle (N)
T	Força de tração (N)
$F_{y,vane}$	Força no eixo y provocada pelas superfícies de controle (N)

$F_{z,vane}$	Força no eixo z provocada pelas superfícies de controle (N)
K_d	Ganho derivativo
K_i	Ganho integrativo
K_p	Ganho proporcional
α_0	Inclinação da curva de sustentação por ângulo de ataque
m	Massa (kg)
J	Matriz de inércia (kg.m ²)
M	Momento (N.m)
b_p	Número de pás
Re	Número de Reynolds
W	Peso (N)
p	Posição (m)
P	Potência (kW)
P_h	Potência para manter voo pairado (kW)
P_v	Potência para manter voo pairado ascendente (kW)
$[\eta_0 \eta_1 \eta_2 \eta_3]$	Quatérnio
r_p	Raio de propulsor (M)
K_{twist}	Torção de pá (rad)
T_h	Tração necessária para manter voo pairado (N)
T_v	Tração necessária para manter voo pairado ascendente (N)
V	Velocidade (m/s)
v_f	Velocidade de escoamento a jusante (m/s)
ω_p	Velocidade de rotação de propulsor (rad/s)
p	Velocidade de rotação no eixo x (rad/s)
q	Velocidade de rotação no eixo y (rad/s)
r	Velocidade de rotação no eixo z (rad/s)
V_v	Velocidade de subida (m/s)
v_b	Velocidade do escoamento na hélice (m/s)
v_h	Velocidade induzida para manter voo pairado (m/s)
$v_{i,v}$	Velocidade induzida para manter voo pairado ascendente (m/s)
u	Velocidade linear no eixo x (m/s)
v	Velocidade linear no eixo y (m/s)
w	Velocidade linear no eixo z (m/s)

μ Viscosidade (kg/m/s)

Subscritos:

ail Aileron

d Baixo (*down*)

c Comando (controlador)

duct Duto

x Eixo x

y Eixo y

z Eixo z

ind Induzido

rud Leme (*rudder*)

e Leste (*east*)

n Norte

elev Profundor (*elevator*)

vane Superfície de controle

thr Tração (*thrust*)

w Vento

v Vertical

SUMÁRIO

CAPÍTULO 1	– INTRODUÇÃO	1
1.1	Introdução	1
1.2	Justificativa	3
1.3	Objetivos	4
1.3.1	<i>Objetivo geral</i>	4
1.3.2	<i>Objetivos específicos</i>	4
1.4	Estrutura do Trabalho	4
1.5	Revisão da Literatura	5
CAPÍTULO 2	– EMBASAMENTO TEÓRICO	6
2.1	Introdução	6
2.2	Sistema de eixos	6
2.3	Quatérnios	7
2.4	Equações de movimento	7
2.4.1	Força gravitacional	8
2.4.2	<i>Grupo motopropulsor</i>	8
2.4.3	<i>Duto</i>	8
2.4.4	<i>Superfícies de controle</i>	9
2.4.5	<i>Conjunto asa-fuselagem</i>	11
2.4.6	<i>Constantes da aeronave</i>	12
2.5	Otimização	12

2.6	Solução numérica de equações diferenciais	13
2.7	Controladores PID	13
CAPÍTULO 3 – ANÁLISES DE PROJETO CONCEITUAL		14
3.1	Estimativa de potência necessária para voo pairado	14
3.2	Estimativas de peso de combustível e desempenho	15
3.3	Considerações sobre o projeto da asa	19
3.4	Considerações sobre o projeto do propulsor	21
3.5	Diagrama V-n	22
CAPÍTULO 4 – METODOLOGIA		24
4.1	Introdução	24
4.2	Modelagem Computacional	24
4.2.1	<i>Variáveis de estados e controles</i>	24
4.2.2	<i>Ajustes nas equações do grupo motopropulsor</i>	25
4.2.3	<i>Ajuste nas equações das superfícies de controle</i>	27
4.2.4	<i>Ajustes nas equações do conjunto asa-fuselagem</i>	27
4.3	Cálculos de trim	29
4.3.1	<i>Trim no voo pairado</i>	29
4.3.2	<i>Trim de cruzeiro</i>	30
4.4	Simulações dinâmicas	31
4.5	Controle de atitude em voo pairado	32
CAPÍTULO 5 – RESULTADOS E DISCUSSÃO		34
5.1	Resultados de trim	34
5.1.1	<i>Voo pairado</i>	34
5.1.2	<i>Voo de cruzeiro</i>	34
5.1.3	<i>Análise do trim na manobra de transição</i>	35
5.2	Testes aplicados no modelo	38
5.2.1	<i>Testes de comandos</i>	38
5.2.2	<i>Testes aerodinâmicos</i>	39

5.3	Simulações em malha aberta	39
5.3.1	<i>Voo pairado</i>	39
5.3.2	<i>Voo reto e nivelado</i>	40
5.3.3	<i>Resposta aos comandos latero-direcionais</i>	41
5.3.4	<i>Resposta ao comando de profundor</i>	42
5.4	Simulações com controlador de atitude em voo pairado	43
CAPÍTULO 6	– CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	46
6.1	Conclusões	48
6.2	Sugestões para Trabalhos Futuros	49
REFERÊNCIAS BIBLIOGRÁFICAS		50
APÊNDICE A	– Código do bloco s-function	52
APÊNDICE B	– Funções	61
APÊNDICE C	– Código de teste do motor antes das correções	63
APÊNDICE D	– Código de trim no voo pairado	65
APÊNDICE E	– Código de trim no voo de cruzeiro	77
APÊNDICE F	– Código de trim em transição	90

6– INTRODUÇÃO

1.1 Introdução

As iniciativas de projeto de aeronaves com capacidade de pouso e decolagem na vertical (VTOL) ganharam importância por ocasião da segunda guerra mundial, pois durante o conflito foi observado que as pistas de pouso eram frequentes alvos de bombardeios e ataques. Outra vantagem da capacidade VTOL é a possibilidade de a aeronave operar a partir de meios navais de menor porte, além disso a importância da operação a partir de meios navais também foi muito valorizada durante o conflito devido ao grande número de operações anfíbias realizadas no teatro do pacífico e pela operação *Overlord* na Europa, descrita em Speller e Tuck (2001). Nesse tipo de operação, o emprego de aeronaves foi fundamental para cumprir diversos tipos de missão, como reconhecimento, observação, apoio de fogo, lançamento de paraquedistas, entre outras.

Nos primeiros projetos era comum a utilização da configuração como *tail-sitter*, que se caracteriza por decolar e pousar sobre sua empenagem e durante a transição entre voo pairado e voo de cruzeiro, toda a aeronave é rotacionada, conforme relatado por Felix, Braun e Bil (2017).

Os principais projetos de *tail-sitter* que conseguiram construir protótipos foram o Convair XFY-1 POGO, Ryan X-13 Vertijet (COWIN, 1999) e o SNECMA C.450 *Coléoptère* (BUTTLER & DELEZENNE, 2012). O POGO adotou a propulsão por 2 hélices contra rotativas frontais e sistemas de comandos de voo por deflexão de superfícies, enquanto os outros 2 utilizaram motores a jato e direcionamento de empuxo para realizar o controle no voo pairado. Essas aeronaves podem ser observadas nas Figuras 1.1 e 1.2.

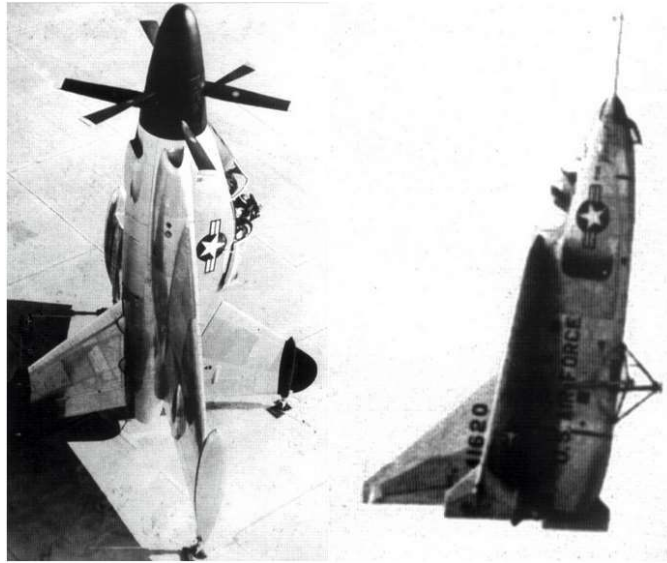


Figura 1.1 – Convair XFY-1 POGO (lado esquerdo) e Ryan X-13 Vertijet (lado direito). (Fonte: COWIN, 1999).



Figura 1.2 – SNECMA C.450 Coléoptère (Fonte: BUTTLER & DELEZENNE, 2012).

Apesar do POGO e do Vertijet terem realizado pousos e decolagens bem-sucedidas, os pilotos relatavam grande dificuldade para realizar o pouso, pois tinham que olhar para trás por cima dos ombros enquanto estabilizavam a aeronave. Enquanto isso o francês C.450 não conseguiu se estabilizar após sua primeira decolagem, consequentemente o piloto ejetou e seu único protótipo foi destruído.

As dificuldades encontradas nos projetos citados e o sucesso da configuração que resultou no *Hawker Harrier* britânico, citado em Felix, Braun e Bil (2017), foram os principais fatores que contribuíram para o fracasso da configuração *tail-sitter* no período da guerra fria.

Atualmente essa configuração está sendo aplicada em Sistemas de Aeronave Remotamente Pilotada (SARP), o que não coloca vidas em risco. Além disso, o avanço das tecnologias de eletrônica e *software* embarcado ajudam a contornar os problemas de estabilidade.

Analisando o recente lançamento do modelo V-BAT 128 e algumas publicações acadêmicas é possível fazer uma análise dos avanços nesse tipo de configuração. Essa análise é muito valiosa para conhecer as características dos SARP *tail-sitter* e fazer considerações para projetos futuros.

1.2 Justificativa

O emprego dos SARP é muito valioso em operações militares. Brasil (2020) descreve as missões de reconhecimento aéreo e observação aérea, ambas podem ser realizadas por aeronaves do porte do V-BAT 128.

Missões de reconhecimento aéreo têm a finalidade de obter informações ou cobertura fotográfica de determinada região. Já as missões de observação aérea são feitas para conduzir fogos de artilharia ou fogo naval, podendo também realizar tarefas de uma missão de reconhecimento (BRASIL, 2020).

A Marinha do Brasil (MB) já opera alguns SARP de asa fixa, que são o Boeing Insitu ScanEagle (Figura 1.3) e o FlightTech FT-100 (Figura 1.4). Nenhum deles possui a capacidade VTOL, que seria grande vantagem, principalmente para o uso naval, pois possibilitaria a operação em navios sem a instalação de equipamentos especiais para lançamento e recolhimento.



Figura 1.3 – Boeing Isitu ScanEagle (Fonte: INSITU, 2022).

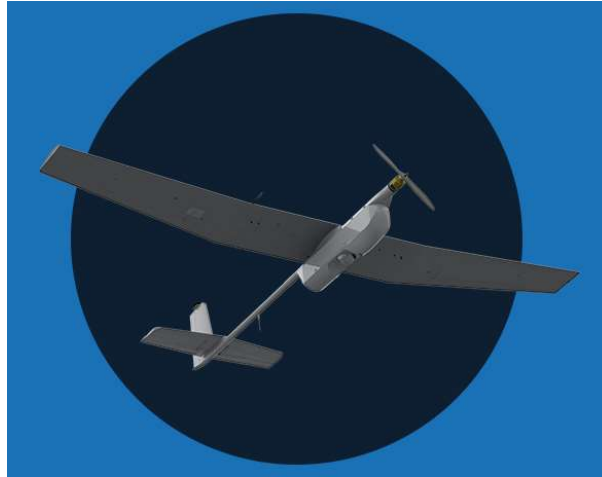


Figura 1.4 – FlightTech FT-100 (Fonte: FT SISTEMAS, 2022).

Por outro lado, a configuração *tail-sitter* apresenta capacidade VTOL, mas não é muito difundida no mercado e traz questionamentos sobre sua estabilidade. Por esse motivo é necessário realizar estudos sobre sua dinâmica, gerando um maior entendimento sobre o comportamento em voo e propostas para melhoria.

1.3 Objetivos

1.3.1 *Objetivo geral*

O objetivo deste trabalho é compreender o comportamento de um SARP *tail-sitter* em voo, vantagens, limitações e propor modificações e melhorias, considerando o emprego no contexto naval.

1.3.2 *Objetivos específicos*

Para atingir o objetivo geral deste trabalho, será necessário:

- Entender dimensionamento de grupo motopropulsor, peso de combustível e seleção de perfis aerodinâmicos utilizando conceitos de projetos conceituais de aeronaves;
- Calcular dados de desempenho;
- Elaborar um código de simulação para entender as características de voo; e
- Estudar a implementação de leis de controle.

1.4 Estrutura do Trabalho

No Capítulo 1 está apresentada a introdução do trabalho, que fornece uma revisão histórica sobre a configuração *tail-sitter*, além de apresentar a justificativa, os objetivos e uma revisão literária citando as principais publicações que embasaram o trabalho.

No Capítulo 2 são apresentados conceitos básicos, como sistemas de eixos utilizados, ferramentas de otimização, representação de atitude por quatérnios, algoritmo de Runge-Kutta para integração das equações descritivas da dinâmica de voo.

No Capítulo 3 o V-BAT 128 é analisado do ponto de vista de projeto conceitual, definindo quais devem ser as diretivas para seleção do grupo motopropulsor e perfis aerodinâmicos. Além disso, serão apresentados cálculos de desempenho e diagrama V-n

No Capítulo 4 é feita uma apresentação detalhada sobre os ajustes feitos no modelo utilizado, quais foram as condições definidas nos cálculos de trim, simulações dinâmicas e a implementação de controladores PID.

No Capítulo 5 são apresentados os resultados obtidos e comentários relacionados.

No Capítulo 6 está a conclusão e sugestões para trabalhos futuros.

1.5 Revisão da Literatura

O livro de Raymer (2018) contém os conceitos sobre projetos de aeronaves utilizados na análise do V-BAT 128. Felix, Braun e Bil (2017) apresentam uma boa definição para a configuração *tail-sitter*, juntamente com uma revisão sobre as principais configurações de aeronave VTOL utilizadas.

Cowin (1999), Buttler e Delezenne (2012) e Speller e Tuck (2001) forneceram informações essenciais para a contextualização histórica apresentada. Brasil (2020) foi consultado para descrever o tipo de missão que as aeronaves estudadas podem realizar. Além disso, diversas aplicações civis e militares dos SARP são citadas em Chaturvedi *et al.* (2019).

Argyle (2016), Zamri (2010), Johnson e Turbe (2005) são trabalhos acadêmicos dedicados à modelagem, prototipagem e implementação de leis de controle em SARP *tail-sitters*. O modelo definido em Argyle (2016) foi escolhido para simulação, devido a intenção de compreender melhor o comportamento de um *tail-sitter* em voo. Zamri (2010), Johnson e Turbe (2005) auxiliaram através de comparações e foram importantes para citar diferentes técnicas de controle que podem ser utilizadas em trabalhos futuros.

Ogata (2015) apresenta os conceitos básicos sobre controladores PID, que foram utilizados em Argyle (2016) e nas simulações deste trabalho. O método numérico utilizado para realizar as simulações dinâmicas, o algoritmo de Runge-Kutta, é descrito em Chapra (2012). As demais publicações referenciadas são fontes de dados e figuras.

CAPÍTULO 2 – EMBASAMENTO TEÓRICO

2.1 Introdução

Este Capítulo cita as técnicas empregadas para os cálculos de condições de equilíbrio (trimagem) e simulação dinâmica do voo. Além disso, é apresentado o sistema de representação de atitude por quatérnios, que se tornou necessário em função das manobras com ângulo de atitude igual a 90° . Essa atitude leva ao efeito *gimbal lock*, que causa uma divisão por zero durante a simulação numérica com ângulos de Euler.

2.2 Sistema de eixos

O sistema de eixos adotado é o mesmo adotado em Argyle (2016), que é um sistema normalmente utilizado na literatura de dinâmica de voo, apresentado também em Stevens, Lewis e Johnson (2016). A posição dos eixos com relação a aeronave e os ângulos de ataque (α) e derrapagem (β), que indicam a direção do vento relativo, podem ser observados na Figura 2.1:

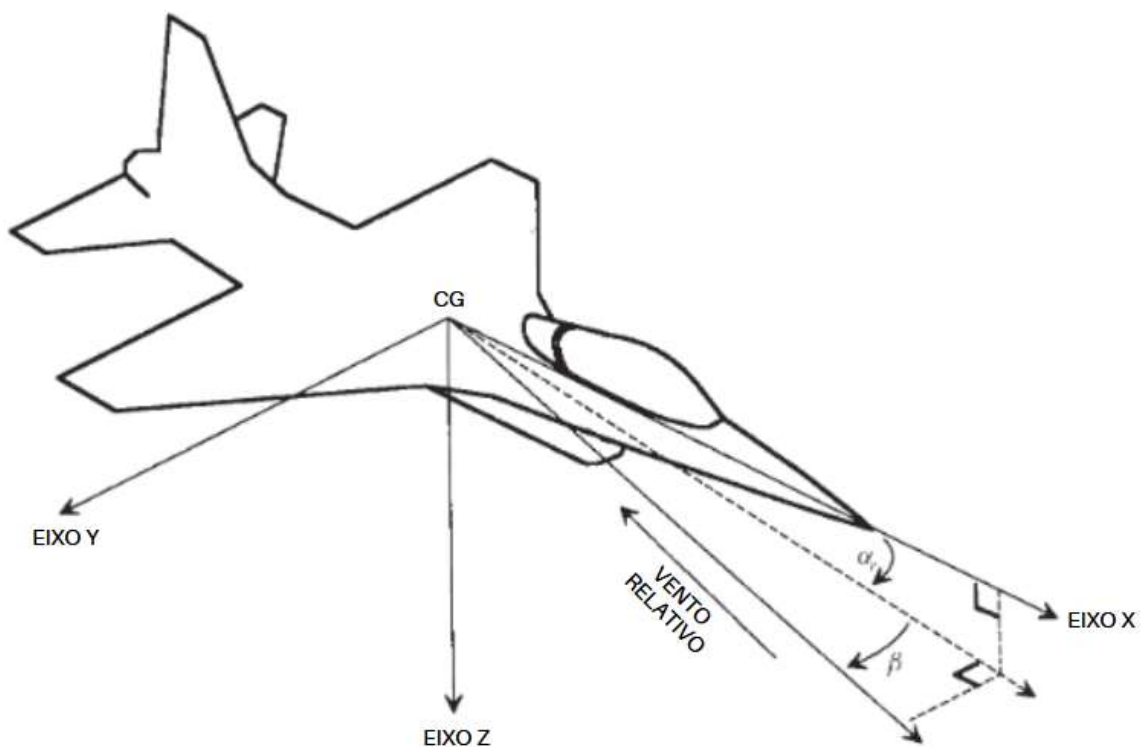


Figura 2.1 – Sistema de eixos do corpo (FONTE: Adaptado de: Stevens, Lewis e Johnson, 2016)

O sistema de eixos da terra tem o eixo x apontado para o norte, o eixo y apontando para o leste e o eixo z apontando para baixo. O sistema de eixos da terra tem a finalidade de localizar a aeronave em relação a um ponto de referência em um plano.

2.3 Quatérnios

O quatérnio de atitude representa a rotação entre o sistema de coordenadas inicial e o sistema de eixos do corpo. Possui 4 componentes, diferentemente dos ângulos de Euler que possuem 3. Segundo Argyle (2016) o quatérnio possui o formato da Equação 2.1.

$$\eta = \eta_0 + \eta_1\hat{i} + \eta_2\hat{j} + \eta_3\hat{k} \quad (2.1)$$

Reescrevendo um quatérnio na forma da Equação 2.2, representa-se uma rotação de γ graus em torno do vetor definido pelos componentes imaginários.

$$\eta = \cos\left(\frac{\gamma}{2}\right)\eta_0 + \sin\left(\frac{\gamma}{2}\right)(\eta_1\hat{i} + \eta_2\hat{j} + \eta_3\hat{k}) \quad (2.2)$$

A utilização de ângulo de Euler causaria erro na simulação, pois ocorreria divisão por zero na Equação 2.3 de Stevens, Lewis e Johnson (2016).

$$\dot{\psi} = \frac{[q\sin(\phi) + r\cos(\phi)]}{\cos(\theta)} \quad (2.3)$$

2.4 Equações de movimento

Foram utilizadas as equações de movimento descritas por Argyle (2016) com algumas adaptações contidas no Capítulo 3. A cinemática da aeronave é definida pelas Equações 2.4 a 2.7:

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} 2\eta_0^2 + 2\eta_1^2 - 1 & 2(\eta_1\eta_2 - \eta_3\eta_0) & 2(\eta_1\eta_3 + \eta_2\eta_0) \\ 2(\eta_1\eta_2 + \eta_3\eta_0) & 2\eta_0^2 + 2\eta_2^2 - 1 & 2(\eta_2\eta_3 - \eta_1\eta_0) \\ 2(\eta_1\eta_3 - \eta_2\eta_0) & 2(\eta_2\eta_3 + \eta_1\eta_0) & 2\eta_0^2 + 2\eta_3^2 - 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = -\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (2.5)$$

$$\dot{\eta} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \quad (2.6)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = J^{-1} \left(- \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times J \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} \right) \quad (2.7)$$

Sendo que p_n , p_e e p_d representam a posição (posições norte, leste e baixo), u , v e w são as velocidades lineares, p , q e r são as velocidades angulares, η é o quatérnio que representa a atitude, J é a matriz de inércia, F_x , F_y e F_z são as forças externas resultantes e M_x , M_y e M_z são os momentos resultantes.

2.4.1 Força gravitacional

A força da gravidade é decomposta para o sistema de eixos do corpo através da Equação 2.8:

$$F_g = mg \begin{bmatrix} 2(\eta_1\eta_3 - \eta_2\eta_0) \\ 2(\eta_2\eta_3 - \eta_1\eta_0) \\ 2\eta_0^2 + 2\eta_3^2 - 1 \end{bmatrix} \quad (2.8)$$

2.4.2 Grupo motopropulsor

Neste modelo é necessário calcular a tração e o torque gerados pela hélice, assim como a velocidade induzida do escoamento de ar, pois esta influencia outras forças atuantes no modelo. Esses cálculos são apresentados em Argyle (2016) e Johnson e Turbe (2005), trata-se de uma solução iterativa de um sistema com 3 equações que têm como base a teoria da quantidade de movimento e a teoria do elemento da pá. Nessa parte do modelo foram feitas algumas adaptações que estão detalhadas no Capítulo 4.

2.4.3 Duto

O duto é a estrutura tubular que envolve a hélice. A velocidade de entrada no duto (V_0) é calculada pela Equação 2.9:

$$V_0 = \sqrt{(u - u_w)^2 + (v - v_w + r l_{duct})^2 + (w - w_w + q l_{duct})^2} \quad (2.9)$$

O subscrito w indica as velocidades do vento no sistema de eixos do corpo, l_{duct} é a distância do centro de gravidade até a entrada do duto.

Os ângulos de ataque e derrapagem no duto são calculados pelas Equações 2.10 e 2.11:

$$\alpha_{duct} = tg^{-1} \left(\frac{\sqrt{(v - v_w + r l_{duct})^2 + (w - w_w + q l_{duct})^2}}{(u - u_w)^2} \right) \quad (2.10)$$

$$\beta_{duct} = tg^{-1} \left(\frac{v - v_w + r l_{duct}}{w - w_w + q l_{duct}} \right) \quad (2.11)$$

A força de arrasto no duto ($F_{d,duct}$) é calculada pela Equação 2.12:

$$F_{d,duct} = \rho \pi r_{duct}^2 (V_0 \cos(\alpha_{duct}) + v_i) V_0 \sin(\alpha_{duct}) \quad (2.12)$$

Onde ρ é a densidade do ar, r_{duct} é o raio do duto. v_i é a velocidade induzida pelo propulsor e é calculada pela rotina de cálculos das forças e momentos do grupo motopropulsor. A força de arrasto no duto é decomposta para o sistema de eixos do corpo pela Equação 2.13:

$$F_{duct} = \begin{bmatrix} 0 \\ -F_{d,duct} \sin(\beta_{duct}) \\ -F_{d,duct} \cos(\beta_{duct}) \end{bmatrix} \quad (2.13)$$

Os momentos provocados por estas forças são calculados pela Equação 2.14:

$$M_{duct} = \begin{bmatrix} 0 \\ -F_{d,duct} \cos(\beta_{duct}) l_{duct} \\ F_{d,duct} \sin(\beta_{duct}) l_{duct} \end{bmatrix} \quad (2.14)$$

2.4.4 Superfícies de controle

Os comandos de aileron, profundor e leme (δ_{ail} , δ_{elev} e δ_{rud}) são distribuídos para as 8 superfícies de controle através de somas saturadas descritas pelas Equações 2.15 a 2.22 de Argyle (2016):

$$\delta_{vane1} = sat(-\delta_{ail} + \delta_{rud}, -1, 1) \quad (2.15)$$

$$\delta_{vane2} = sat(-\delta_{ail} + \delta_{elev} + \delta_{rud}, -1, 1) \quad (2.16)$$

$$\delta_{vane3} = sat(-\delta_{ail} + \delta_{elev}, -1, 1) \quad (2.17)$$

$$\delta_{vane4} = sat(-\delta_{ail} + \delta_{elev} - \delta_{rud}, -1, 1) \quad (2.18)$$

$$\delta_{vane5} = sat(-\delta_{ail} - \delta_{rud}, -1, 1) \quad (2.19)$$

$$\delta_{vane6} = sat(-\delta_{ail} - \delta_{elev} - \delta_{rud}, -1, 1) \quad (2.20)$$

$$\delta_{vane7} = sat(-\delta_{ail} - \delta_{elev}, -1, 1) \quad (2.21)$$

$$\delta_{vane8} = sat(-\delta_{ail} - \delta_{elev} + \delta_{rud}, -1, 1) \quad (2.22)$$

As forças de sustentação ($F_{L,vane}$) e arrasto ($F_{D,vane}$) em cada superfície são calculadas pelas Equações 2.23 e 2.24 de Argyle (2016):

$$F_{L,vane\ i} = \frac{1}{2} \rho v_{vane}^2 S_{vane} \theta_{vane\ i} C_{L,vane} \quad (2.23)$$

$$F_{D,vane\ i} = \frac{1}{2} \rho v_{vane}^2 S_{vane} \theta_{vane\ i} C_{D,vane} \quad (2.24)$$

A velocidade v_{vane} é a soma da velocidade induzida com componente x da velocidade de entrada no duto, S_{vane} é a área da superfície de controle, $\theta_{vane\ i}$ é a deflexão da superfície em radianos, $C_{L,vane}$ e $C_{D,vane}$ são respectivamente os coeficientes de sustentação e arrasto das superfícies de controle.

Essas forças podem ser decompostas para o sistema de eixos do corpo pela equação 2.25 de Argyle (2016):

$$F_{vane} = \begin{bmatrix} -\sum_{i=1}^8 |F_{D,vane\ i}| \\ \sum_{i=1}^8 \cos\left((i-1)\frac{\pi}{4}\right) F_{L,vane\ i} \\ \sum_{i=1}^8 \sin\left((i-1)\frac{\pi}{4}\right) F_{L,vane\ i} \end{bmatrix} \quad (2.25)$$

A equação que calcula os momentos resultantes foi ajustada e está descrita no Capítulo 3.

2.4.5 Conjunto asa-fuselagem

Para calcular as forças e momentos aerodinâmicos, a velocidade do ar e os ângulos de ataque e derrapagem são calculados respectivamente pelas Equações 2.26 a 2.28 de Argyle (2016):

$$V_a = \sqrt{(u - u_w)^2 + (v - v_w)^2 + (w - w_w)^2} \quad (2.26)$$

$$\alpha = \tan^{-1} \left(\frac{w - w_w}{u - u_w} \right) \quad (2.27)$$

$$\beta = \sin^{-1} \left(\frac{v - v_w}{V_{ar}} \right) \quad (2.28)$$

O cálculo das forças e momentos é feito pelas Equações 2.29 e 2.30 de Argyle (2016):

$$F_{asa-fuselagem} = \frac{1}{2} \rho S V_a^2 \begin{bmatrix} C_x + \frac{c}{2V_a} C_{x_q} q \\ C_y + \frac{b}{2V_a} (C_{y_p} p + C_{y_r} r) \\ C_z + \frac{c}{2V_a} C_{z_q} q \end{bmatrix} \quad (2.29)$$

$$M_{asa-fuselagem} = \frac{1}{2} \rho S V_a^2 \begin{bmatrix} b \left(C_l + \frac{b}{2V_a} (C_{l_p} p + C_{l_r} r) \right) \\ c \left(C_m + \frac{c}{2V_a} C_{m_q} q \right) \\ b \left(C_n + \frac{b}{2V_a} (C_{n_p} p + C_{n_r} r) \right) \end{bmatrix} \quad (2.30)$$

Os coeficientes aerodinâmicos C_x , C_y , C_z , C_l , C_m e C_n são influenciados por algumas derivadas de estabilidade e pelos estados da aeronave, que são suas velocidades e ângulos de ataque, derrapagem e atitude. Para calcular os coeficientes aerodinâmicos existem dois procedimentos, um para velocidades acima de 10 m/s e outro para velocidades abaixo ou iguais a esse valor. O modelo utilizado abaixo dos 10 m/s é mais voltado para o voo pairado, enquanto

o modelo utilizado acima de 10 m/s é focado no voo convencional. O detalhamento desses cálculos pode ser observado em Argyle (2016) ou no Apêndice A.

2.4.6 Constantes da aeronave

Tabela 2.1 – Dados do modelo de simulação

Dados gerais		$C_{m\alpha}$	-0,712
Massa (kg)	5,443	$C_{y\beta}$	0,125
J_x (kg m ²)	0,3745	$C_{l\beta}$	-0,167
J_y (kg m ²)	1,4327	$C_{n\beta}$	0,0475
J_z (kg m ²)	15,723	Dados do propulsor	
J_{xz} (kg m ²)	0,015	K_{twist}	0,4860
S (m ²)	0,696	a_0	1,3860
b (m)	1,8479	b_p	2
c (m)	0,2	c_p	0,04
$C_{y,0}$	0	r_p	0,2286
$C_{l,0}$	0	k_{aug}	0,4548
$C_{n,0}$	0	Dados das superfícies de controle	
C_{Lq}	2,8932	$C_{L,v}$	0,25
C_{dq}	0	$C_{D,v}$	0,1659
C_{mq}	-1,3990	S_v (m ²)	0,17
C_{yp}	0	$d_{v,cp}$ (m)	0,71
C_{yr}	0	l_v (m)	0,15
C_{lp}	-0,3209	Dados do duto	
C_{lr}	0,03066	c_{duct} (m)	0,1524
C_{np}	0,011297	r_{duct} (m)	0,2377
C_{nr}	-0,00434	l_{duct} (m)	0,7
C_{m0}	-0,0721		

Fonte: ARGYLE (2016).

2.5 Otimização

Para os cálculos de condições de equilíbrio da aeronave, foram utilizadas funções internas de um *software* de simulação que aplicam técnicas de otimização, que é um procedimento que calcula iterativamente o valor mínimo de uma função. Essa técnica é empregada rotineiramente na literatura de dinâmica em voo.

Nos pontos de equilíbrio, o sistema deve apresentar acelerações lineares e angulares nulas. Além disso, outros parâmetros podem ser definidos para os pontos de equilíbrio, neste

trabalho também foi considerada a variação de altitude nula. Consequentemente a função objetivo definida foi a soma dos quadrados dos parâmetros selecionados, o que significa que quando a otimização for bem-sucedida, o valor da função objetivo será próximo de zero.

As variáveis de entrada da função objetivo são compostas por comandos de voo e alguns estados da aeronave. Os estados que não fazem parte da função objetivo e suas variáveis de entrada devem ser definidos como constantes. O detalhamento completo dos procedimentos de otimização é apresentado no Capítulo 4.

2.6 Solução numérica de equações diferenciais

Para fazer as simulações dinâmicas foram utilizados algoritmos de Runge-Kutta de quarta e quinta ordem através de uma função interna de *software* de simulação. Esse algoritmo resolve as equações diferenciais da dinâmica de voo, fazendo a integração ao longo do tempo. Esse algoritmo é detalhado em Chapra (2012).

2.7 Controladores PID

Estes controladores foram implementados nos comandos de aileron, profundor e leme a fim de manter a ARP em equilíbrio no voo pairado durante a simulação. Trata-se de um controlador simples e amplamente utilizado na engenharia, cujas definições são apresentadas em Ogata (2010).

CAPÍTULO 3 – ANÁLISES DE PROJETO CONCEITUAL

3.1 Estimativa de potência necessária para voo pairado

Zamri (2010) apresenta uma sequência de cálculos que permite fazer estimativas para a potência necessária para o voo pairado estático e para o voo pairado ascendente. Os cálculos são baseados na teoria da quantidade de movimento. Primeiramente calcula-se a velocidade induzida pela hélice para manter o voo pairado pela Equação 3.1.

$$v_h = \sqrt{\frac{T_h}{2\rho\pi r_p^2}} \quad (3.1)$$

Onde v_h e T_h são respectivamente a velocidade induzida e a tração necessárias para o voo pairado, ρ é a densidade do ar e r_p é o raio da hélice. Multiplicando a velocidade induzida pela tração encontra-se a potência necessária para voo pairado (P_h), conforme a Equação 3.2.

$$P_h = T_h v_h = T_h \sqrt{\frac{T_h}{2\rho\pi r_p^2}} \quad (3.2)$$

Para o voo pairado ascendente com velocidade V_v , a velocidade induzida necessária ($v_{i,v}$) é calculada pela Equação 3.3:

$$v_{i,v} = -0,5V_v + v_h \sqrt{\left(\frac{V_v}{2v_h}\right)^2 + 1} \quad (3.3)$$

Logo a potência para o voo pairado ascendente em uma velocidade V_v é definida pela Equação 3.4:

$$P_v = T_v V_v + T_h v_{i,v} \quad (3.4)$$

A tração T_h é igual ao peso do veículo e T_v é igual ao peso do veículo acrescido de uma estimativa de arrasto.

Utilizando as equações acima pode-se fazer uma comparação entre os protótipos utilizados por Argyle (2016) e Zamri (2010), além de compará-los com o V-BAT 128. Os dados obtidos estão na Tabela 3.1.

Tabela 3.1 – Comparação de massa e potência.

	V-BAT 128	Zamri (2010)	Argyle (2016)
Massa (Kg)	56,7	2,5	5,443
Potência para voo pairado (kW)	13,3	0,412	0,615
Potência para voo pairado ascendente a 8 m/s (kW)	15,723	0,546	0,88
Potência do motor (kW)	17,6	0,6	-

Fonte: ARGYLE (2016); SHIELD AI, 2022 e ZAMRI, 2010.

Para os cálculos do V-BAT 128 foi considerado uma hélice com 28 polegadas ou 0,711 m. Para o protótipo de Argyle (2016), o raio da hélice foi informado na tese.

A análise dessas informações mostra que os motores selecionados nos projetos não apresentam muita sobra de potência em relação ao que é necessário no voo pairado. Isso indica que a hélice deve ser otimizada para esta condição de voo e isso pode ter como consequência um pior desempenho da hélice em altas velocidades.

3.2 Estimativas de peso de combustível e desempenho

A sequência de cálculos parte dos dados oficialmente divulgados sobre o V-BAT 128 para estimar o seu peso de combustível e alguns dados desconhecidos sobre seu desempenho. Os dados fornecidos oficialmente estão na Tabela 3.2:

Tabela 3.2 – Dados oficiais sobre o V-BAT 128.

Envergadura	3 m
Comprimento	2,74 m
Massa	56,7 Kg
Autonomia	10 h
Autonomia (voo pairado)	45 min
Carga paga	11,34 Kg
Teto de serviço	20000 pés

Fonte: SHIELD AI, 2022.

Os dados utilizados do motor TOA 288, utilizado no V-BAT 128, estão na Tabela 3.3.

Tabela 3.3 – Informações sobre o motor TOA 288.

Potência máxima	17,6 kW
Rotação máxima	6500 RPM
Consumo de combustível (4700 RPM e 6,7 kW)	317 g/kW/h
Consumo de combustível (6500 RPM e 17,6 kW)	454 g/kW/h

Fonte: SUTER INDUSTRIES, 2022.

Inicialmente para uma estimativa de eficiência aerodinâmica máxima, foi utilizado o método descrito em Raymer (2018). Para utilizar esse método, foi feita uma estimativa de área molhada através de um esboço no *software* de desenho, que pode ser observado na figura 3.1:

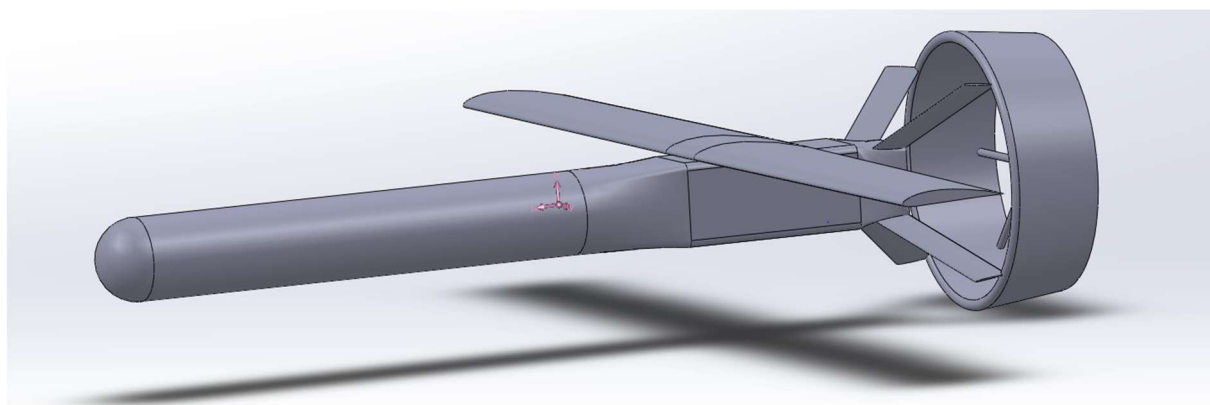


Figura 3.1 – Esboço do V-BAT 128

Conhecendo a área molhada e a envergadura, é possível estimar a máxima eficiência aerodinâmica através da Figura 3.2:

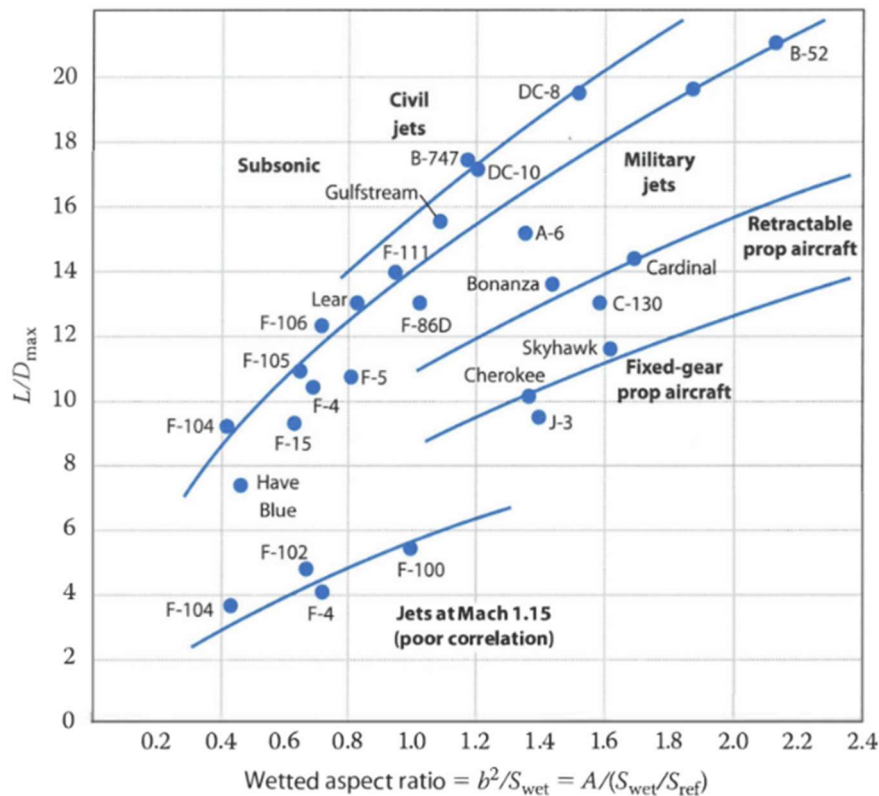


Figura 3.2 – Estimativa de eficiência aerodinâmica (L/D) máxima (Fonte: Raymer, 2018).

Por causa da geometria do duto, que é uma grande causa de arrasto, foi utilizada a curva de aeronaves a hélice com trem de pouso fixo. Isso resulta em uma estimativa de 11 para a máxima eficiência aerodinâmica.

Considerando o consumo de combustível (C_{bhp}) fornecido na ficha técnica do motor Suter TOA 288 (0,52 lb/bhp/h para 4700 RPM e 6,7kW) e as equações de apresentadas em Raymer (2018), estima-se o peso de combustível. Foram consideradas as seguintes hipóteses: eficiência de 0,6 na hélice, velocidade de cruzeiro (V) de 30 m/s. Além disso, foi considerada a autonomia (E) de 10 horas informada pelo fabricante. Esses cálculos foram realizados utilizando as Equações 3.5 e 3.6 de Raymer (2018):

$$sfc = C_{bhp} \frac{V}{550\eta_p} \quad (3.5)$$

$$E = \frac{L/D}{sfc} \ln \left(\frac{W_{i-1}}{W_i} \right) \quad (3.6)$$

Sendo sfc O consumo específico de combustível. W_{i-1} e W_i são os pesos inicial e final da aeronave. Consequentemente, o valor estimado inicialmente para o peso de combustível é de 8,48 kgf (20 lbs).

A fim de confirmar os valores calculados anteriormente, foi feita uma sequência de cálculos com os dados geométricos estimados através da Figura 3.3, utilizando proporcionalidade em relação à envergadura. Com estes dados foram calculados os dados da Tabela 3.4:

Tabela 3.4 – Dados geométricos do V-BAT 128.

Envergadura	3 m
Afilamento	0,58
Área de referência	0,7485 m ²
Alongamento	12,02
Corda média aerodinâmica	0,25 m



Figura 3.3 – Vista superior do V-BAT 128 (Fonte: KAUFFMAN, 2020)

Conhecendo esses valores estima-se o coeficiente de Oswald pela equação fornecida em Raymer (2018). Além disso, foi considerado o coeficiente de arrasto parasita para sustentação nula fornecido no protótipo de Argyle (2016). Utilizando esses valores, foram calculadas a curva polar de arrasto, e a curva de potência requerida. Com os valores de coeficientes de sustentação e arrasto para cada velocidade, foram estimadas as velocidades de máximo alcance e máxima autonomia.

Ao recalculer a autonomia, foi encontrada uma pequena divergência em relação ao informado pelo fabricante. Ajustando o peso de combustível para 10 kgf, calcula-se uma autonomia em torno de 10,5 h, muito próximo ao valor informado oficialmente. A máxima eficiência aerodinâmica calculada a partir da polar de arrasto foi 10, o valor inicialmente

considerado para a primeira estimativa de peso de combustível foi 11. Isso indica que os valores calculados até agora são coerentes e podem ser observados na Tabela 3.5:

Tabela 3.5 – Dados de desempenho calculados para o V-BAT 128.

Autonomia	10,57 h
Alcance	727,68 km
Velocidade de máxima autonomia	25 m/s
Velocidade de máximo alcance	33,2 m/s

3.3 Considerações sobre o projeto da asa

Considerando o tipo de missão para qual este SARP será empregado, é razoável otimizar a asa para a condição de maior alcance, pois apenas nas missões de observação o SARP pode precisar fazer voo de espera e apenas durante um trecho da missão. Resumidamente, a condição de voo de cruzeiro será predominante nas missões. Além disso, a velocidade de máximo alcance é superior à velocidade de máxima autonomia, o que significa que na condição de máximo alcance o veículo cumprirá missões mais rápido e poderá visualizar alvos mais distantes.

Consequentemente o passo inicial para a seleção do perfil da asa deve ser o cálculo de coeficiente de sustentação no peso médio de cruzeiro e velocidade de máximo alcance. Para uma altitude de 10.000 pés na atmosfera padrão, o coeficiente de sustentação nessas condições será em torno de 1,36.

Outro parâmetro que precisa ser considerado na seleção de um perfil aerodinâmico é o número de Reynolds (Re), que é calculado pela Equação 3.7 e resultará em torno de 440.000.

$$Re = \frac{\rho V c}{\mu} = \frac{(0,9046 \text{ kg/m}^3)(33,2 \text{ m/s})(0,25 \text{ m})}{(1,71 \cdot 10^{-5} \text{ kg/(m} \cdot \text{s)})} \quad (3.7)$$

Sendo ρ densidade do ar, μ a viscosidade dinâmica do ar, V a velocidade do escoamento e c a corda da asa.

Nesse caso o perfil NACA 6412 seria um bom candidato, seus dados estão nas Figuras 3.4 e 3.5:

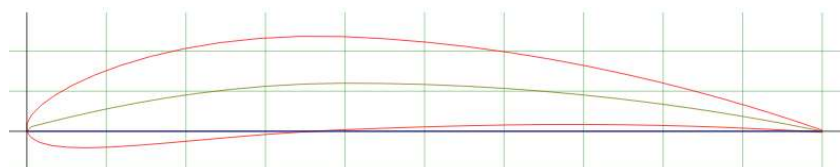


Figura 3.4 – Aerofólio NACA 6412 (Fonte: *AIRFOIL TOOLS*, 2022).

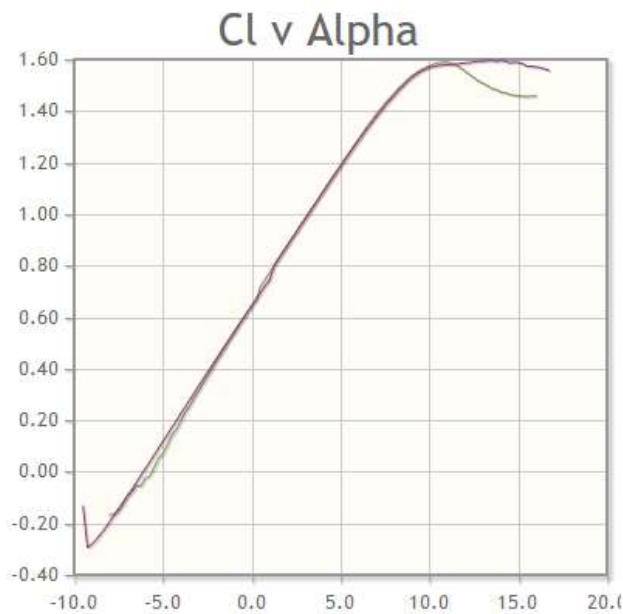


Figura 3.5 – Curva de C_L x α do aerofólio NACA 6412 (Fonte: *AIRFOIL TOOLS*, 2022).

Nota: a curva roxa corresponde a um número de Reynolds de 500.000, enquanto a curva verde representa um número de Reynolds de 200.000. C_L é o coeficiente de sustentação.

Conforme a Figura 3.5, o perfil tem um C_L máximo próximo a 1,6. Ainda é necessário considerar que a asa sofrerá um pequeno decréscimo nesse valor conforme a Figura 3.6. Conforme exposto anteriormente, está sendo considerada uma asa com alongamento de 12,02.

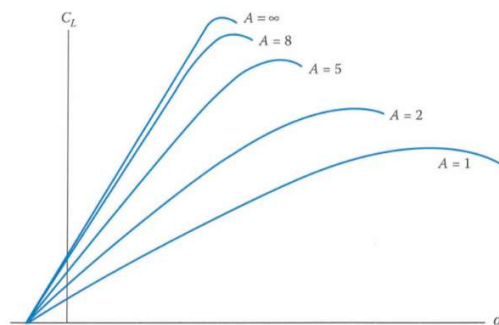


Figura 3.6 – Efeito do alongamento no coeficiente de sustentação (Fonte: Raymer, 2018).

Apesar do valor médio do coeficiente de sustentação durante o cruzeiro ser de 1,35, na configuração de peso máximo de decolagem (MTOW) esse valor é próximo de 1,5, o que também é muito próximo do valor máximo do perfil. Isso indica que em um projeto detalhado pode ser considerado aumentar levemente o arqueamento do perfil, a fim de aumentar a folga entre o C_L máximo e o C_L de cruzeiro.

A Figura 3.5 também pode ser utilizada para definir o ângulo de montagem da asa na fuselagem. Neste caso o valor em torno de 7° , que corresponde ao C_L de cruzeiro, seria o ideal.

Esse parâmetro faz com que a fuselagem fique o mais próximo possível da posição horizontal durante o voo de cruzeiro, contribuindo para reduzir o arrasto.

A asa sem enflexamento do V-BAT 128 está muito adequada ao projeto, pois o envelope de voo é restrito a pequenos números de Mach. O valor do afilamento também está adequado, pois é um valor que aproxima ao carregamento elíptico e não gera grandes complicações para fabricação.

3.4 Considerações sobre o projeto do propulsor

Para o projeto da hélice, deve-se levar em consideração que ela deve apresentar boa eficiência tanto no voo pairado quanto em voo de cruzeiro. A fim de atender essas duas situações, foram utilizados dados de uma hélice de duas pás com torção de 15° em 75% do raio. Esses dados foram obtidos em Hartman e Biermann (1938) e estão na Figura 3.7:

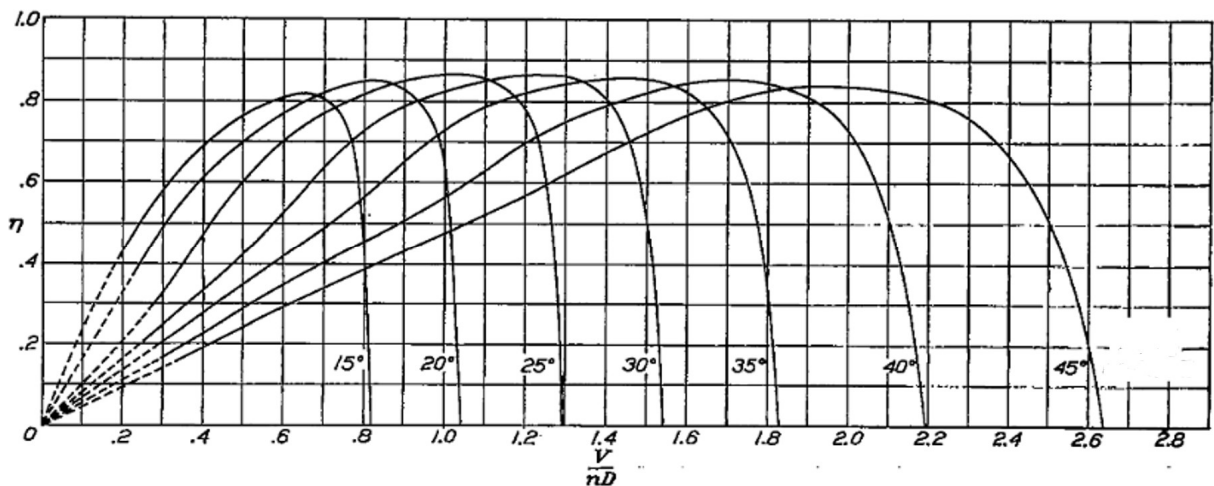


Figura 3.7 – Eficiência em hélices de 2 pás em função da razão de avanço (Fonte: HARTMAN & BIERMANN, 1938).

Nota: As curvas apresentam dados de diferentes hélices, diferenciadas pelo ângulo de torção na posição correspondente a 75% do raio. O ângulo é indicado ao lado de cada curva. A variável do eixo y é a eficiência da hélice. A variável do eixo x é a razão de avanço, onde V é a velocidade do escoamento, n é a velocidade de rotação da hélice em RPM e D é o diâmetro da hélice.

Conforme a figura, a hélice selecionada é a que apresenta maior eficiência nas menores velocidades. Para relacionar os dados de eficiência da hélice com a curva de potência, foi calculada a curva de eficiência em função da velocidade para um diâmetro de 0,711 m e 3 situações de rotação do motor (3500, 4500 e 6500 RPM). O resultado pode ser visto na Figura

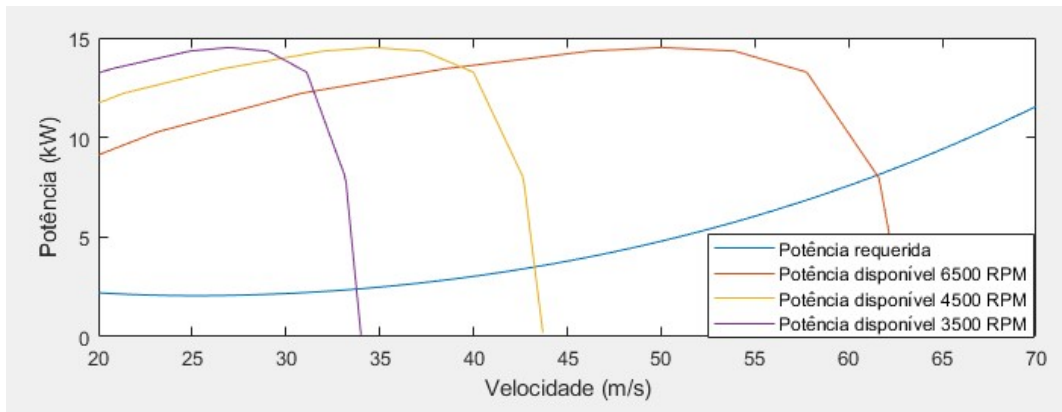


Figura 3.8 – Curvas de potência requerida e disponível.

Neste gráfico é importante observar que de acordo com a Equação 3.1, a velocidade induzida no voo pairado é em torno de 24 m/s e a potência requerida é em torno de 13,3 kw conforme a Tabela 3.1. Como no voo pairado espera-se que o motor esteja em alta rotação, é possível que esta hélice não produza a tração suficiente. O duto gera um acréscimo de tração de 10 a 15% segundo Zamri (2010), isso ajuda, mas ainda não é o suficiente. Tudo indica que a hélice deve ser ajustada em um projeto detalhado para atender a esta condição de voo.

3.5 Diagrama V-n

O diagrama V-n é uma ferramenta que permite visualizar a máxima intensidade da força de sustentação em cada ponto do envelope de voo, tendo grande importância no projeto estrutural da aeronave. O fator de carga representa a razão entre a força de sustentação e o peso da aeronave.

Utilizando os dados calculados até agora é possível calcular a velocidade de estol e plotar o diagrama V-n, que está na Figura 3.9:

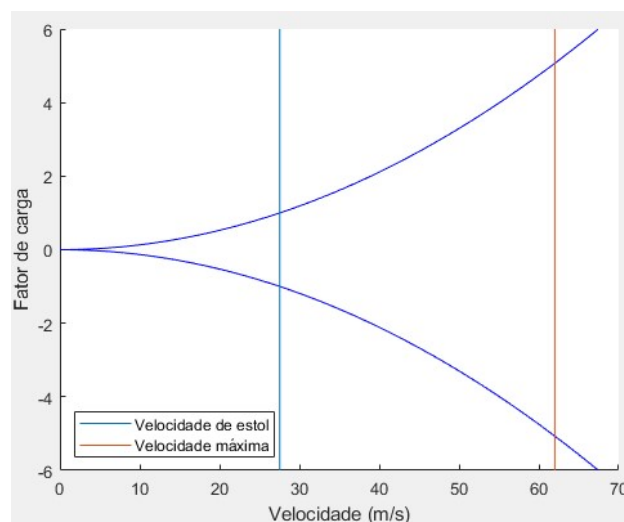


Figura 3.9 – Diagrama V-n.

Sobre este diagrama V-n é importante considerar que a velocidade máxima provavelmente será reduzida em função de ajustes na hélice comentados anteriormente, pois o ganho de desempenho em baixa velocidade certamente irá reduzir o desempenho em velocidades altas. Isso faz sentido considerando que a velocidade máxima do V-BAT 128 é em torno de 46 m/s conforme a entrevista contida em Aero-News Network (2021). Ainda assim é importante considerar que uma velocidade máxima de 62 m/s é possível com uma hélice de passo variável.

O estol deste tipo de aeronave também deve ser estudado com mais detalhes, pois o sistema de comandos de voo não perde autoridade por estar a jusante do propulsor. Além disso, as superfícies de comando podem direcionar o empuxo, amenizando a condição de estol.

CAPÍTULO 4 – METODOLOGIA

4.1 Introdução

Para uma melhor compreensão do comportamento de um *tail-sitter* em voo, foram feitas simulações através da implementação das equações de movimento descritas em Argyle (2016). Além das equações do movimento, a tese ainda fornece uma grande quantidade de dados como as informações de geometria, massa, matriz de inércia e derivadas de estabilidade do protótipo. A implementação completa das equações pode ser observada no código do bloco s-function, contido no Apêndice A, o Apêndice B contém algumas funções criadas para esta simulação.

4.2 Modelagem Computacional

4.2.1 Variáveis de estados e controles

As equações de cinemática calculam as derivadas de cada estado tendo como entrada os estados iniciais e as forças e momentos no sistema de eixos do corpo. Os 13 estados calculados nestas equações são:

- u , v e w : Velocidades lineares no sistema de eixos do corpo;
- p , q e r : Velocidades angulares no sistema de eixos do corpo;
- η_0 , η_1 , η_2 e η_3 : Componentes do quatérnio de atitude;
- p_n e p_e : Posições no eixo norte e leste; e
- h : Altitude.

As variáveis de controle que influenciam o sistema são:

- δ_{ail} : Comando de aileron;
- δ_{elev} : Comando de profundor;
- δ_{rud} : Comando de leme; e
- δ_{thr} : Comando de tração.

Os cálculos de trim que serão descritos posteriormente calculam condições de equilíbrio onde todas as variáveis de estados e controle são conhecidas. Essas condições de equilíbrio são utilizadas como condições iniciais nas simulações dinâmicas.

4.2.2 Ajustes nas equações do grupo motopropulsor

Para o desenvolvimento do modelo do grupo motopropulsor foram utilizadas as Equações 4.1 a 4.6 de Argyle (2016). Essa contribuição consiste na força de tração e no momento de rolagem aplicado pelo motor. Primeiramente calcula-se a velocidade do escoamento na hélice (v_b) pela Equação 4.1.

$$v_b = u - u_w + \frac{1}{2} \omega_p r_p K_{twist} \quad (4.1)$$

Onde u é a velocidade da aeronave no eixo x, u_w é a velocidade do vento no eixo x, ω_p é a velocidade de rotação da hélice em radianos por segundo, r_p é o raio da hélice e K_{twist} é a constante de torção da hélice.

Em seguida para calcular a tração deve-se calcular iterativamente a solução para o sistema composto pelas Equações 4.2, 4.3 e 4.4 que têm como base as teorias da quantidade de movimento e elemento da pá.

$$T = \frac{1 + k_{aug}}{4} (v_b - v_i) \omega_p r_p^2 \rho a_0 b_p c_p \quad (4.2)$$

$$v_f = \sqrt{(v - v_w)^2 + (w - w_w)^2 + (u - u_w + v_i)^2} \quad (4.3)$$

$$v_i = \frac{T}{2\rho\pi r_p^2 v_f} \quad (4.4)$$

Onde T é a tração, v_i é a velocidade induzida na hélice, ρ é a densidade do ar, a_0 é inclinação da curva de sustentação, b_p é o número de pás, c_p é a corda da pá, v_f é a velocidade a jusante e k_{aug} é uma constante de ajuste. Após a solução dessas equações calcula-se a potência induzida (P_{ind}) pela Equação 4.5:

$$P_{ind} = \frac{v_i T}{1 + k_{aug}} \quad (4.5)$$

Consequentemente o momento (M_{ind}) aplicado no eixo x do pode ser calculado pela Equação 4.6:

$$M_{ind} = \frac{P_{ind}}{\omega_p} \quad (4.6)$$

A implementação dessas equações foi testada através dos códigos contidos no Apêndice C e os resultados podem ser comparados com os mostrados em Argyle (2016) na Figura 4.1:

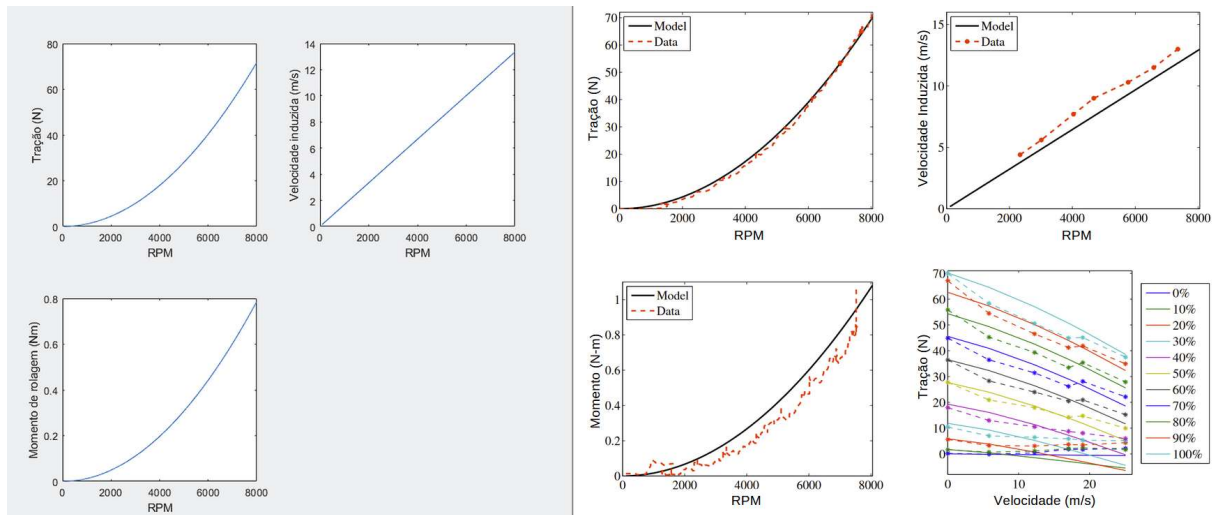


Figura 4.1 – Comparação dos resultados do modelo do motor. (FONTE: Adaptado de: ARGYLE, 2016)

Nota: Os gráficos do lado esquerdo, com curvas em azul, mostram os resultados obtidos em *software* de simulação a partir das equações do modelo matemático, os do lado direito mostram os resultados de Argyle (2016). As curvas tracejadas (*Data*) representam dados experimentais e as curvas pretas (*Model*) representam resultados do modelo matemático.

Apesar da proximidade de resultados, o comportamento do motor em uma simulação com 5 m/s de velocidade mostrou um aumento na tração como pode ser observado na Figura 4.2.

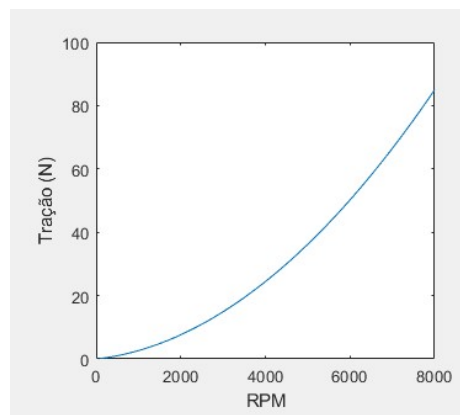


Figura 4.2 – Valores de tração no motor quando com velocidade $u = 5$ m/s

Para contornar esse problema, o valor da tração foi obtido através do gráfico de tração por velocidade mostrado na Figura 4.1, interpolando dados de velocidade (u) e rotação do motor. Apesar dessa interpolação fornecer o valor da tração, ainda é necessário calcular a velocidade induzida, pois esta influencia as forças aerodinâmicas que atuam no duto e nas superfícies de controle. Tendo o valor da tração, a velocidade induzida pode ser calculada por uma solução iterativa das equações 4.3 e 4.4.

A fim de testar como o modelo ajustado do motor se comporta de acordo com a variação da velocidade, foram plotados os gráficos da Figura 4.3, relacionando a tração e a velocidade induzida com a velocidade u . As curvas de cores diferentes representam a variação de rotação do motor.

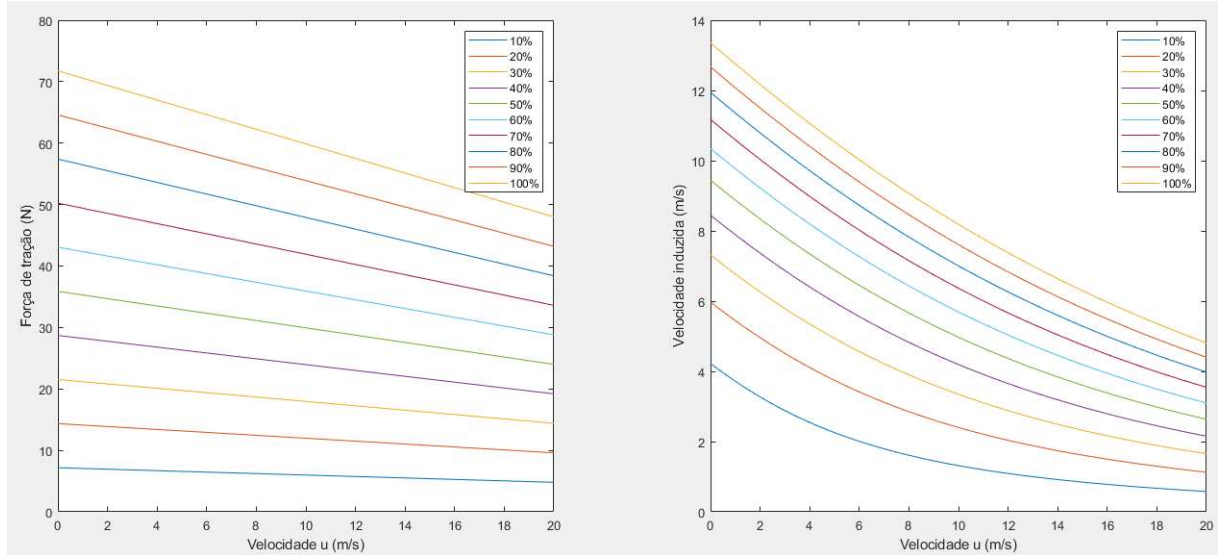


Figura 4.3 – Resultado com as adaptações feitas na modelagem do motor

4.2.3 Ajuste nas equações das superfícies de controle

Na parte de comandos de voo foi feita uma correção. Nas equações de Argyle (2016), o torque gerado no eixo x é em função das forças de arrasto nas superfícies, mas na verdade esse momento é gerado pelas forças de sustentação. A força de arrasto nas superfícies gera apenas uma força no sentido contrário do eixo x do corpo. A correção feita pode ser observada na Equação 4.7.

$$M_{vane} = \begin{bmatrix} -\frac{l_{vane}}{2} \sum_{i=1}^8 F_{L,vi} \\ d_{vane,cp} F_{z,vane} \\ -d_{vane,cp} F_{y,vane} \end{bmatrix} \quad (4.7)$$

Onde $F_{y,vane}$, $F_{z,vane}$ são as resultantes do sistema de controle nos eixos y e z .

4.2.4 Ajustes nas equações do conjunto asa-fuselagem

Dois modelos são propostos para o conjunto asa-fuselagem. Para velocidades abaixo 10 m/s é utilizado o modelo para voo pairado e acima dessa velocidade é utilizado o modelo de voo convencional. As equações foram implementadas conforme o modelo, somente os valores de algumas derivadas de estabilidade foram alterados.

Primeiramente foi alterado o valor de C_{m0} , que foi fornecido como um valor negativo, mas geralmente seu valor é positivo. Possivelmente esse foi um erro de digitação, pois no gráfico da Figura 4.4, é apresentado um resultado de ensaio onde o valor é positivo.

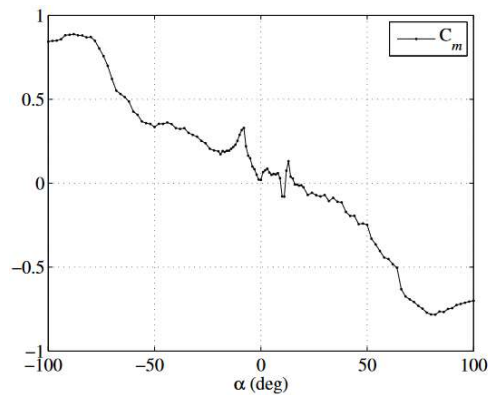


Figura 4.4 – Gráfico de $C_m \times \alpha$ (FONTE: Argyle, 2016)

Nota: Ângulo de ataque (α) em graus

Outro valor que sofreu ajuste foi a derivada de estabilidade $C_{y\beta}$. Esse valor precisa ser negativo, pois representa ação de um vento lateral na força aerodinâmica na direção do eixo y do corpo. Esse problema foi notado ao executar uma simulação dinâmica em uma posição de equilíbrio no voo de cruzeiro, pois foi observada uma instabilidade dinâmica nas velocidades angulares. Os resultados da simulação estão na Figura 4.5:

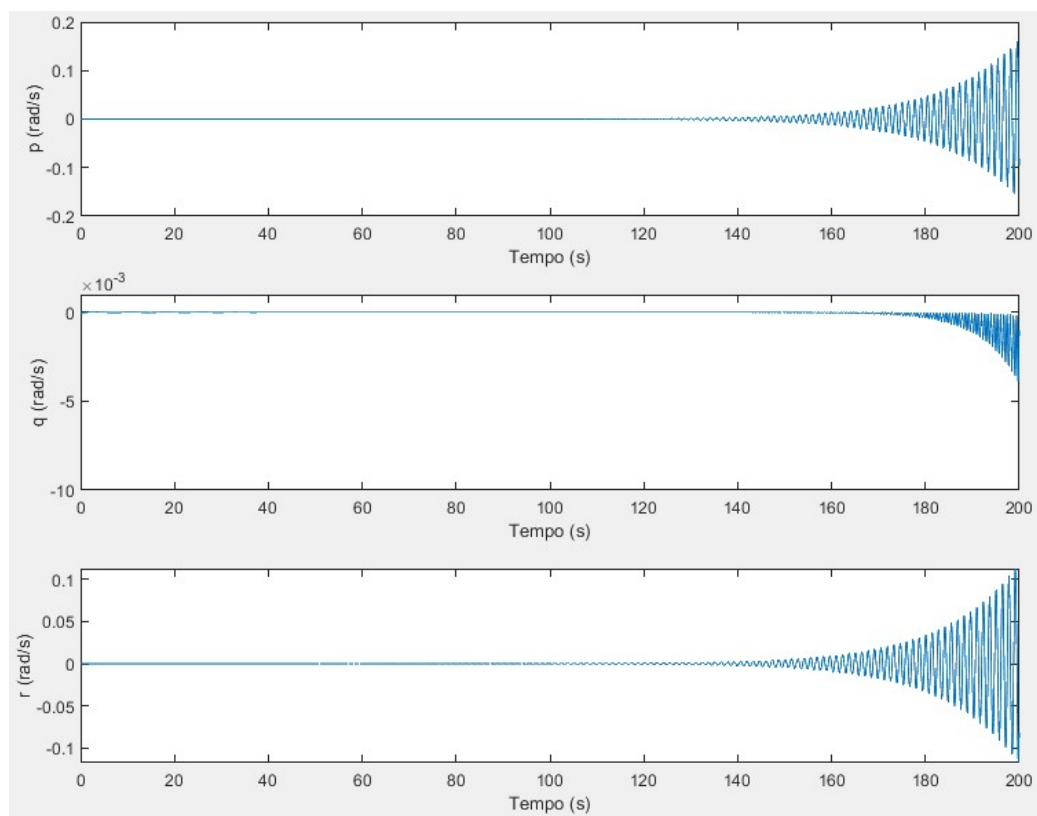


Figura 4.5 – Resultados de simulação com valor de $C_{y\beta}$ positivo

Após a correção de $C_{y\beta}$, essa instabilidade não ocorreu, conforme mostrado na Figura 4.6:

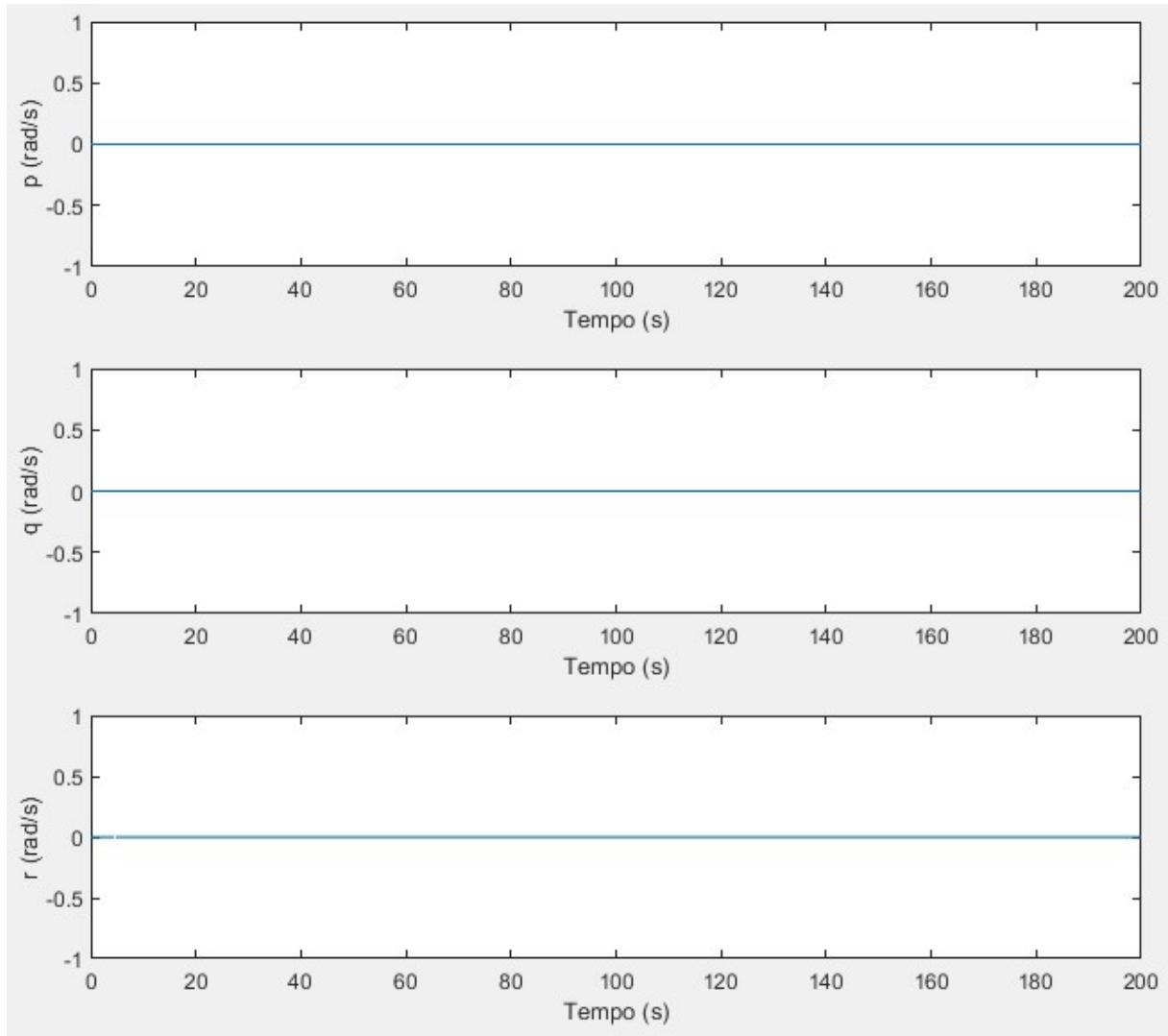


Figura 4.6 – Resultados de simulação com valor de $C_{y\beta}$ negativo

4.3 Cálculos de trim

Esses cálculos consistem em encontrar uma condição de equilíbrio para a aeronave, quando se usa o termo trimado, significa que aeronave está ajustada para se manter na condição de equilíbrio. Todos os cálculos de trim foram feitos por funções de otimização internas de um *software* de simulação. Para cada caso foi definida uma função objetivo (FObj) com a soma dos quadrados dos parâmetros que devem ser anulados.

4.3.1 Trim no voo pairado

Para esta condição foram fixados os seguintes valores indicados na Tabela 4.1:

Tabela 4.1 – Valores fixados no trim pairado.

u	v	w	p	q	r	η_0	η_1	η_2	η_3	p_n	p_e	h
0	0	0	0	0	0	0,7071	0	0,7071	0	0	0	100 m

Estes valores representam o veículo com velocidade zero, nariz apontando para cima e asa direita apontando para leste. As velocidades do vento também foram definidas como zero. A função objetivo escolhida está representada na Equação 4.8:

$$f(\delta_{ail}, \delta_{elev}, \delta_{rud}, \delta_{thr}) = \dot{p}^2 + \dot{q}^2 + \dot{r}^2 + \dot{u}^2 \quad (4.8)$$

Na prática, essa otimização calcula os valores que precisam ser aplicados nos comandos para manter a posição inicial descrita na Tabela 4.1. O código utilizado encontra-se no Apêndice D.

4.3.2 Trim de cruzeiro

Os valores fixados para trim no voo de cruzeiro estão na Tabela 4.2:

Tabela 4.2 – Valores fixados no trim de cruzeiro.

u	v	p	q	r	ψ	p_n	p_e	h
15 m/s	0	0	0	0	0	0	0	100 m

A função objetivo utilizada é a Equação 4.9:

$$f(\delta_{ail}, \delta_{elev}, \delta_{rud}, \delta_{thr}, w, \theta, \phi) = \dot{u}^2 + \dot{v}^2 + \dot{w}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2 + \dot{h}^2 \quad (4.9)$$

Essa função objetivo foi utilizada para gerar uma condição inicial para as simulações dinâmicas. Existe outra forma de calcular posições de equilíbrio que ajuda a analisar o comportamento da aeronave durante as manobras de transição. Esse procedimento consiste em fixar um valor para a velocidade total e na entrada da função objetivo trocar a velocidade no eixo z pelo ângulo de ataque. O ângulo de derrapagem é considerado nulo, então as velocidades nos eixos x e z são calculadas através da velocidade total e do ângulo de ataque. Logo a Equação 4.10 representa a função objetivo modificada:

$$f(\delta_{ail}, \delta_{elev}, \delta_{rud}, \delta_{thr}, \alpha, \theta, \phi) = \dot{u}^2 + \dot{v}^2 + \dot{w}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2 + \dot{h}^2 \quad (4.10)$$

Esses procedimentos de trim podem ser observados com mais detalhes nos códigos contidos nos Apêndices E e F.

4.4 Simulações dinâmicas

As simulações dinâmicas foram realizadas a partir das condições iniciais definidas nas trimagens. Os cálculos dinâmicos foram implementados em um *software* de simulação. O diagrama de blocos da simulação pode ser observado na Figura 4.7:

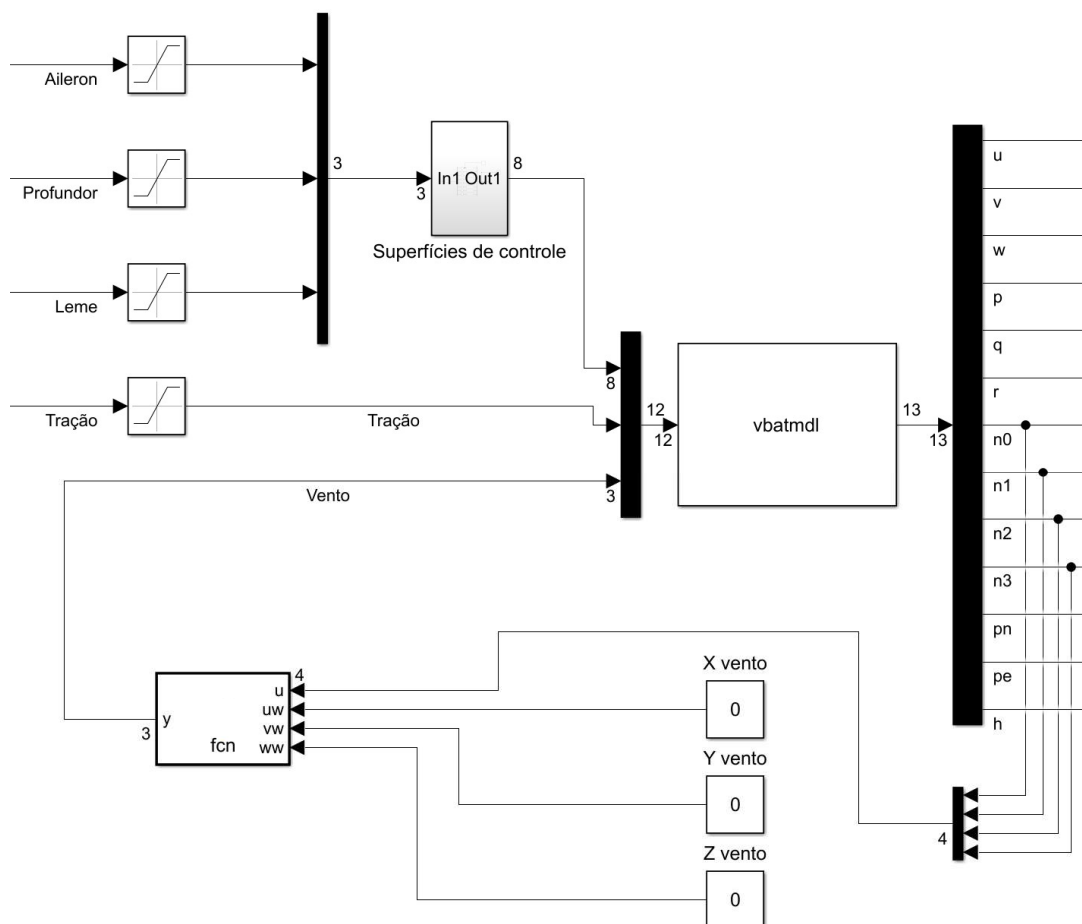


Figura 4.7 – Modelo de simulação

O modelo possui 13 estados, sendo 3 velocidades lineares, 3 velocidades angulares, 4 componentes do quatérnio de atitude e 3 componentes de posição. Os comandos de aileron, profundor e leme são distribuídos entre as superfícies de controle, resultando em 8 comandos, isso permite inserir funções de transferências nos atuadores. A perturbação causada pelo vento também é inserida como variável. O vento é definido no sistema de eixos da terra e passado para o sistema de eixos da aeronave através de um *loop* que verifica o quatérnio atitude e calcula as componentes do vento no sistema de eixos da aeronave através de uma matriz de rotação.

4.5 Controle de atitude em voo pairado

Os resultados de simulação em malha aberta mostraram que esta condição de voo é crítica em relação à estabilidade. Nos trabalhos estudados, diferentes métodos foram implementados para controlar este tipo de aeronave. Johnson e Turbe (2005) utilizaram controle através de redes neurais. Zamri (2010) utilizou lógica nebulosa. Argyle (2016) utilizou controladores PID.

O método escolhido foi implementar controladores PID com base nos ângulos de Euler para a posição vertical descritos em Zamri (2010). Essa sequência de ângulos de Euler tem como ponto inicial a aeronave na posição vertical com a asa direita apontada na direção leste. Essa sequência de rotações é demonstrada na Figura 4.8:

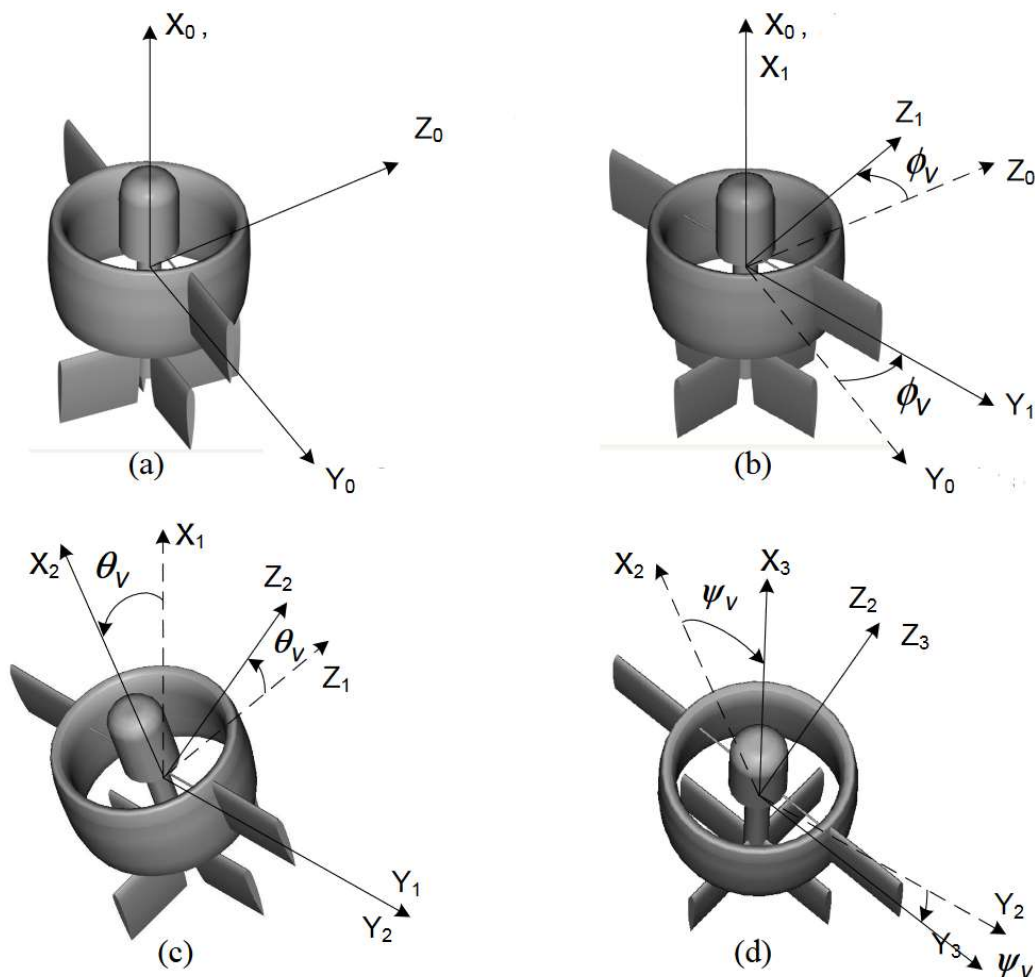


Figura 4.8 – Ângulos de Euler na vertical (Fonte: ZAMRI, 2010).

O cálculo desses ângulos é feito através de uma conversão de quatérnios para ângulos de Euler e em seguida uma conversão através das Equações 4.11, 4.12 e 4.13:

$$\phi_v = -\tan^{-1} \left(\frac{\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi}{\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi} \right) \quad (4.11)$$

$$\theta_v = \sin^{-1}(-\cos\phi \cos\theta) \quad (4.12)$$

$$\psi_v = -\tan^{-1} \left(\frac{\sin\phi \cos\theta}{-\sin\theta} \right) \quad (4.13)$$

O diagrama de blocos deste controlador pode ser observado na Figura 4.9:

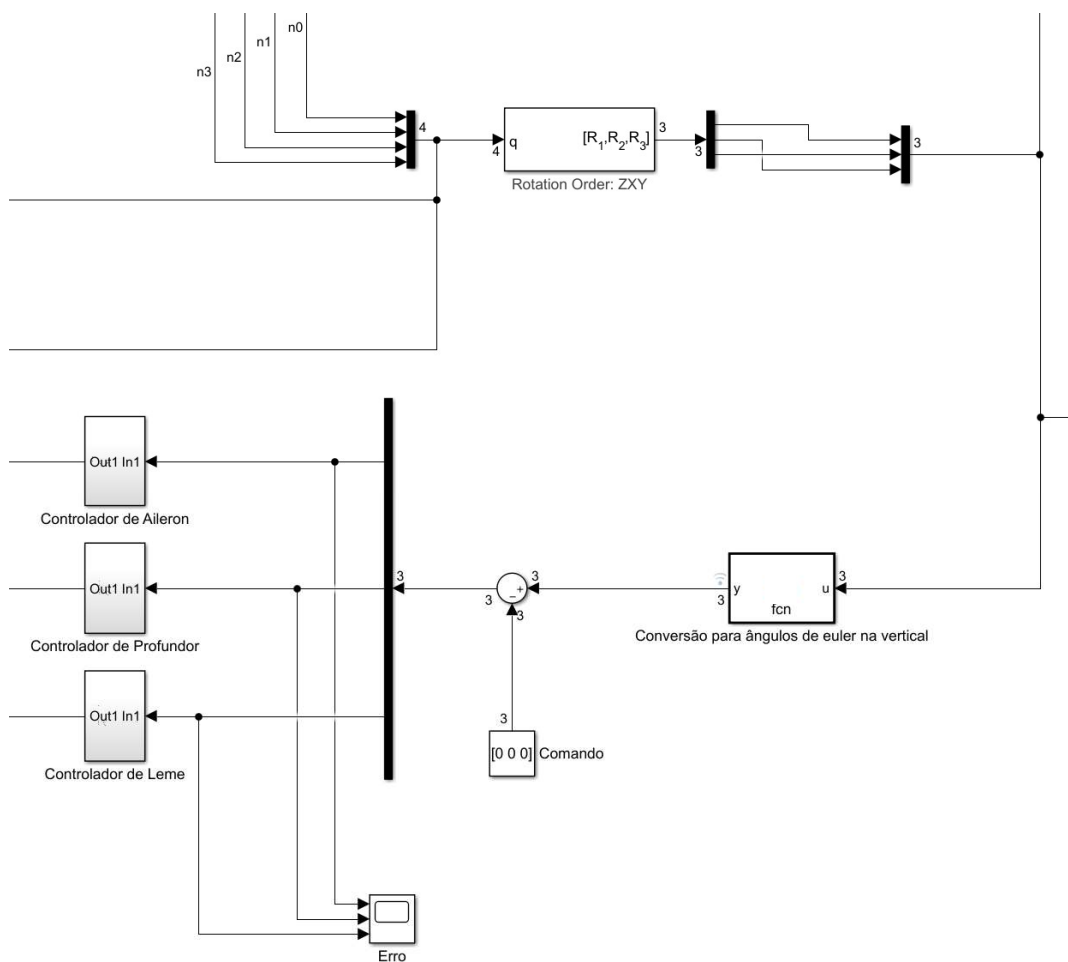


Figura 4.9 – Controlador de atitude por ângulos de Euler na vertical.

CAPÍTULO 5 – RESULTADOS E DISCUSSÃO

5.1 Resultados de trim

5.1.1 Voo pairado

Os resultados dos comandos para estabilizar o veículo em voo pairado estão na Tabela 5.1:

Tabela 5.1 – Condições de equilíbrio no voo pairado.

δ_{ail}	-0,5934
δ_{elev}	$1,1064 \times 10^{-16}$
δ_{rud}	$5,9389 \times 10^{-17}$
δ_{thr}	0,8330

Conforme esperado, os valores de profundor e leme foram muito próximos de zero, o comando de aileron gera um momento oposto ao torque aplicado pelo motor e a tração equilibra o peso do veículo e o arrasto nas superfícies de comando. Como pode ser observado no código do Apêndice D, foi feita uma verificação através de uma função que calcula as derivadas dos 13 estados nas condições calculadas. O resultado foi muito próximo de zero para todas as derivadas, conforme esperado.

5.1.2 Voo de cruzeiro

O ponto inicial selecionado para as simulações dinâmicas descrito nas Seção 4.3.2 gerou os resultados da Tabela 5.2.

Tabela 5.2 – Condições de equilíbrio no voo de cruzeiro.

δ_{ail}	-0,0244
δ_{elev}	-0,0611
δ_{rud}	$2,2657 \times 10^{-11}$
δ_{thr}	0,1394
w	0,1890
θ	0,0126
ϕ	$-8,1773 \times 10^{-11}$

Assim como no trim pairado, os valores foram testados em uma função que faz os cálculos de dinâmica e os resultados estavam de acordo com o esperado, o código pode ser observado no Apêndice E.

5.1.3 Análise do trim na manobra de transição

Nessa análise foram feitos cálculos de trim das velocidades de 0 até 20 m/s, esses resultados fornecem um melhor entendimento sobre a manobra e permite uma comparação com os resultados de Argyle (2016), que estão na Figura 5.1:

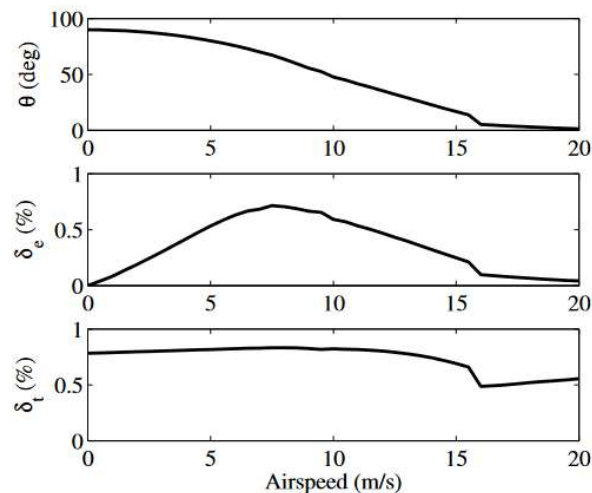


Figura 5.1 – Dados de trim na manobra de transição (Fonte: ARGYLE, 2016).

Nota: θ em graus (*deg*) e velocidade (*Airspeed*) em m/s

Para velocidade nula, os valores encontrados foram bem semelhantes. Entretanto, para velocidades de 15 a 20 m/s, que correspondem ao voo convencional, os valores do comando de tração estão muito altos em relação ao que é esperado. Conforme os resultados obtidos nos capítulos 3 e 4, a tração necessária para o voo convencional costuma ser bem menor. Isso pode ser observado inclusive nos cálculos feitos por Zamri (2010) que estimou uma potência necessária para o voo pairado de seu protótipo era de 412 W, enquanto em um voo de cruzeiro a 30 m/s a potência requerida seria de 154,9 W.

Os resultados obtidos para comparação com a Figura 5.1 estão nas Tabelas 5.3 a 5.6:

Tabela 5.3 – Condições de equilíbrio para velocidades de 0 a 5 m/s.

V_a (m/s)	0	1	2	3	4	5
δ_{ail}	-0,5935	-0,5923	-0,8871	-0,6216	-0,4912	-0,4085
δ_{elev}	0,0000	0,2785	1,0000	1,0000	1,0000	1,0000
δ_{rud}	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
δ_{thr}	0,8330	0,8329	0,8480	0,9015	0,9694	1,0000
α (°)	2,4621	88,3524	82,2178	66,0648	52,6848	44,2057
θ (°)	89,9975	88,3525	83,2978	73,6835	60,7774	44,4058
ϕ (°)	0,0000	-0,0000	-0,0000	-0,0049	-0,0056	0,0058
FObj	0,0000	0,0000	0,0071	0,4521	0,7601	0,6461

Tabela 5.4 – Condições de equilíbrio para velocidade de 6 a 10 m/s.

V_a (m/s)	6	7	8	9	10
δ_{ail}	-0,2343	-0,1941	-0,0810	0,0000	-0,0499
δ_{elev}	0,7657	0,5367	0,3019	0,0000	0,3044
δ_{rud}	0,0000	0,0000	0,0000	0,0000	-0,0173
δ_{thr}	0,9998	1,0000	1,0000	1,0000	0,1392
α (°)	35,0705	22,8351	15,2583	7,6554	7,5
θ (°)	24,3172	0,5263	-9,0000	-8,9985	7,8
ϕ (°)	0,0000	0,0039	-0,0100	0,0000	-0,5898
FObj	2,4311	10,8491	38,4701	118,8706	1,1545

Tabela 5.5 – Condições de equilíbrio para velocidades de 11 a 15 m/s.

V_a (m/s)	11	12	13	14	15
δ_{ail}	-0,0591	-0,0430	-0,0322	-0,0254	-0,0243
δ_{elev}	0,3468	0,2146	0,1042	0,0126	-0,0611
δ_{rud}	0,0000	0,0000	0,0000	0,0000	0,0000
δ_{thr}	0,1307	0,1187	0,1114	0,1106	0,1393
α (°)	6,6368	4,6058	3,0139	1,7466	0,7231
θ (°)	6,6368	4,6058	3,0139	1,7466	0,7231
ϕ (°)	0,0000	0,0000	0,0000	0,0000	0,0000
FObj	0,0000	0,0000	0,0000	0,0000	0,0000

Tabela 5.6 – Condições de equilíbrio para velocidades de 16 a 20 m/s.

V_a (m/s)	16	17	18	19	20
δ_{ail}	-0,0245	-0,0245	-0,0240	-0,0234	-0,0226
δ_{elev}	-0,1196	-0,1669	-0,2057	-0,2379	-0,2650
δ_{rud}	0,0000	0,0000	0,0000	0,0000	0,0000
δ_{thr}	0,1833	0,2341	0,2918	0,3567	0,4291
α (°)	-0,1145	-0,8080	-1,3886	-1,8793	-2,2976
θ (°)	-0,1145	-0,8080	-1,3886	-1,8793	-2,2976
ϕ (°)	0,0000	0,0000	0,0000	0,0000	0,0000
FObj	0,0000	0,0000	0,0000	0,0000	0,0000

Primeiramente é interessante observar que os resultados para velocidade 0 foram muito próximo aos valores encontrados no trim do voo pairado. Nesse caso ainda é importante destacar que o valor do ângulo de ataque não influencia o modelo por causa da velocidade nula.

Os resultados indicam que o veículo não consegue apresentar uma posição de equilíbrio em todas as velocidades testadas. Os valores destacados em vermelho mostram os pontos de teste onde não foi possível aproximar o valor da função objetivo a zero. A impossibilidade de calcular o trim possivelmente representa uma situação de estol. Nesse caso é importante destacar que os efeitos do estol estão considerados, pois os coeficientes aerodinâmicos foram modelados para uma grande faixa de ângulos de ataque.

Na velocidade de 1 m/s pode-se afirmar que o veículo está fazendo a manobra de voo inclinado de baixa velocidade descrita em Zamri (2010). A partir dos 11 m/s o comportamento é totalmente dentro do esperado para uma aeronave em voo reto e nivelado. Os ângulos de ataque e atitude são iguais e diminuem com o aumento da velocidade. O comportamento do comando de tração aumenta juntamente com a velocidade conforme esperado.

Na Figura 5.2 estão representados os resultados experimentais com o protótipo de Argyle (2016). Esses resultados mostram uma pequena perda de altitude durante as manobras de transição, isso é mais um indício de que por alguns instantes o protótipo se encontra em estol durante essas manobras.

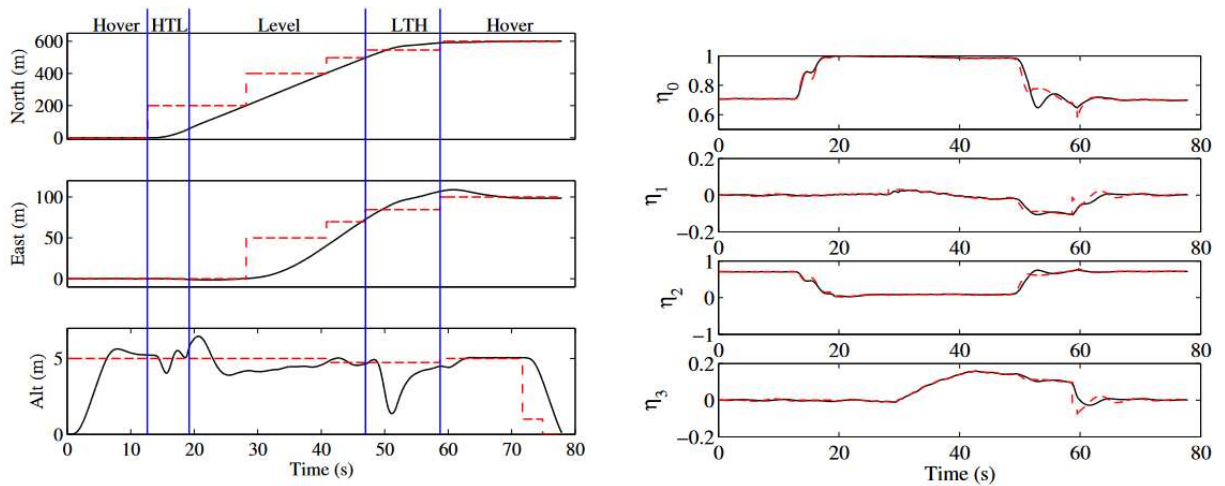


Figura 5.2 – Dados obtidos em voo com o protótipo do V-BAT (Fonte: ARGYLE, 2016).

Nota: Os gráficos da esquerda mostram posição norte (*north*), leste (*east*) e altitude em metros ao longo do tempo em segundos. Os gráficos da direita mostram a variação dos componentes do quatérnio ao longo do tempo em segundos. *Hover* é voo pairado e *Level* é voo convencional. HTL e LTH são as fases de transição.

5.2 Testes aplicados no modelo

5.2.1 Testes de comandos

Nos testes foram inseridas perturbações em cada comando, a condição inicial calculada para o voo pairado foi utilizada neste teste. As forças e momentos causados por cada perturbação estão na Tabela 5.7:

Tabela 5.7 – Forças e momentos gerados por atuação nos comandos.

	F_x	F_y	F_z	M_x	M_y	M_z
$\Delta\delta_{ail} = +0,1$	1,0754	0,0000	0,0000	0,1215	0,0000	0,0000
$\Delta\delta_{ail} = -0,1$	-1,0754	0,0000	0,0000	-0,1215	0,0000	0,0000
$\Delta\delta_{elev} = +0,1$	0,0000	0,0000	0,9781	0,0000	0,6944	0,0000
$\Delta\delta_{elev} = -0,1$	0,0000	0,0000	-0,9781	0,0000	-0,6944	0,0000
$\Delta\delta_{rud} = +0,1$	0,0000	0,9781	0,0000	0,0000	0,0000	-0,6944
$\Delta\delta_{rud} = -0,1$	0,0000	-0,9781	0,0000	0,0000	0,0000	0,6944
$\Delta\delta_{thr} = +0,1$	6,4100	0,0000	0,0000	-0,0445	0,0000	0,0000
$\Delta\delta_{thr} = -0,1$	-6,4099	0,0000	0,0000	0,0419	0,0000	0,0000

Os comandos de aileron, profundor e leme geram momentos nos eixos esperados. Ainda é importante observar que para o aileron e para o profundor, o momento gerado é do mesmo sinal do comando, enquanto no leme o sinal é oposto. O acoplamento entre os comandos de aileron e tração também pode ser observado. O comando de aileron muda o arrasto total gerado

pelas superfícies de controle, enquanto o comando de tração influencia o torque gerado pelo motor.

5.2.2 Testes aerodinâmicos

Nesses testes foram inseridos valores para o vento em cada eixo. Os testes foram aplicados nas condições iniciais de voo pairado e voo convencional. Os resultados podem ser observados respectivamente nas Tabelas 5.8 e 5.9:

Tabela 5.8 – Forças e momentos gerados por ventos no voo pairado.

	F_x	F_y	F_z	M_x	M_y	M_z
$u_w = 1 \text{ m/s}$	-2,0079	0,0000	-0,2039	0,0797	0,0061	0,0000
$u_w = -1 \text{ m/s}$	3,0420	0,0000	-0,2039	-0,0951	0,0061	0,0000
$v_w = 1 \text{ m/s}$	-0,3087	2,6357	0,0000	0,0012	0,0061	-1,9018
$v_w = -1 \text{ m/s}$	-0,2861	-2,6357	0,0000	0,0012	0,0061	1,9018
$w_w = 1 \text{ m/s}$	-0,0086	0,0000	4,2643	0,0012	1,9441	0,1164
$w_w = -1 \text{ m/s}$	0,0367	0,0000	-4,2643	0,0012	-1,9319	0,0000

Tabela 5.9 – Forças e momentos gerados por ventos no voo de cruzeiro.

	F_x	F_y	F_z	M_x	M_y	M_z
$u_w = 1 \text{ m/s}$	1,0787	0,0000	6,4509	0,0111	-0,0592	0,0000
$u_w = -1 \text{ m/s}$	-1,1451	0,0000	-6,8670	-0,0109	0,0655	0,0000
$v_w = 1 \text{ m/s}$	-0,0259	2,6357	-0,2294	1,9605	0,0056	-2,9945
$v_w = -1 \text{ m/s}$	-0,0259	-2,6873	-0,2294	-1,9605	0,0056	2,9945
$w_w = 1 \text{ m/s}$	-3,4115	0,0000	34,2252	0,0000	3,3443	0,0000
$w_w = -1 \text{ m/s}$	5,1530	0,0000	-34,4220	-0,0001	-3,3344	0,0000

Em geral as perturbações aerodinâmicas estão gerando forças e momentos nos eixos esperados. Somente para o vento igual a 1 m/s no eixo z surgiu um momento no eixo z, ao investigar esse caso foi verificado que o ângulo de derrapagem no duto é igual a π radianos e isso gera um pequeno valor ao calcular o seno deste ângulo em função da quantidade de casas decimais consideradas pelo *software* de simulação.

5.3 Simulações em malha aberta

5.3.1 Voo pairado

Nessa simulação foi observado que o modelo sai da posição de equilíbrio em torno dos 10 segundos de simulação. Isso acontece em função da configuração instável da aeronave e do

método numérico utilizado para calcular o trim. No cálculo do trim, as forças e momentos são aproximadas a zero, mas não chegam exatamente em zero, e isso faz com que a posição de equilíbrio seja perdida com o tempo.

A atitude em ângulos de Euler ao longo do tempo durante a simulação pode ser observada na Figura 5.3:

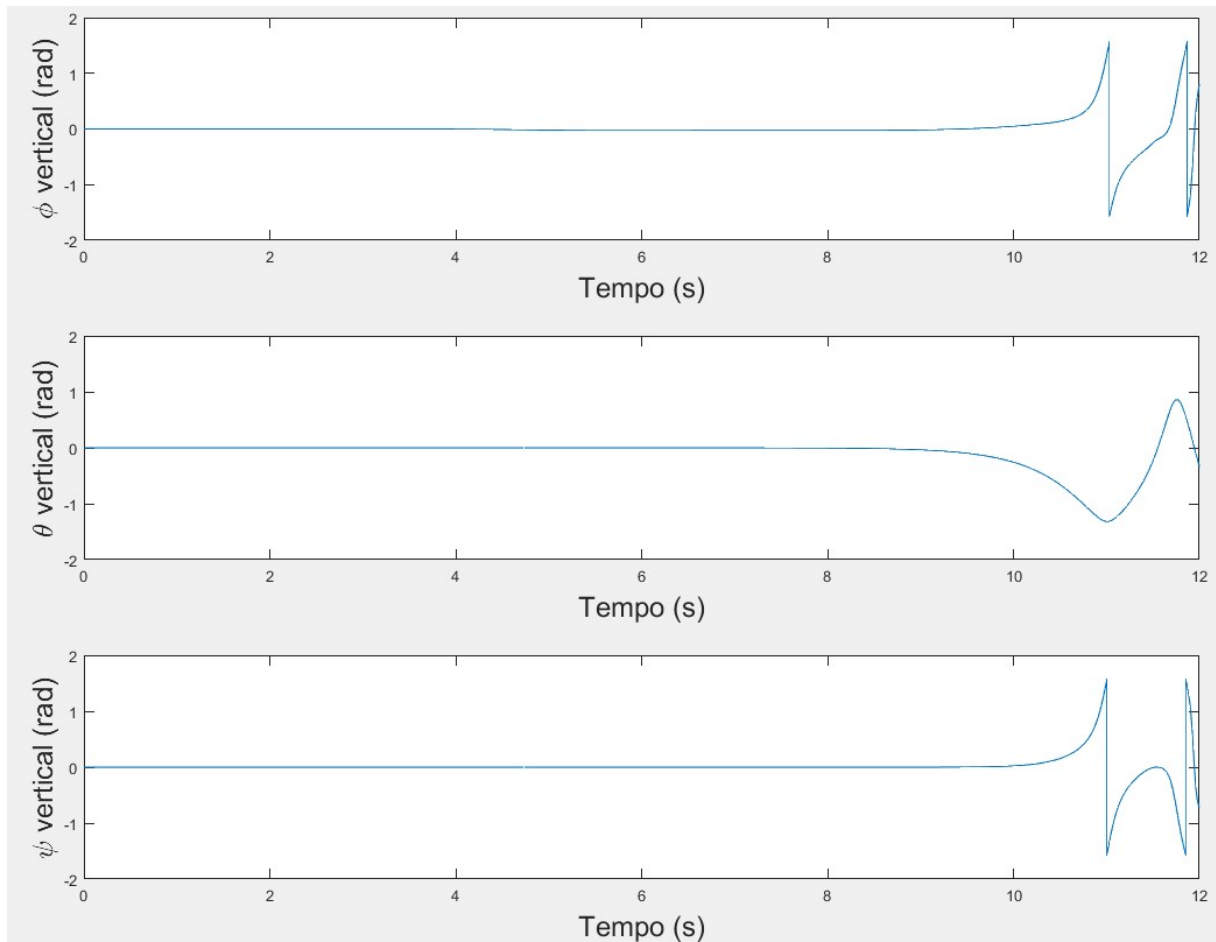


Figura 5.3 – Ângulos de atitude durante simulação de voo pairado sem controlador.

5.3.2 Voo reto e nivelado

Nesta simulação a aeronave manteve sua posição de equilíbrio durante todo o tempo simulado. Esse resultado está na Figura 5.4:

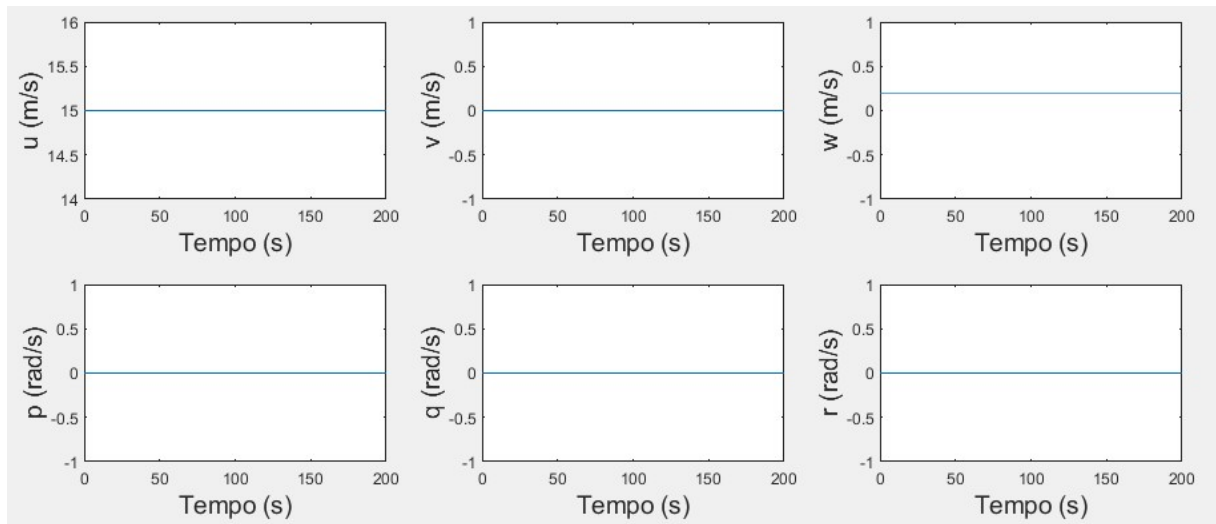


Figura 5.4 – Simulação de voo reto e nivelado.

Nota: Tempo em segundos.

5.3.3 Resposta aos comandos latero-direcionais

Nesta simulação o valor trimado para o comando de aileron sofreu um degrau de 0,01. Resultando nos gráficos da Figura 5.5:

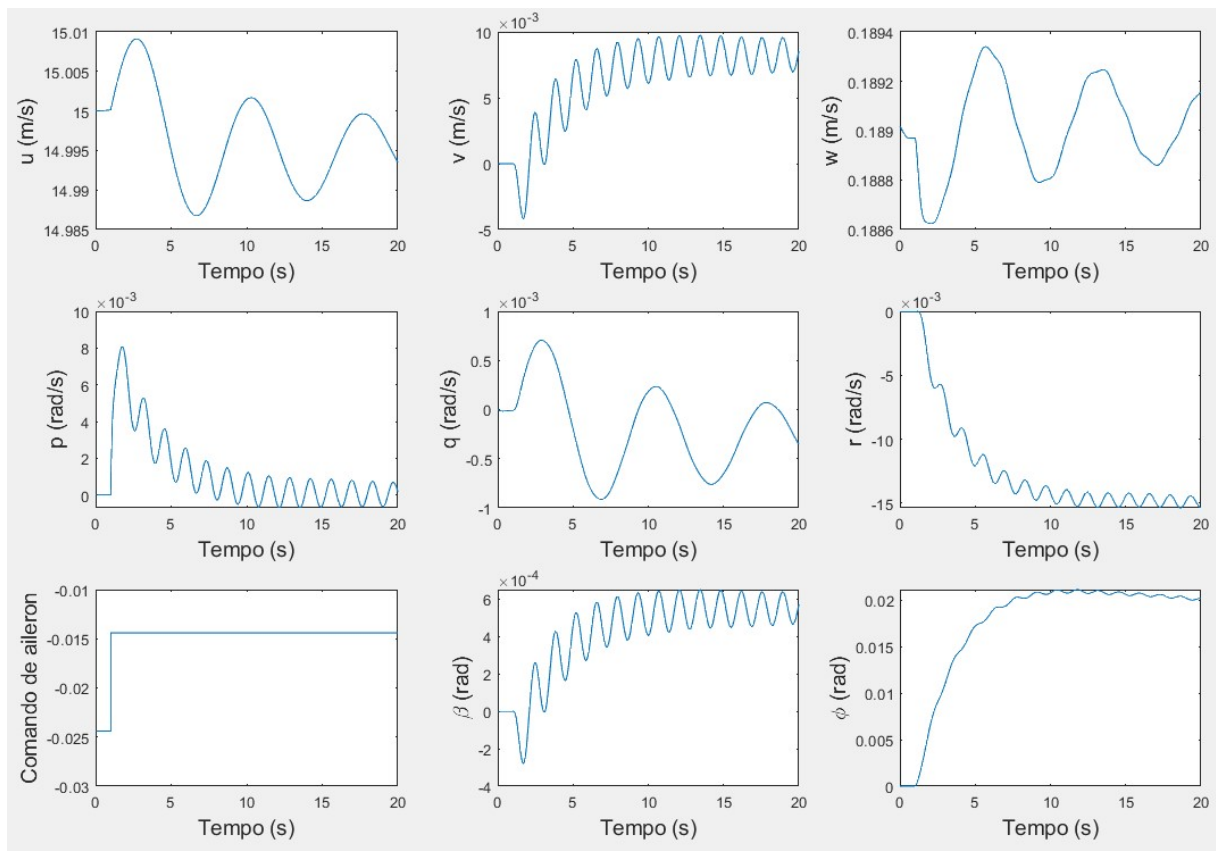


Figura 5.5 – Resposta a um comando de aileron.

A resposta positiva na taxa de rolamento era esperada, entretanto foi observada uma forte guinada adversa. O que também pode ser observado na plotagem da vista superior do trajeto, que pode ser visto na Figura 5.6:

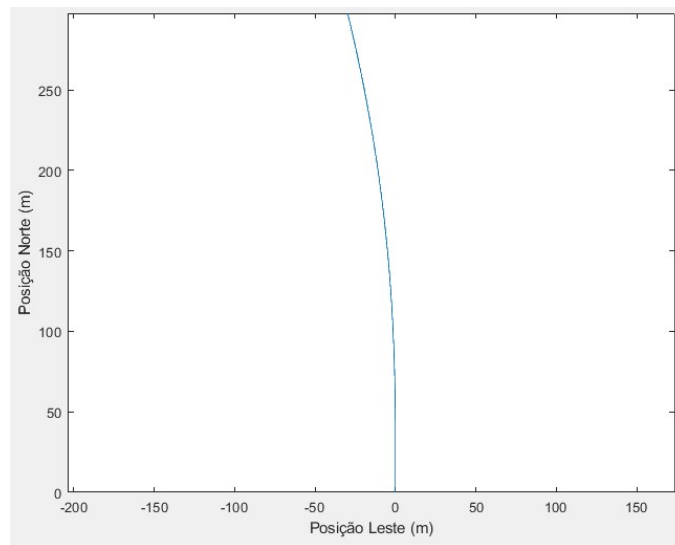


Figura 5.6 – Vista superior da simulação com comando de aileron.

Nota: A aeronave parte da origem com a asa direita voltada para o leste e proa para o norte.

Ao examinar mais detalhadamente foi verificado que o momento responsável pela guinada vem das forças aerodinâmicas do duto. Ao simular uma curva com o leme, a guinada também ocorreu para o lado contrário em função das forças do duto. Possivelmente este modelo não é muito adequado para manobras com comandos latero-direcionais no voo de cruzeiro, coincidentemente, Argyle (2016) utilizou um modelo alternativo em seu capítulo sobre controle nesta condição de voo.

5.3.4 Resposta ao comando de profundor

Nesta simulação o valor trimado para o comando de profundor foi acrescido de 0,01. Resultando nos gráficos da Figura 5.7:

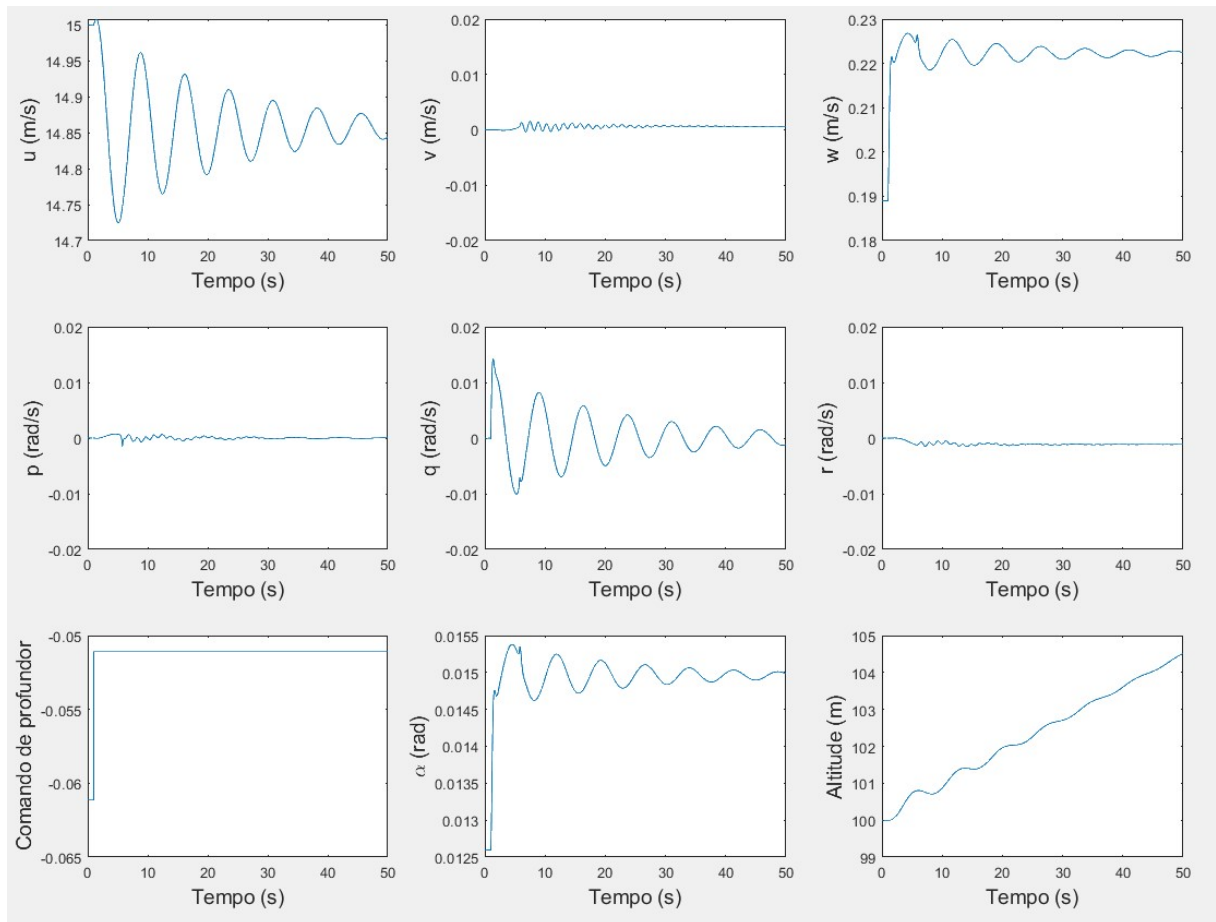


Figura 5.7 – Resposta a um comando de profundor.

Conforme esperado, a taxa de arfagem foi positiva inicialmente, seguida por oscilações amortecidas, indicando em efeito de estabilidade longitudinal. As oscilações e desequilíbrios nas taxas latero-direcionais ocorrem por divergência da posição de equilíbrio inicial.

5.4 Simulações com controlador de atitude em voo pairado

Nessa simulação o objetivo foi manter a posição inicial definida para o voo pairado. O resultado apresentou uma melhora em relação à simulação sem controlador, mas surgiram oscilações crescentes ao longo do tempo. Os valores de erro de para cada ângulo de Euler estão na Figura 5.8:

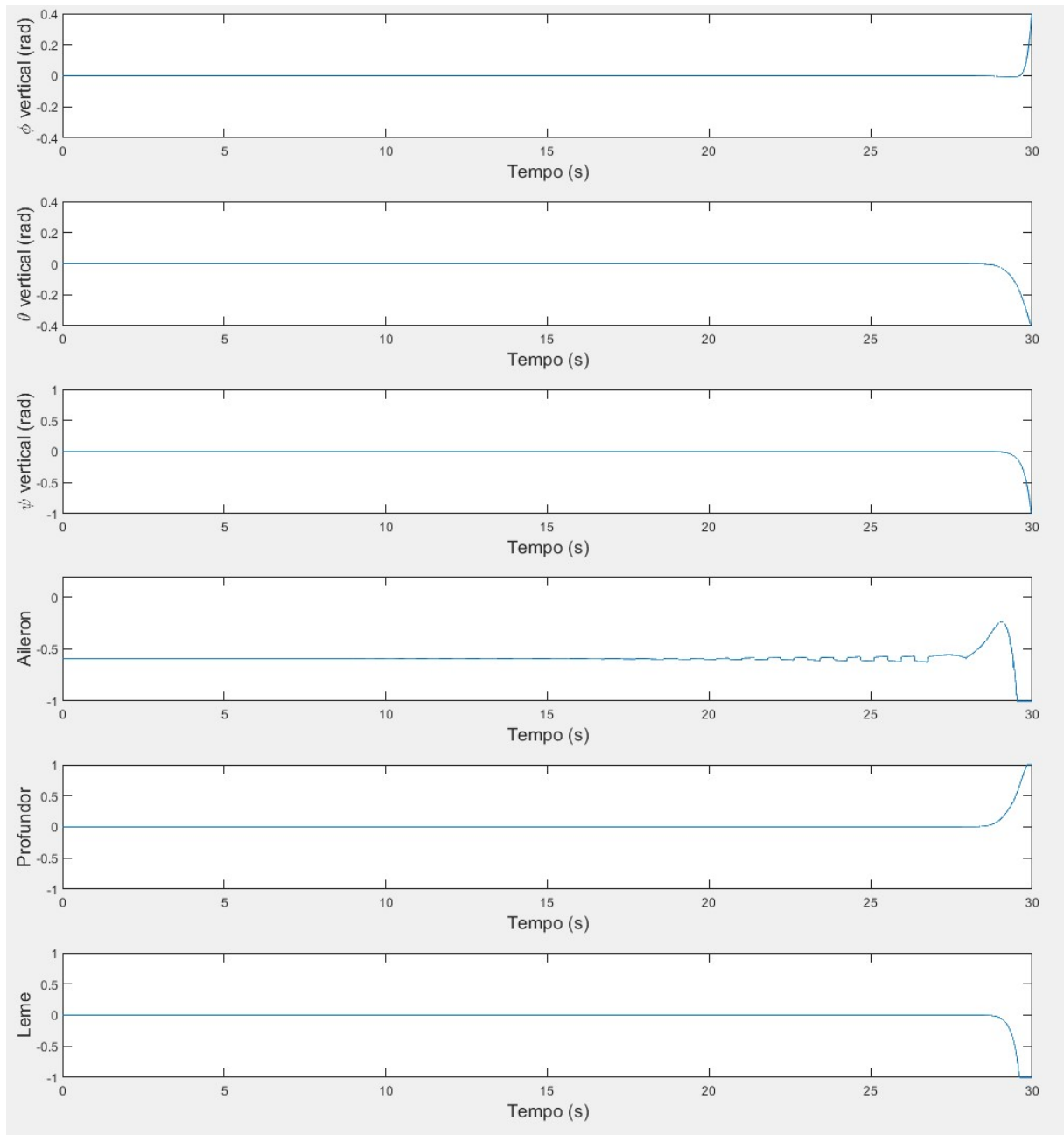


Figura 5.8 – Simulação em voo pairado com controlador de atitude.

A Figura 5.9 mostra os resultados da mesma simulação, mas com uma escala diferente para que seja possível observar oscilações de pequenas amplitudes:

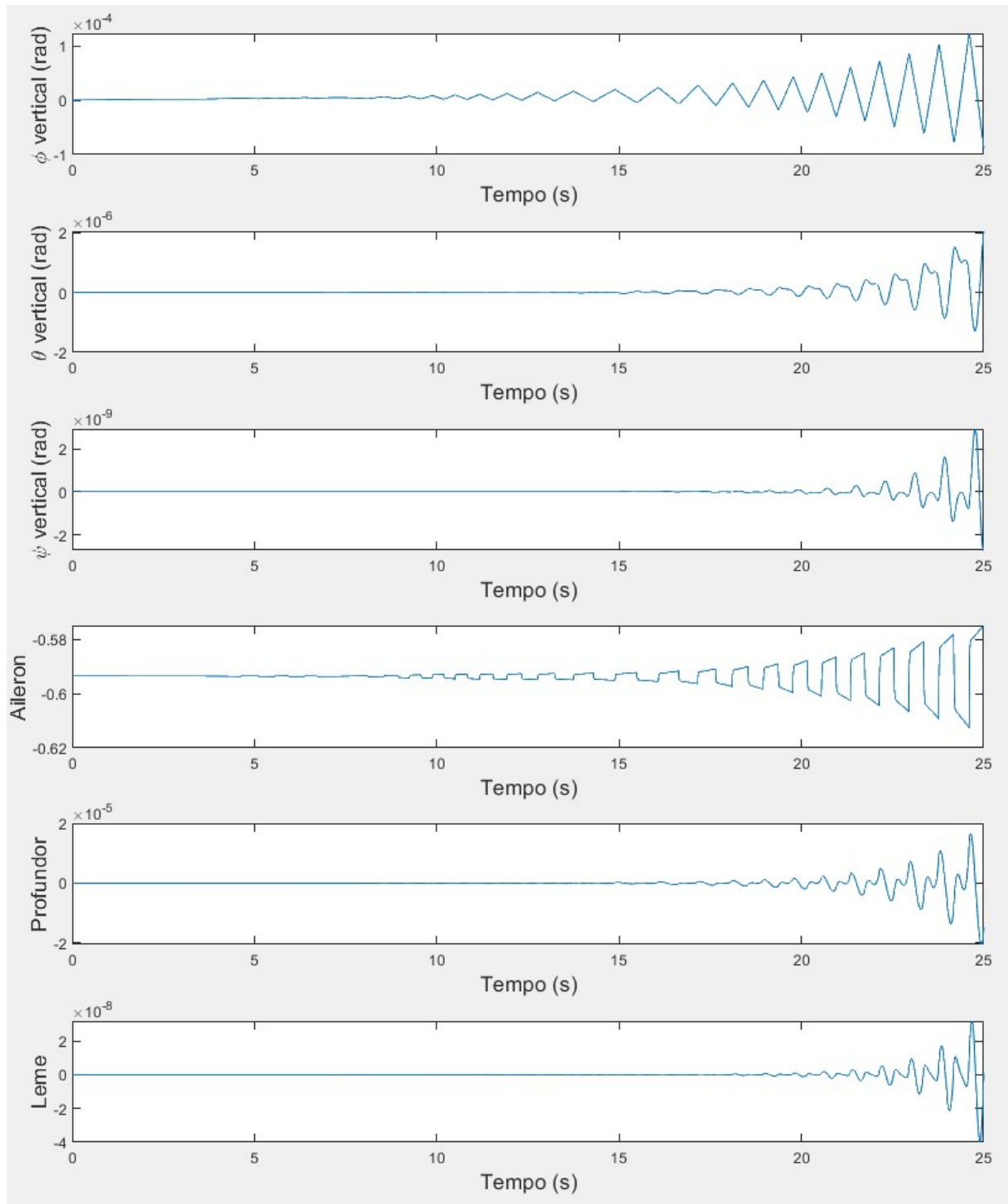


Figura 5.9 – Oscilações na simulação em voo pairado com controlador de atitude.

É importante destacar que antes da implementação dos controladores a estabilidade era perdida antes dos 12 segundos conforme demonstrado na Figura 5.3. Nessas simulações, os controladores atuaram conforme a Equação 5.1, utilizando os ganhos (K) da Tabela 5.10.

$$\Delta\delta = K_p e + K_d \dot{e} + K_i \int_{t_0}^t e \, d\tau \quad (5.1)$$

Tabela 5.10 – Ganhos dos controladores.

	Proporcional	Derivativo	Integrativo
Aileron	30	30	1
Profundor	1	1	0,02
Leme	1	1	0,02

Conforme a Figura 4.9, os erros (e) são calculados em ângulos de Euler na vertical e podem ser representados pelas Equações 5.2 a 5.5:

$$e_{\phi_v} = \phi_v - \phi_{v_c} \quad (5.2)$$

$$e_{\theta_v} = \theta_v - \theta_{v_c} \quad (5.3)$$

$$e_{\psi_v} = \psi_v - \psi_{v_c} \quad (5.4)$$

O erro ϕ_v controla o comando de aileron, θ_v controla o comando de profundor e ψ_v controla o comando de leme. Na simulação realizada, o comando para todos os ângulos de atitude foi zero.

5.5 Comparações de desempenho

Tabela 5.11 – Tabela de comparação de desempenho.

	V-BAT 128	ScanEagle	FT-100
Envergadura	3 m	3,1 m	2,7 m
Peso máximo de decolagem	56,7 kg	26,5 kg	7 kg
Carga paga	11,3 kg	5 kg	3 kg
Autonomia	10 h	18 h	2h
Velocidade máxima	46,3 m/s	41,2 m/s	17 m/s

Fonte: SHIELD AI, 2022; INSITU, 2022; FT SISTEMAS, 2022.

A Tabela 5.11 mostra um comparativo de desempenho do V-BAT 128 com outros modelos de porte semelhantes já utilizados na MB. É nítido que o V-BAT 128 se destaca no quesito de carga paga (quantidade de carga que pode ser transportada além do peso máximo de combustível), mas é bem inferior ao ScanEagle em autonomia. Entretanto, é importante considerar que quando se deseja informações em tempo real, o alcance não será limitado pelo desempenho de cada modelo, e sim pela capacidade de transmitir as informações obtidas.

Do ponto de vista de dimensionamento para o voo convencional, o projeto do V-BAT 128 se mostrou muito coerente com os conceitos apresentados em Raymer (2018). Mas o projeto de um *tail-sitter* deve ser norteador por uma análise de viabilidade do voo pairado, através da teoria da quantidade de movimento. Essas análises foram muito interessantes, pois indicaram que a implementação de uma hélice de passo variável poderia melhorar muito o desempenho em velocidade máxima.

CAPÍTULO 6 – CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

6.1 Conclusões

Neste trabalho foi feita a modelagem e análises de desempenho, estabilidade e controle de um veículo tipo *tail-sitter*.

Os resultados obtidos mostram que a grande dificuldade do projeto está na implementação de leis de controle durante o voo pairado. A modelagem computacional apresentada em Argyle (2016) se mostrou adequada para estudar esse problema, mas se trata de um protótipo em escala. De forma que o ideal seria alcançar um domínio das leis de controle nesse modelo e posteriormente ajustar os dados do modelo para aeronaves de outras dimensões.

Para a implementação de controladores PID, é necessário utilizar técnicas avançadas para determinação dos ganhos corretos, ainda assim, os ganhos adequados podem variar entre as condições de voo segundo Argyle (2016). Os métodos de controle por redes neurais e lógica nebulosa implementados por Johnson e Turbe (2005) e Zamri (2010) respectivamente, também podem ser implementados futuramente nesse modelo para comparação.

Além da necessidade de estabilizar a aeronave na condição de voo pairado, também é necessário que a aeronave tenha a capacidade de se estabilizar com pequenas inclinações, de forma que permita manobras para pequenos ajustes na manobra de pouso. O domínio de uma lei de controle que consiga realizar essas manobras, juntamente com a capacidade de ajustar o modelo para outras dimensões e massas, resultarão na capacidade de realizar simulações de voo com perturbações de vento. Isso geraria dados fundamentais para definir os limites de operação na presença de ventos, o que é fundamental para tomadas de decisão com relação ao emprego deste tipo de aeronave em um cenário marítimo.

Apesar das dificuldades na implementação de controle, a confecção dos códigos de simulação e comparação com os dados experimentais permitiu uma melhor compreensão da manobra de transição. Tudo indica que não ocorre grande variação de altitude durante a manobra, isso é um ponto extremamente positivo. Caso houvesse um ganho de altitude na manobra de transição do voo de cruzeiro para o voo pairado, o pouso seria muito demorado, pois o veículo não pode perder altitude rapidamente no voo pairado, a fim de evitar perda de autoridade de controle por diminuição da velocidade do escoamento nas superfícies de controle.

Também foi verificado que o modelo não está adequado para simular manobras latero-direcionais no voo de cruzeiro. Para uma melhor compreensão dessas manobras seria necessário fazer ajustes no modelo e nas equações de movimento. Argyle (2016), implementou um controlador desenvolvido em um modelo de outra aeronave para esta condição de voo.

Na condição de voo de cruzeiro é muito provável que um *tail-sitter* consiga ser pilotado por atuação direta nas superfícies de controle como uma aeronave convencional. Entretanto, a implementação de controle nessa condição permite que um SARP consiga exercer efetivamente o seu desempenho em alcance, seguindo uma rota predefinida.

6.2 Sugestões para Trabalhos Futuros

Com base no que foi estudado sobre a configuração *tail-sitter* em SARP, os seguintes estudos seriam de grande contribuição para o tema:

- Avaliação de instalação de sistema de hélice de passo variável, indicando quais seriam os ganhos de desempenho e compromisso com carga paga;
- Implementação e comparação de diferentes métodos de controle para estabilização e movimentação no voo pairado;
- Desenvolvimento da capacidade para ajustar o modelo para aeronaves com diferentes dimensões e distribuições de massa;
- Aprimoramento do modelo para melhor avaliação de manobras latero-direcionais no voo de cruzeiro e implementação de controle para seguir uma rota predefinida; e
- Desenvolvimento de sistemas para pouso em caso de falha do motor, considerando ambiente de operação terrestre ou marítimo.

REFERÊNCIAS BIBLIOGRÁFICAS

- AERO-NEWS NETWORK. **Aero-TV At XPO21: Martin UAV's V-BAT 128 Is impressing Military Operators.** 11 de outubro de 2021. Disponível em: <<https://www.youtube.com/watch?v=QBHbahd3hiY>>. Acesso em: 14 de outubro de 2022.
- AIRFOIL TOOLS. **Airfoil Tools**, 2022. Search 1638 airfoils. Disponível em <<http://airfoiltools.com/>> Acesso em: 14 de outubro de 2022.
- ARGYLE, M. E. **MODELING AND CONTROL OF A TAILSITTER WITH A DUCTED FAN**: Brigham Young University, 2016. 208 p.
- BRASIL. Marinha. Corpo de Fuzileiros Navais. Comando-Geral. **CGCFN-321**: Manual de Apoio Aéreo de Fuzileiros Navais. Rio de Janeiro, 2020.
- BUTTLER, T.; DELEZENNE, J. **X-Planes of Europe: Secret Research Aircraft from the Golden Age 1946-1974**. 1ª. ed. Reino Unido: Hikoki Publications, 2012.
- CHAPRA, S. C. **Applied Numerical Methods with MATLAB for Engineers and Scientists**. 3ª. ed. New York: McGraw-Hill. 2012.
- CHATURVEDI, S. K. *et al.* Comparative Review Study of Military and Civilian Unmanned Aerial Vehicles (UAVs). **INCAS BULLETIN**, Bucareste, Romênia, Setembro 2019.
- COWIN, H. W. **RESEARCH AIRCRAFT 1891-1970: X-PLANES: A UNIQUE PICTORAL RECORD OF FLYING PROTOTYPES, THEIR DESIGNERS AND PILOTS**. 1ª. ed. Reino Unido: Osprey Publishing, 1999.
- FELIX, F. F.; BRAUN, C.; BIL, C. **Asia-Pacific International Symposium on Aerospace Technology**. A Review of Configuration Design for Distributed Propulsion Transition VTOL Aircraft, Seul, Coreia, 2017
- FT SISTEMAS. **FT-100**. Disponível em: <<http://flighttech.com.br/rotor-ft/ft-100.html>>. Acesso em: 14 de outubro de 2022.
- HARTMAN, E. P.; BIERMANN, D. REPORT No. 640: **THE AERODYNAMIC CHARACTERISTICS OF FULL-SCALE PROPELLERS HAVING 2,3 AND 4 BLADES OF CLARK Y AND R.A.F. 6 AIRFOIL SECTIONS**. Estados Unidos, NACA, 1938.
- INSITU. **ScanEagle**. Disponível em: <<https://www.insitu.com/products/scaneagle>>. Acesso em: 14 de outubro de 2022.
- JOHNSON, E. N; TURBE, M. A. **Journal of Guidance, Control, and Dynamics**. Modeling, Control, and Flight Testing of a Small Ducted Fan Aircraft. San Francisco, CA, Estados Unidos, 2005.

- KAUFFMAN, A. Martin UAV Downselected to Participate in Mi2 Expeditionary UAS Tech Demo. **NEWSWIRE**. Texas, 24 de novembro de 2020. Disponível em <<https://www.newswire.com/news/martin-uav-downselected-to-participate-in-mi2-expeditionary-uas-tech-21263613>> Acesso em: 14 de outubro de 2022.
- OGATA, K. **Modern Control Engineering**. 5ª. ed. Nova Jersey, Estados Unidos: Prentice-Hall, 2010.
- RAYMER, D. **Aircraft Design – A Conceptual Approach**. 6ª. ed. Estados Unidos: AIAA, 2018.
- SHIELD AI. **V-BAT**. Disponível em: <<https://shield.ai/v-bat/>>. Acesso em: 14 de outubro de 2022.
- SPELLER, I.; TUCK, C. **AMPHIBIOUS WARFARE: THE THEORY AND PRACTICE OF AMPHIBIOUS OPERATIONS IN THE 20TH CENTURY**. Reino Unido: Spellmount, 2001.
- STEVENS, B. L.; LEWIS, F. L.; JOHNSON, E. N. **AIRCRAFT CONTROL AND SIMULATION: Dynamics, Controls Design, and Autonomous Systems**. 3ª. Ed. New Jersey: John Wiley & Sons, Inc, 2016.
- SUTTER INDUSTRIES. **TOA 288**. Disponível em: <<https://www.suter-aviation.ch/toa-288.html>>. Acesso em: 14 de outubro de 2022.
- ZAMRI, O. **INTELLIGENT CONTROL OF A DUCTED-FAN VTOL UAV WITH CONVENTIONAL CONTROL SURFACES**: RMIT University, 2010. 269 p.

APÊNDICE A – Código do bloco s-function

```
function [sys,x0,str,ts,simStateCompliance] = vbatmdl(t,x,U,flag)
%SFUNTMPL GeneralS-Function Template
% With S-functions, you can define your own ordinary differential
% equations (ODEs), discrete system equations, and/or just about
% any type of algorithm to be used within a block diagram.
%
% The general form of S-function syntax is:
% [SYS,X0,STR,TS,SIMSTATECOMPLIANCE] = SFUNC(T,X,U,FLAG,P1,...,Pn)
%
% What is returned by SFUNC at a given point in time, T, depends on the
% value of the FLAG, the current state vector, X, and the current
% input vector, U.
%
% FLAG      RESULT      DESCRIPTION
% -----
% 0          [SIZES,X0,STR,TS] Initialization, return system sizes in SYS,
%                               initial state in X0, state ordering strings
%                               in STR, and sample times in TS.
% 1          DX          Return continuous state derivatives in SYS.
% 2          DS          Update discrete states SYS = X(n+1)
% 3          Y           Return outputs in SYS.
% 4          TNEXT       Return next time hit for variable step sample
%                               time in SYS.
% 5                               Reserved for future (root finding).
% 9          []          Termination, perform any cleanup SYS=[].
%
% The state vectors, X and X0 consists of continuous states followed
% by discrete states.
%
% Optional parameters, P1,...,Pn can be provided to the S-function and
% used during any FLAG operation.
%
% When SFUNC is called with FLAG = 0, the following information
% should be returned:
%
%   SYS(1) = Number of continuous states.
%   SYS(2) = Number of discrete states.
%   SYS(3) = Number of outputs.
%   SYS(4) = Number of inputs.
%           Any of the first four elements in SYS can be specified
%           as -1 indicating that they are dynamically sized. The
%           actual length for all other flags will be equal to the
%           length of the input, U.
%   SYS(5) = Reserved for root finding. Must be zero.
%   SYS(6) = Direct feedthrough flag (1=yes, 0=no). The s-function
%           has direct feedthrough if U is used during the FLAG=3
%           call. Setting this to 0 is akin to making a promise that
%           U will not be used during FLAG=3. If you break the promise
%           then unpredictable results will occur.
%   SYS(7) = Number of sample times. This is the number of rows in TS.
%
%   X0      = Initial state conditions or [] if no states.
%
%   STR     = State ordering strings which is generally specified as [].
%
%   TS      = An m-by-2 matrix containing the sample time
%           (period, offset) information. Where m = number of sample
%           times. The ordering of the sample times must be:
%
%           TS = [0      0,      : Continuous sample time.
```

```

%           0      1,      : Continuous, but fixed in minor step
%                               sample time.
%           PERIOD OFFSET, : Discrete sample time where
%                               PERIOD > 0 & OFFSET < PERIOD.
%           -2      0];    : Variable step discrete sample time
%                               where FLAG=4 is used to get time of
%                               next hit.
%
%           There can be more than one sample time providing
%           they are ordered such that they are monotonically
%           increasing. Only the needed sample times should be
%           specified in TS. When specifying more than one
%           sample time, you must check for sample hits explicitly by
%           seeing if
%           abs(round((T-OFFSET)/PERIOD) - (T-OFFSET)/PERIOD)
%           is within a specified tolerance, generally 1e-8. This
%           tolerance is dependent upon your model's sampling times
%           and simulation time.
%
%           You can also specify that the sample time of the S-function
%           is inherited from the driving block. For functions which
%           change during minor steps, this is done by
%           specifying SYS(7) = 1 and TS = [-1 0]. For functions which
%           are held during minor steps, this is done by specifying
%           SYS(7) = 1 and TS = [-1 1].
%
%           SIMSTATECOMPLIANCE = Specifies how to handle this block when saving and
%                               restoring the complete simulation state of the
%                               model. The allowed values are: 'DefaultSimState',
%                               'HasNoSimState' or 'DisallowSimState'. If this value
%                               is not specified, then the block's compliance with
%                               simState feature is set to 'UnknownSimState'.
%
%
% The following outlines the general structure of an S-function.
%
switch flag,

    %%%%%%%%%%%%%%%
    % Initialization %
    %%%%%%%%%%%%%%%
    case 0,
        [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;

    %%%%%%%%%%%%%%%
    % Derivatives %
    %%%%%%%%%%%%%%%
    case 1,
        sys=mdlDerivatives(t,x,U);

    %%%%%%%%%%%
    % Update %
    %%%%%%%%%%%
    case 2,
        sys=mdlUpdate(t,x,U);

    %%%%%%%%%%%
    % Outputs %
    %%%%%%%%%%%
    case 3,
        sys=mdlOutputs(t,x,U);

    %%%%%%%%%%%%%%%
    % GetTimeOfNextVarHit %
    %%%%%%%%%%%%%%%

```

```

case 4,
    sys=mdlGetTimeOfNextVarHit(t,x,U);

    %%%%%%%%%%%%%%
    % Terminate %
    %%%%%%%%%%%%%%
case 9,
    sys=mdlTerminate(t,x,U);

    %%%%%%%%%%%%%%
    % Unexpected flags %
    %%%%%%%%%%%%%%
otherwise
    DAStudio.error('blocks:unhandledFlag', num2str(flag));

end

% end sfuntmpl

%
%=====
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=====
%
function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes

%
% call simsizes for a sizes structure, fill it in and convert it to a
% sizes array.
%
% Note that in this example, the values are hard coded. This is not a
% recommended practice as the characteristics of the block are typically
% defined by the S-function parameters.
%
sizes = simsizes;

sizes.NumContStates = 13;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 13;
sizes.NumInputs = 12;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1; % at least one sample time is needed

sys = simsizes(sizes);

%
% initialize the initial conditions
%
% Condições iniciais para voo pairado
n0=1/sqrt(2);n1=0;n2=1/sqrt(2);n3=0;
x0 = [0 0 0 0 0 0 n0 n1 n2 n3 0 0 100];

% Condições iniciais para voo de cruzeiro
% load('workspace_LevelFlightTrimData', 'Trim')
%
%
% TrimQuat=eul2quat([0,Trim(6),Trim(7)]);
% n0=TrimQuat(1);n1=TrimQuat(2);n2=TrimQuat(3);n3=TrimQuat(4);
%
%
%
% str is always an empty matrix
%
str = [];

```

```

%
% initialize the array of sample times
%
ts = [0 0];

% Specify the block simStateCompliance. The allowed values are:
%   'UnknownSimState', < The default setting; warn and assume DefaultSimState
%   'DefaultSimState', < Same sim state as a built-in block
%   'HasNoSimState',   < No sim state
%   'DisallowSimState' < Error out when saving or restoring the model sim state
simStateCompliance = 'UnknownSimState';

% end mdlInitializeSizes

%
%=====
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
%
function sys=mdlDerivatives(t,x,U)
    global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
    global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
    global t0 eli e2i e3i e0
    global u v w uw vw ww h

    % Controles

    Dv(1)=U(1);
    Dv(2)=U(2);
    Dv(3)=U(3);
    Dv(4)=U(4);
    Dv(5)=U(5);
    Dv(6)=U(6);
    Dv(7)=U(7);
    Dv(8)=U(8);
    Dt=U(9);

    % vento
    uw=U(10);
    vw=U(11);
    ww=U(12);

    % Asas e fuselagem
    m=5.443;
    Jx=0.3745;
    Jy=1.4327;
    Jz=1.6931;
    Jxz=0.0015;
    J=[Jx 0 -Jxz;0 Jy 0;-Jxz 0 Jz];
    S=0.696;
    b=1.8479;
    c=0.2;
    Cy0=0;
    Cl0=0;
    Cn0=0;
    Clq=2.8932;
    Cdq=0;
    Cm q=-1.399;
    Cyp=0;
    Cyr=0;
    Clp=-0.3209;
    Clr=0.03066;
    Cnp=0.01297;

```

```

Cnr=-0.00434;
Cm0=-0.0721;
Cma=-0.712;
Cyb=-0.125;
Clb=-0.167;
Cnb=0.0475;

% Propulsor

global Ktwist a0 bp cp rp kaug
Ktwist=0.486;
a0=1.386;
bp=2;
cp=0.04;
rp=0.2286;
kaug=0.4548;

% Aletas de controle

global Clv Cdv Sv dvcp lv
Clv=0.25;
Cdv=0.1659;
Sv=0.17;
dvcp=0.71;
lv=0.15;

% Duto

global cduct rduct lduct
cduct=0.1542;
rduct=0.2377;
lduct=0.7;

% gravidade
global g
g=9.81;

u=x(1);
v=x(2);
w=x(3);
p=x(4);
q=x(5);
r=x(6);
n0=x(7);
n1=x(8);
n2=x(9);
n3=x(10);
pn=x(11);
pe=x(12);
h=x(13);

% Atmosfera padrão
[~,~,rho]=AtmPadrao(h);
disp(['Tempo =',num2str(t,3)])

% Gravidade

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

% Motor
global vi sol

TMax=71.7604
wp=Dt*8000*2*pi/60;

```

```

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);

SatT=saturation([0,20]);
VelT=evaluate(SatT, u-uw);

T=interp1([0,20],[TMax,48],VelT);

T=Dt*T;
    if isempty(sol)==1
        sol=[10;10]
    end
[sol,~]=fminsearch(@(X) Fprop2(X,wp,T,u,v,w),sol);
vi=sol(1);

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;
ud=u-uw; vd=v-vw+r*lduct; wd=w-ww+q*lduct;

V0=sqrt(ud^2+vd^2+wd^2);
if ud~=0
    alphad=atan(sqrt(vd^2+wd^2)/ud)
elseif (vd^2+wd^2)~=0
    alphad=pi/2
else
    alphad=0
end

if wd~=0 && vd~=0
    betad=atan(vd/wd)
elseif wd>0 && vd==0
    betad=0
elseif wd<0 && vd==0
    betad=pi
elseif wd==0 && vd>0
    betad=pi/2
elseif wd==0 && vd<0
    betad=-pi/2
else
    betad=0
end

fdd=(kd*rho*pi*rduct^2)*(V0*cos(alphad)+vi)*V0*sin(alphad);

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes
Vv=ud+vi;

Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Cdv;

```



```

    Fxv=Fxv-abs(Fdv(i));
    Fyv=Fyv+cos((i-1)*pi/4)*Flv(i);
    Fzv=Fzv+sin((i-1)*pi/4)*Flv(i);
    Mxv=Mxv-(lv/2)*Flv(i);
end
Fv=[Fxv;Fyv;Fzv]
Mv=[Mxv;dvcp*Fzv;-dvcp*Fyv]

% Airframe

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
if Va<10
    if ur~=0
        alpha=atan(wr/ur);
    else
        alpha=sign(wr)*pi/2;
    end

    if wr~=0 && vr~=0
        betah=atan(vr/wr);
    elseif wr>0 && vr==0
        betah=0;
    elseif wr<0 && vr==0
        betah=pi;
    elseif wr==0 && vr>0
        betah=pi/2;
    elseif wr==0 && vr<0
        betah=-pi/2;
    else
        betah=0;
    end

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cs*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*sqrt(vr^2+wr^2)*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]
    else
        Fw=[0;0;0]
        Mw=[0;0;0]
    end
end

else
    alpha=atan(wr/ur);
    beta=asin(vr/Va);
    CL=CLift(alpha);

```

```

Cd=CDrag(alpha);
Cx=-Cd*cos(alpha)+CL*sin(alpha);
Cy=Cy0+Cyb*beta;
Cz=-Cd*sin(alpha)-CL*cos(alpha);
Cxq=Clq*sin(alpha);
Czq=-Clq*cos(alpha);
Cl=0+Clb*beta;
Cm=Cm0+Cma*alpha;
Cn=Cn0+Cnb*beta;

Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
Fw=[Fwx;Fwy;Fwz]
Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
Mw=[Mwx;Mwy;Mwz]
end
% Somatório das forças

F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw

% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];
dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);
domega=J\ (cross(-omega,J*omega)+M);
dp=domega(1);dq=domega(2);dr=domega(3);
OMEGA=(1/2)*[0 -p -q -r;p 0 r -q;q -r 0 p;r q -p 0];
dquat=OMEGA*[n0;n1;n2;n3];
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);
dpos=R*[u;v;w];
dpn=dpos(1);dpe=dpos(2);dh=-dpos(3);
DY=[du dv dw dp dq dr dn0 dn1 dn2 dn3 dpn dpe dh];
DY=DY.';
sys = DY;

% end mdlDerivatives

%
%=====
% mdlUpdate
% Handle discrete state updates, sample time hits, and major time step
% requirements.
%=====
%
function sys=mdlUpdate(t,x,U)

sys = [];

% end mdlUpdate

%
%=====
% mdlOutputs
% Return the block outputs.
%=====
%
function sys=mdlOutputs(t,x,U)

sys = [x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) x(10) x(11) x(12) x(13)];

```

```
% end mdlOutputs

%
%=====
% mdlGetTimeOfNextVarHit
% Return the time of the next hit for this block. Note that the result is
% absolute time. Note that this function is only used when you specify a
% variable discrete-time sample time [-2 0] in the sample time array in
% mdlInitializeSizes.
%=====
%
function sys=mdlGetTimeOfNextVarHit(t,x,U)

sampleTime = 1;    % Example, set the next hit to be one second later.
sys = t + sampleTime;

% end mdlGetTimeOfNextVarHit

%
%=====
% mdlTerminate
% Perform any end of simulation tasks.
%=====
%
function sys=mdlTerminate(t,x,U)

sys = [];

% end mdlTerminate
```

APÊNDICE B – Funções

```

function [T,P,RO] = AtmPadrao(h)
    %Código para calcular temperatura, densidade e pressão em função da
    %altitude
    Ps=1.01325*10^5;ROs=1.225;Ts=288.16;R=286.9;g=9.81;a1=-6.5*10^-3;

    if (0<=h & h<=11000)

        T=Ts+h*a1;
        P=Ps*(T/Ts)^(-g/(a1*R));
        RO=ROs*(T/Ts)^-(g/(a1*R)+1);

    else
        T=216.66;
        P1=Ps*(T/Ts)^(-g/(a1*R));
        RO1=ROs*(T/Ts)^-(g/(a1*R)+1);
        P=P1*exp(-g*(h-11000)/(R*T));
        RO=RO1*exp(-g*(h-11000)/(R*T));
    end
end

function CD = CDrag(a)
    % Cálculo do coeficiente de arrasto

    ad=rad2deg(a);

    if abs(ad)>100
        if ad<(-100)
            a=a+pi;
            ad=ad+180;
        end
        if ad>100
            a=a-pi;
            ad=ad-180;
        end
    end

    if (-100<= ad & ad<=-66)
        CD=-0.7-3.37*a-1.15*a^2;
    end
    if (-66<=ad & ad<=-12)
        CD=-0.1074-1.55*a;
    end
    if (-12<= ad & ad<=11)
        CD=0.07-0.14*a+2.7*a^2;
    end
    if (11<= ad & ad<=70)
        CD=-0.15+1.55*a;
    end
    if (70<= ad & ad<=100)
        CD=-1.57+4.6*a-1.55*a^2;
    end
end

function CL = CLift(a)
    % Cálculo do coeficiente de sustentação

    ad=rad2deg(a);

    if abs(ad)>100
        if ad<(-100)

```

```

        a=a+pi;
        ad=ad+180;
    end
    if ad>100
        a=a-pi;
        ad=ad-180;
    end
end

if ((-35.5<=ad) && (ad<=-9.5))
    CL=-0.0894+1.27*a;
end
if ((-54<=ad) && (ad<=-35.5))
    CL=-0.7522+0.2*a;
end
if ((-100<=ad) && (ad<=-54))
    CL=-2.4427-1.6*a;
end
if ((-9.5<= ad) && (ad<=7.5))
    CL=0.483+4.78*a;
end
if ((7.5<=ad) && (ad<=12))
    CL=1.2399-a;
end
if ((12<=ad) && (ad<=53))
    CL=0.7+1.27*a-0.941*a^2;
end
if ((53<=ad) && (ad<=100))
    CL=2.55-1.6*a;
end

end

function [Y] = Fprop2(X,wp,T,u,v,w)

    global m Jx Jy Jz Jxz S b cw Cy0 Cl0 Cn Clq Cdq
    global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
    global Ktwist a0 bp cp rp kaug
    global Clv Cdv Sv dvcp lv
    global cduct rduct lduct
    global uw vw ww h

    [~,~,rho]=AtmPadrao(h);

    vi=X(1);
    vf=X(2);

    vb=u+(1/2)*wp*rp*Ktwist;

    %eq1=((1+kaug)/4)*(vb-vi)*wp*rp^2*rho*a0*bp*cp-T;
    eq2=sqrt((v-vw)^2+(w-ww)^2+(u-uw+vi)^2)-vf;
    eq3=T/(2*rho*pi*rp^2*vf)-vi;

    Y=eq2^2+eq3^2;
end

```

APÊNDICE C – Código de teste do motor antes das correções

```

clear all
clc
% Este código tem o propósito demonstrar os resultados do modelo do
% conjunto motopropulsor

global Ktwist a0 bp cp rp kaug
global u v w uw vw ww h

% Dados da hélice
Ktwist=0.486;
a0=1.386;
bp=2;
cp=0.04;
rp=0.2286;
kaug=0.4548;

h=0;%Altitude
u=5;v=0;w=0;%Velocidades
uw=0;vw=0;ww=0;%Velocidades do vento

rho=AtmPadrao(h);%Densidade do ar

i=1;
vi=3; % Chute inicial de velocidade induzida
for Dt=0.01:0.01:1

    wp=Dt*8000*2*pi/60;
    [T,vi] = Fprop(wp,vi)
    Pind=(vi*T)/(1+kaug);
    Mind=Pind/wp;

    VetDt(i)=Dt;
    VetFr(i)=T;
    VetMr(i)=Mind;
    VetVi(i)=vi;
    VetRPM(i)=wp*60/(2*pi);
    i=i+1;
end

subplot(2,2,1)

plot(VetRPM,VetFr)
xlabel('RPM')
ylabel('Tração (N)')

subplot(2,2,3)
plot(VetRPM,VetMr)
xlabel('RPM')
ylabel('Momento de rolagem (Nm)')

subplot(2,2,2)
plot(VetRPM,VetVi)
xlabel('RPM')
ylabel('Velocidade induzida (m/s)')

function [T,vilast] = Fprop(wp,vi)

global Ktwist a0 bp cp rp kaug
global u v w uw vw ww h

```

```
[Temp, Pr, rho]=AtmPadrao(h);  
  
vb=u-uw+(2/3)*wp*rp*(3/4)*Ktwist;  
  
Te=((1+kaug)/4)*(vb-vi)*wp*rp^2*rho*a0*bp*cp;  
vf=sqrt((v-vw)^2+(w-ww)^2+(u-uw+vi)^2);  
vi2=Te/(2*rho*pi*rp^2*vf);  
  
while abs(vi2-vi)>0.0001  
    vi=vi+0.001*(vi2-vi);  
    Te=((1+kaug)/4)*(vb-vi)*wp*rp^2*rho*a0*bp*cp;  
    vf=sqrt((v-vw)^2+(w-ww)^2+(u-uw+vi)^2);  
    vi2=Te/(2*rho*pi*rp^2*vf);  
end  
T=Te;  
vilast=vi2;  
end
```

APÊNDICE D – Código de trim no voo pairado

```
clear all
clc

% Dados do SARP

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb

% Asas e fuselagem
m=5.443;
Jx=0.3745;
Jy=1.4327;
Jz=1.6931;
Jxz=0.0015;
J=[Jx 0 -Jxz;0 Jy 0;-Jxz 0 Jz];
S=0.696;
b=1.8479;
c=0.2;
Cy0=0;
Cl0=0;
Cn0=0;
Clq=2.8932;
Cdq=0;
Cm q=-1.399;
Cyp=0;
Cyr=0;
Clp=-0.3209;
Clr=0.03066;
Cnp=0.01297;
Cnr=-0.00434;
Cm0=0.0721;
Cma=-0.712;
Cyb=-0.125;
Clb=-0.167;
Cnb=0.0475;

% Propulsor

global Ktwist a0 bp cp rp kaug
Ktwist=0.486;
a0=1.386;
bp=2;
cp=0.04;
rp=0.2286;
kaug=0.4548;
```



```
% Aletas de controle

global Clv Cdv Sv dvcp lv
Clv=0.25;
Cdv=0.1659;
Sv=0.17;
dvcp=0.71;
lv=0.15;

% Duto

global cduct rduct lduct
cduct=0.1542;
rduct=0.2377;
lduct=0.7;

% Condições iniciais
global u v w uw vw ww h g p q r quat

p=0;q=0;r=0;

g=9.81
u=0; v=0; w=0;

uw=0; vw=0; ww=0;

h=100;

quat=eul2quat([0 pi/2 0]);

X0=[-0.6 0 0 0.9];

options=optimset('TolFun',10^-30,'MaxIter',10000,'MaxFunEvals',10000);

[TRIM,FVAL]=fminsearch(@FObjTrim, X0,options);

global Da De Dr Dt

Da=TRIM(1);De=TRIM(2); Dr=TRIM(3); Dt=TRIM(4);

n0=quat(1);
n1=quat(2);
n2=quat(3);
n3=quat(4);
```

```

Y0=[u v w p q r n0 n1 n2 n3 0 0 h];

Teste=Dinamica(0,[0 0 0 p q r n0 n1 n2 n3 0 0 h])

fprintf('Resultados do Trim:\n Aileron: %5.4f\n Profundor: %5.4f\n Leme: %5.4f\n
Motor: %5.4f\n\n\n',Da,De,Dr,Dt)
fprintf('Teste de Dinâmica\n ndu/dt: %5.4f\ndv/dt: %5.4f\ndw/dt:
%5.4f\n',Teste(1),Teste(2),Teste(3))
fprintf('dp/dt: %5.4f\ndq/dt: %5.4f\ndr/dt: %5.4f\n',Teste(4),Teste(5),Teste(6))
fprintf('dn0/dt: %5.4f\ndn1/dt: %5.4f\ndn2/dt: %5.4f\ndn3/dt:
%5.4f\n',Teste(7),Teste(8),Teste(9),Teste(10))
fprintf('dpn/dt: %5.4f\ndpe/dt: %5.4f\ndh/dt:
%5.4f\n',Teste(11),Teste(12),Teste(13))

function Y = FObjTrim(X)
% Hover Flight Trim
global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
global Ktwist a0 bp cp rp kaug
global Clv Cdv Sv dvcp lv
global cduct rduct lduct
global quat u v w uw vw ww h g p q r

Da=X(1);
De=X(2);
Dr=X(3);
Dt=X(4);

[Temp,Pr,rho]=AtmPadrao(h);
% Gravidade

n0=quat(1);
n1=quat(2);
n2=quat(3);
n3=quat(4);

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

% Motor

```

```

global vi sol

TMax=71.7604
wp=Dt*8000*2*pi/60;
ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);

SatT=saturation([0,20])
VelT=evaluate(SatT, u-uw)

T=interp1([0,20],[TMax,48],VelT)
T=Dt*T;
if isempty(sol)==1
    sol=[10;10];
end
[sol,~]=fminsearch(@(X) Fprop2(X,wp,T,u,v,w),sol);
vi=sol(1);

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;
ud=u-uw; vd=v-vw+r*lduct; wd=w-ww+q*lduct;

V0=sqrt(ud^2+vd^2+wd^2);
if ud~=0
    alphas=atan(sqrt(vd^2+wd^2)/ud);
elseif (vd^2+wd^2)~=0
    alphas=pi/2;
else
    alphas=0;
end

if wd~=0 && vd~=0
    betas=atan(vd/wd);
elseif wd>0 && vd==0
    betas=0;
elseif wd<0 && vd==0
    betas=pi;
elseif wd==0 && vd>0
    betas=pi/2;
elseif wd==0 && vd<0
    betas=-pi/2;

```

```

else
    betad=0;
end

fdd=(kd*rho*pi*rduct^2)*(V0*cos(alphad)+vi)*V0*sin(alphad);

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes

Dv=zeros(1,8);
Vv=ud+vi;
sat=saturation([-1 1]);
Dv(1)=evaluate(sat,-Da+Dr);
Dv(2)=evaluate(sat,-Da+De+Dr);
Dv(3)=evaluate(sat,-Da+De);
Dv(4)=evaluate(sat,-Da+De-Dr);
Dv(5)=evaluate(sat,-Da-Dr);
Dv(6)=evaluate(sat,-Da-De-Dr);
Dv(7)=evaluate(sat,-Da-De);
Dv(8)=evaluate(sat,-Da-De+Dr);
Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*(Vv^2)*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*(Vv^2)*Sv*Dv(i)*thetamax*Cdv;
    F xv=F xv-abs(F dv(i));
    F yv=F yv+cos((i-1)*pi/4)*Flv(i);
    F zv=F zv+sin((i-1)*pi/4)*Flv(i);
    M xv=M xv-(lv/2)*Flv(i);
end
Fv=[F xv;F yv;F zv]
Mv=[M xv;dvcp*F zv;-dvcp*F yv]

% Airframe

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
if Va<10
    if ur~=0
        alpha=atan(wr/ur);
    else

```

```

        alpha=sign(wr)*pi/2;
    end

    if wr~=0 && vr~=0
        betah=atan(vr/wr);
    elseif wr>0 && vr==0
        betah=0;
    elseif wr<0 && vr==0
        betah=pi;
    elseif wr==0 && vr>0
        betah=pi/2;
    elseif wr==0 && vr<0
        betah= -pi/2;
    else
        betah=0;
    end

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cs*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/2/Va)*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/2/Va)*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*sqrt((vr^2+wr^2))^2*(Cl+(b/(2*Va))*Clp*p+Clr*r);
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]
    else
        Fw=[0;0;0]
        Mw=[0;0;0]
    end
end
else

```

```

alpha=atan(wr/ur);
beta=asin(vr/Va);
CL=CLift(alpha);
Cd=CDrag(alpha);
Cx=-Cd*cos(alpha)+CL*sin(alpha);
Cy=Cy0+Cyb*beta;
Cz=-Cd*sin(alpha)-CL*cos(alpha);
Cxq=Clq*sin(alpha);
Czq=-Clq*cos(alpha);
Cl=0+Clb*beta;
Cm=Cm0+Cma*alpha;
Cn=Cn0+Cnb*beta;

Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
Fw=[Fwx;Fwy;Fwz]
Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/(2*Va))*Clp*p+Clr*r);
Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
Mw=[Mwx;Mwy;Mwz]

end

%Fw=[0;0;0] % Zerado
%Mw=[0;0;0] % Zerado

% Somatório das forças

F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw

% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];

dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);

domega=J\ (cross(-omega,J*omega)+M);
dp=domega(1); dq=domega(2); dr=domega(3);

OMEGA=(1/2)*[0 -p -q -r;p 0 r -q; q -r 0 p;r q -p 0];
dquat=OMEGA*[n0;n1;n2;n3];
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);

```

```

dpos=R*[u;v;w];
dpn=dpos(1);dpe=dpos(2);dh=-dpos(3);

Y=dp^2+dq^2+dr^2+du^2;

end

function DY = Dinamica(t,Y,cmd)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

u=Y(1);
v=Y(2);
w=Y(3);
p=Y(4);
q=Y(5);
r=Y(6);
n0=Y(7);
n1=Y(8);
n2=Y(9);
n3=Y(10);
pn=Y(11);
pe=Y(12);
h=Y(13);

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm0 Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
global Ktwist a0 bp cp rp kaug
global Clv Cdv Sv dvcp lv
global cduct rduct lduct
global quat uw vw ww g
global Da De Dr Dt t0 eli e2i e3i e0
global vi0

[Temp,Pr,rho]=AtmPadrao(h);

disp(t)

% Gravidade

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

```

```

% Motor
global vi sol

TMax=71.7604
wp=Dt*8000*2*pi/60;
ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
T=interp1([0,20],[TMax,48],abs(u-uw));
T=Dt*T;

if isempty(sol)==1
    sol=[10;10];
end

[sol,~]=fmincon(@ (X) Fprop2(X,wp,T,u,v,w),sol);
vi=sol(1);

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;
ud=u-uw; vd=v-vw+r*lduct; wd=w-ww+q*lduct;

V0=sqrt(ud^2+vd^2+wd^2);
if ud~=0
    alphad=atan(sqrt(vd^2+wd^2)/ud)
elseif (vd^2+wd^2)~=0
    alphad=pi/2
else
    alphad=0
end

if wd~=0 && vd~=0
    betad=atan(vd/wd)
elseif wd>0 && vd==0
    betad=0
elseif wd<0 && vd==0
    betad=pi
elseif wd==0 && vd>0
    betad=pi/2
elseif wd==0 && vd<0
    betad=-pi/2
else
    betad=0
end

```



```

fdd=kd*rho*pi*rduct^2*(V0*cos(alphad)+vi)*V0*sin(alphad);

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes

Vv=ud+vi;
sat=saturation([-1 1]);
Dv(1)=evaluate(sat,-Da+Dr);
Dv(2)=evaluate(sat,-Da+De+Dr);
Dv(3)=evaluate(sat,-Da+De);
Dv(4)=evaluate(sat,-Da+De-Dr);
Dv(5)=evaluate(sat,-Da-Dr);
Dv(6)=evaluate(sat,-Da-De-Dr);
Dv(7)=evaluate(sat,-Da-De);
Dv(8)=evaluate(sat,-Da-De+Dr);
Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Cdv;
    F xv=F xv-abs(F dv(i));
    F yv=F yv+cos((i-1)*pi/4)*Flv(i);
    F zv=F zv+sin((i-1)*pi/4)*Flv(i);
    M xv=M xv-(lv/2)*Flv(i);
end
Fv=[F xv;F yv;F zv]
Mv=[M xv;dvcp*F zv;-dvcp*F yv]

% Airframe

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
if Va<10
    if ur~=0
        alpha=atan(wr/ur);
    else
        alpha=sign(wr)*pi/2;
    end

    if wr~=0 && vr~=0
        betah=atan(vr/wr);
    elseif wr>0 && vr==0

```

```

        betah=0;
    elseif wr<0 && vr==0
        betah=pi;
    elseif wr==0 && vr>0
        betah=pi/2;
    elseif wr==0 && vr<0
        betah= -pi/2;
    else
        betah=0;
    end

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cs*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/2/Va)*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/2/Va)*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/2/Va)*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/2*Va)*Clp*p+Clr*r);
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/2/Va)*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/2/Va)*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]
    else
        Fw=[0;0;0]
        Mw=[0;0;0]
    end

else
    alpha=atan(wr/ur);
    beta=asin(wr/Va);
    CL=CLift(alpha);
    Cd=CDrag(alpha);
    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cy0+Cyb*beta;
    Cz=-Cd*sin(alpha)-CL*cos(alpha);

```

```

Cxq=Clq*sin(alpha);
Czq=-Clq*cos(alpha);
Cl=0+Clb*beta;
Cm=Cm0+Cma*alpha;
Cn=Cn0+Cnb*beta;

Fwx=(1/2)*rho*S*Va^2*(Cx+(c/2/Va)*Cxq*q);
Fwy=(1/2)*rho*S*Va^2*(Cy+(b/2/Va)*(Cyp*p+Cyrr));
Fwz=(1/2)*rho*S*Va^2*(Cz+(c/2/Va)*Czq*q);
Fw=[Fwx;Fwy;Fwz]
Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/2*Va)*Clp*p+Clr*r);
Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/2/Va)*Cmq*q);
Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/2/Va)*(Cnp*p+Cnr*r));
Mw=[Mwx;Mwy;Mwz]
end

F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw
% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];

dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);

domega=J\ (cross(-omega,J*omega)+M);
dp=domega(1); dq=domega(2); dr=domega(3);

OMEGA=(1/2)*[0 -p -q -r;p 0 r -q; q -r 0 p;r q -p 0];
dquat=OMEGA*[n0;n1;n2;n3];
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);

dpos=R*[u;v;w];
dpn=dpos(1);dpe=dpos(2);dh=-dpos(3);

DY=[du dv dw dp dq dr dn0 dn1 dn2 dn3 dpn dpe dh];
DY=DY.';

end

```

APÊNDICE E – Código de trim no voo de cruzeiro

```
clear all
clc

% Dados do SARP

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb

% Asas e fuselagem
m=5.443;
Jx=0.3745;
Jy=1.4327;
Jz=1.6931;
Jxz=0.0015;
J=[Jx 0 -Jxz;0 Jy 0;-Jxz 0 Jz];
S=0.696;
b=1.8479;
c=0.2;
Cy0=0;
Cl0=0;
Cn0=0;
Clq=2.8932;
Cdq=0;
Cm q=-1.399;
Cyp=0;
Cyr=0;
Clp=-0.3209;
Clr=0.03066;
Cnp=0.01297;
Cnr=-0.00434;
Cm0=0.0721;
Cma=-0.712;
Cyb=-0.125;
Clb=-0.167;
Cnb=0.0475;

% Propulsor

global Ktwist a0 bp cp rp kaug
Ktwist=0.486;
a0=1.386;
bp=2;
cp=0.04;
rp=0.2286;
kaug=0.4548;
```

```

% Aletas de controle

global Clv Cdv Sv dvcv lv
Clv=0.25;
Cdv=0.1659;
Sv=0.17;
dvcv=0.71;
lv=0.15;

% Duto

global cduct rduct lduct
cduct=0.1542;
rduct=0.2377;
lduct=0.7;

% Condições iniciais
global u v w uw vw ww h g p q r quat theta Va

p=0;q=0;r=0;

g=9.81
u=15; v=0;

uw=0; vw=0; ww=0;

h=100;

X0=[-0.6 0 0 0.8 0.1 0.1 0];

options=optimset('TolFun',10^-30,'MaxIter',10000,'MaxFunEvals',10000);

[Trim,FVAL]=fmincon(@FObjTrim1,X0,[],[],[],[],[-1 -1 -1 0 -2 -pi/2 -pi],[1 1 1 1 2
pi/2 pi],[],options)

X1=Trim;

X1(5)=atan(X1(5)/u);
X1(6)=rad2deg(X1(6));
X1(7)=rad2deg(X1(7));

global Da De Dr Dt
Da=Trim(1);De=Trim(2);Dr=Trim(3);Dt=Trim(4);

TrimQuat=eul2quat([0,Trim(6),Trim(7)]);
n0=TrimQuat(1);n1=TrimQuat(2);n2=TrimQuat(3);n3=TrimQuat(4);

```

```

X0 = [u 0 Trim(5) 0 0 0 n0 n1 n2 n3 0 0 100];

Teste=Dinamica(0,X0)

fprintf('Resultados do Trim:\n Aileron: %5.4f\n Profundor: %5.4f\n Leme: %5.4f\n
Motor: %5.4f\n w: %5.4f\n Theta: %5.4f\n Phi:
%5.4f\n\n\n',Da,De,Dr,Dt,Trim(5),Trim(6),Trim(7))
fprintf('Teste de Dinâmica\n\ndu/dt: %5.4f\ndv/dt: %5.4f\ndw/dt:
%5.4f\n',Teste(1),Teste(2),Teste(3))
fprintf('dp/dt: %5.4f\ndq/dt: %5.4f\ndr/dt: %5.4f\n',Teste(4),Teste(5),Teste(6))
fprintf('dn0/dt: %5.4f\ndn1/dt: %5.4f\ndn2/dt: %5.4f\ndn3/dt:
%5.4f\n',Teste(7),Teste(8),Teste(9),Teste(10))

function Y = FObjTrim1(X)
global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
global Ktwist a0 bp cp rp kaug
global Clv Cdv Sv dvcp lv
global cduct rduct lduct
global u v uw vw ww h g p q r

Da=X(1);
De=X(2);
Dr=X(3);
Dt=X(4);
w=X(5);
theta=X(6);
phi=X(7);

quat=eul2quat([0,theta,phi]);

n0=quat(1);n1=quat(2);n2=quat(3);n3=quat(4);

% Atmosfera padrão
[~,~,rho]=AtmPadrao(h);

% Gravidade

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

% Motor
global vi sol

load('workspace_motor', 'VetFr','VetVi','VetDt')
TMax=VetFr(100);
wp=Dt*8000*2*pi/60;

```

```

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);

SatT=saturation([0,20])
VelT=evaluate(SatT, u-uw)

T=interp1([0,20],[TMax,48],VelT)
disp(Dt)
T=Dt*T

if isempty(sol)==1
    sol=[10 ;10]
end

[sol,~]=fminsearch(@(X) Fprop2(X,wp,T,u,v,w),sol)

vi=sol(1)

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;
ud=u-uw; vd=v-vw+r*lduct; wd=w-ww+q*lduct;

V0=sqrt(ud^2+vd^2+wd^2);
if ud~=0
    alphad=atan(sqrt(vd^2+wd^2)/ud);
elseif (vd^2+wd^2)~=0
    alphad=pi/2;
else
    alphad=0;
end

if wd~=0 && vd~=0
    betad=atan(vd/wd);
elseif wd>0 && vd==0
    betad=0;
elseif wd<0 && vd==0
    betad=pi;
elseif wd==0 && vd>0
    betad=pi/2;

```

```

elseif wd==0 && vd<0
    betad= -pi/2;
else
    betad=0;
end

fdd=(kd*rho*pi*rduct^2)*(V0*cos(alphad)+vi)*V0*sin(alphad);

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes

Dv=zeros(1,8);
Vv=ud+vi;
sat=saturation([-1 1]);
Dv(1)=evaluate(sat,-Da+Dr);
Dv(2)=evaluate(sat,-Da+De+Dr);
Dv(3)=evaluate(sat,-Da+De);
Dv(4)=evaluate(sat,-Da+De-Dr);
Dv(5)=evaluate(sat,-Da-Dr);
Dv(6)=evaluate(sat,-Da-De-Dr);
Dv(7)=evaluate(sat,-Da-De);
Dv(8)=evaluate(sat,-Da-De+Dr);
Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Cdv;
    F xv=F xv-abs(F dv(i));
    F yv=F yv+cos((i-1)*pi/4)*Flv(i);
    F zv=F zv+sin((i-1)*pi/4)*Flv(i);
    M xv=M xv-(lv/2)*Flv(i);
end
Fv=[F xv;F yv;F zv]
Mv=[M xv;dvcp*F zv;-dvcp*F yv]

% Airframe

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
if Va<10
    if ur~=0

```



```

        alpha=atan(wr/ur);
    else
        alpha=sign(wr)*pi/2;
    end

    if wr~=0 && vr~=0
        betah=atan(vr/wr);
    elseif wr>0 && vr==0
        betah=0;
    elseif wr<0 && vr==0
        betah=pi;
    elseif wr==0 && vr>0
        betah=pi/2;
    elseif wr==0 && vr<0
        betah= -pi/2;
    else
        betah=0;
    end

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cs*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*sqrt(vr^2+wr^2)*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]
    else
        Fw=[0;0;0]
        Mw=[0;0;0]
    end
end

```

```

else
    alpha=atan(wr/ur);
    beta=asin(vr/Va);
    CL=CLift(alpha);
    Cd=CDrag(alpha);
    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cy0+Cyb*beta;
    Cz=-Cd*sin(alpha)-CL*cos(alpha);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cl=0+Clb*beta;
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*beta;

    Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
    Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
    Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
    Fw=[Fwx;Fwy;Fwz]
    Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
    Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
    Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
    Mw=[Mwx;Mwy;Mwz]
end

% Somatório das forças

F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw

% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];

dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);

domega=inv(J)*(cross(-omega,J*omega)+M);
dp=domega(1); dq=domega(2); dr=domega(3);

OMEGA=(1/2)*[0 -p -q -r;p 0 r -q; q -r 0 p;r q -p 0];
dquat=OMEGA*[n0;n1;n2;n3];
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);

```

```

dpos=R*[u;v;w];
dpn=dpos(1);dpe=dpos(2);dh=-dpos(3);

Y=du^2+dv^2+dw^2+dp^2+dq^2+dr^2+dh^2
end

function DY = Dinamica(t,Y)

u=Y(1);
v=Y(2);
w=Y(3);
p=Y(4);
q=Y(5);
r=Y(6);
n0=Y(7);
n1=Y(8);
n2=Y(9);
n3=Y(10);
pn=Y(11);
pe=Y(12);
h=Y(13);

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
global Ktwist a0 bp cp rp kaug
global Clv Cdv Sv dvcp lv
global cduct rduct lduct
global quat uw vw ww g
global Da De Dr Dt t0 e1i e2i e3i e0
global vi0

[Temp,Pr,rho]=AtmPadrao(h);

disp(t)

% Gravidade

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

% Motor
global vi sol

```

```

load('workspace_motor', 'VetFr', 'VetVi', 'VetDt')
TMax=VetFr(100);
wp=Dt*8000*2*pi/60;
ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);

SatT=saturation([0,20])
VelT=evaluate(SatT, u-uw)

T=interp1([0,20],[TMax,48],VelT)
disp(Dt)
T=Dt*T

if isempty(sol)==1
    sol=[10;10]
end

[sol,~]=fminsearch(@(X) Fprop2(X,wp,T,u,v,w),sol)

vi=sol(1)

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;
ud=u-uw; vd=v-vw+r*lduct; wd=w-ww+q*lduct;

V0=sqrt(ud^2+vd^2+wd^2);
if ud~=0
    alphad=atan(sqrt(vd^2+wd^2)/ud);
elseif (vd^2+wd^2)~=0
    alphad=pi/2;
else
    alphad=0;
end

if wd~=0 && vd~=0
    betad=atan(vd/wd);
elseif wd>0 && vd==0
    betad=0;

```

```

elseif wd<0 && vd==0
    betad=pi;
elseif wd==0 && vd>0
    betad=pi/2;
elseif wd==0 && vd<0
    betad= -pi/2;
else
    betad=0;
end

fdd=(kd*rho*pi*rduct^2)*(V0*cos(alphad)+vi)*V0*sin(alphad);

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes

Dv=zeros(1,8);
Vv=ud+vi;
sat=saturation([-1 1]);
Dv(1)=evaluate(sat,-Da+Dr);
Dv(2)=evaluate(sat,-Da+De+Dr);
Dv(3)=evaluate(sat,-Da+De);
Dv(4)=evaluate(sat,-Da+De-Dr);
Dv(5)=evaluate(sat,-Da-Dr);
Dv(6)=evaluate(sat,-Da-De-Dr);
Dv(7)=evaluate(sat,-Da-De);
Dv(8)=evaluate(sat,-Da-De+Dr);
Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Cdv;
    F xv=F xv-abs(F dv(i));
    F yv=F yv+cos((i-1)*pi/4)*Flv(i);
    F zv=F zv+sin((i-1)*pi/4)*Flv(i);
    M xv=M xv-(lv/2)*Flv(i);
end
Fv=[F xv;F yv;F zv]
Mv=[M xv;dvcv*F zv;-dvcv*F yv]

% Airframe

```

```

ur=u-uw; vr=v-vw; wr=w-ww;
Va=sqrt(ur^2+vr^2+wr^2);
if Va<10
    if ur~=0
        alpha=atan(wr/ur);
    else
        alpha=sign(wr)*pi/2;
    end

    if wr~=0 && vr~=0
        betah=atan(vr/wr);
    elseif wr>0 && vr==0
        betah=0;
    elseif wr<0 && vr==0
        betah=pi;
    elseif wr==0 && vr>0
        betah=pi/2;
    elseif wr==0 && vr<0
        betah= -pi/2;
    else
        betah=0;
    end

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cx*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*sqrt(vr^2+wr^2)*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]

```

```

else
    Fw=[0;0;0]
    Mw=[0;0;0]
end

else
    alpha=atan(wr/ur);
    beta=asin(vr/Va);
    CL=CLift(alpha);
    Cd=CDrag(alpha);
    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cy0+Cyb*beta;
    Cz=-Cd*sin(alpha)-CL*cos(alpha);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cl=0+Clb*beta;
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*beta;

    Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
    Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
    Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
    Fw=[Fwx;Fwy;Fwz]
    Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
    Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
    Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
    Mw=[Mwx;Mwy;Mwz]
end

% Somatório das forças
F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw

% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];

dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);

domega=inv(J)*(cross(-omega,J*omega)+M);
dp=domega(1); dq=domega(2); dr=domega(3);

OMEGA=(1/2)*[0 -p -q -r;p 0 r -q; q -r 0 p;r q -p 0];
dquat=OMEGA*[n0;n1;n2;n3];

```

```
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);

dpos=R*[u;v;w];
dpos=[dpos(1);dpos(2);dpos(3)];

DY=[du dv dw dp dq dr dn0 dn1 dn2 dn3 dpn dpe dh];
DY=DY.';

end
```


APÊNDICE F – Código de trim em transição

```
clear all
clc

% Dados do SARP

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cmq Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb

% Asas e fuselagem
m=5.443;
Jx=0.3745;
Jy=1.4327;
Jz=1.6931;
Jxz=0.0015;
J=[Jx 0 -Jxz;0 Jy 0;-Jxz 0 Jz];
S=0.696;
b=1.8479;
c=0.2;
Cy0=0;
Cl0=0;
Cn0=0;
Clq=2.8932;
Cdq=0;
Cmq=-1.399;
Cyp=0;
Cyr=0;
Clp=-0.3209;
Clr=0.03066;
Cnp=0.01297;
Cnr=-0.00434;
Cm0=0.0721;
Cma=-0.712;
Cyb=0.125;
Clb=-0.167;
Cnb=0.0475;

% Propulsor

global Ktwist a0 bp cp rp kaug
Ktwist=0.486;
a0=1.386;
bp=2;
cp=0.04;
rp=0.2286;
kaug=0.4548;
```

```

% Aletas de controle

global Clv Cdv Sv dvcp lv
Clv=0.25;
Cdv=0.1659;
Sv=0.17;
dvcp=0.71;
lv=0.15;

% Duto

global cduct rduct lduct
cduct=0.1542;
rduct=0.2377;
lduct=0.7;

% Condições iniciais
global u v w uw vw ww h g p q r quat theta Va

p=0;q=0;r=0;

g=9.81
Va=0; v=0;

uw=0; vw=0; ww=0;

h=100;

X0=[-0.6 0 0 0.8 0.1 0.1 0];

options=optimset('TolFun',10^-30,'MaxIter',10000,'MaxFunEvals',10000);

for i=1:1:21
    [Trim(:,i),FVAL(i)]=fmincon(@FObjTrim2,X0,[],[],[],[],[-1 -1 -1 0 -pi/2 -
    0.1*pi/2 -pi],[1 1 1 1 pi/2 pi/2 pi],[],options)

    Va=Va+1;
end

TrimDeg=Trim;

TrimDeg(5,:)=rad2deg(TrimDeg(5,:));
TrimDeg(6,:)=rad2deg(TrimDeg(6,:));
TrimDeg(7,:)=rad2deg(TrimDeg(7,:));

```

```

Table = array2table(TrimDeg,
'VariableNames',{ 'Vel_0','Vel_1','Vel_2','Vel_3','Vel_4',...
'Vel_5','Vel_6','Vel_7','Vel_8','Vel_9','Vel_10','Vel_11','Vel_12','Vel_13','Vel_14',...
'Vel_15','Vel_16','Vel_17','Vel_18','Vel_19','Vel_20',},
'RowName',{ 'Ail','Elev','Rud'...
'Thr','Alpha','Theta','Phi'});

disp(Table)

function Y = FObjTrim2(X)

global m J Jx Jy Jz Jxz S b c Cy0 Cl0 Cn0 Clq Cdq
global Cm q Cyp Cyr Clp Clr Cnp Cnr Cm0 Cma Cyb Clb Cnb
global Ktwist a0 bp cp rp kaug
global Clv Cdv Sv dvcp lv
global cduct rduct lduct
global v uv vw ww h g p q r Va

Da=X(1);
De=X(2);
Dr=X(3);
Dt=X(4);
alpha=X(5);
theta=X(6);
phi=X(7);

quat=eul2quat([0,theta,phi]);

n0=quat(1);n1=quat(2);n2=quat(3);n3=quat(4);

% Atmosfera padrão
[~,~,rho]=AtmPadrao(h);

% Gravidade

Fg=m*g*[2*(n1*n3-n2*n0);2*(n2*n3-n1*n0);2*n0^2+2*n3^2-1]

% Motor
global vi

load('workspace_motor','VetFr','VetVi','VetDt')
TMax=71.7604;
wp=Dt*8000*2*pi/60

```

```

u=Va*cos(alpha)
w=Va*sin(alpha)

SatT=saturation([0,20])
VelT=evaluate(SatT, u)

T=interp1([0,20],[TMax,48],VelT)
disp(Dt)
T=Dt*T

[sol,~]=fsolve(@(X) Fprop2(X,wp,T,u,v,w),[10;10])

vi=sol(1)

Pind=(vi*T)/(1+kaug);
Mind=Pind/wp;

Fr=[T;0;0]
Mr=[Mind;0;0]

% Duto
kd=1;

V0=Va;

betad=0

alphad=alpha

fdd=(kd*rho*pi*rduct^2)*(V0*cos(alphad)+vi)*V0*sin(alphad)

Fd=[0;-fdd*sin(betad);-fdd*cos(betad)]
Md=[0;Fd(3)*lduct;-Fd(2)*lduct]

% Vanes

Dv=zeros(1,8);
Vv=u+vi;
sat=saturation([-1 1]);
Dv(1)=evaluate(sat,-Da+Dr);
Dv(2)=evaluate(sat,-Da+De+Dr);
Dv(3)=evaluate(sat,-Da+De);
Dv(4)=evaluate(sat,-Da+De-Dr);
Dv(5)=evaluate(sat,-Da-Dr);
Dv(6)=evaluate(sat,-Da-De-Dr);
Dv(7)=evaluate(sat,-Da-De);

```

```

Dv(8)=evaluate(sat,-Da-De+Dr);
Flv=[0 0 0 0 0 0 0 0];
Fdv=[0 0 0 0 0 0 0 0];
F xv=0;F yv=0;F zv=0;
M xv=0;
for i=1:1:8
    thetamax=deg2rad(30);

    Flv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Clv;
    Fdv(i)=(1/2)*rho*Vv^2*Sv*Dv(i)*thetamax*Cdv;
    F xv=F xv-abs(F dv(i));
    F yv=F yv+cos((i-1)*pi/4)*Flv(i);
    F zv=F zv+sin((i-1)*pi/4)*Flv(i);
    M xv=M xv-(lv/2)*Flv(i);
end
Fv=[F xv;F yv;F zv]
Mv=[M xv;dvcv*F zv;-dvcv*F yv]

% Airframe

if Va<10
    betah=0

    CL=CLift(alpha)*(0.9*abs(cos(betah))+0.1);
    CL=(0.9*abs(cos(betah))+0.1)*CL;
    Cd=3.14*sin(1.03*betah+1.48)^2+0.7220;
    Cs=0.94*asin(sin(2*betah));
    Cl=-0.13*asin(sin(2*betah));
    Ch=abs(-CL*cos(alpha)-Cd*sin(alpha));

    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cs*cos(betah)-Ch*sin(betah);
    Cz=-Ch*cos(betah)-Cs*sin(betah);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*betah;

    if Va~=0
        Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
        Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
        Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
        Fw=[Fwx;Fwy;Fwz]
        Mwx=(1/2)*rho*S*b*sqrt(w^2)*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
        Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
        Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
        Mw=[Mwx;Mwy;Mwz]
    end
end

```

```

else
    Fw=[0;0;0]
    Mw=[0;0;0]
end

else

    beta=0
    CL=CLift(alpha);
    Cd=CDrag(alpha);
    Cx=-Cd*cos(alpha)+CL*sin(alpha);
    Cy=Cy0+Cyb*beta;
    Cz=-Cd*sin(alpha)-CL*cos(alpha);
    Cxq=Clq*sin(alpha);
    Czq=-Clq*cos(alpha);
    Cl=0+Clb*beta;
    Cm=Cm0+Cma*alpha;
    Cn=Cn0+Cnb*beta;

    Fwx=(1/2)*rho*S*Va^2*(Cx+(c/(2*Va))*Cxq*q);
    Fwy=(1/2)*rho*S*Va^2*(Cy+(b/(2*Va))*(Cyp*p+Cyr*r));
    Fwz=(1/2)*rho*S*Va^2*(Cz+(c/(2*Va))*Czq*q);
    Fw=[Fwx;Fwy;Fwz]
    Mwx=(1/2)*rho*S*b*Va^2*(Cl+(b/(2*Va))*(Clp*p+Clr*r));
    Mwy=(1/2)*rho*S*c*Va^2*(Cm+(c/(2*Va))*Cmq*q);
    Mwz=(1/2)*rho*S*b*Va^2*(Cn+(b/(2*Va))*(Cnp*p+Cnr*r));
    Mw=[Mwx;Mwy;Mwz]
end

% Somatório das forças

F=Fg+Fr+Fd+Fv+Fw
M=Mr+Md+Mv+Mw
% Cinemática

R=[2*n0^2+2*n1^2-1, 2*(2*n1*n2-n3*n0), 2*(n1*n3+n2*n0)
    2*(n1*n2+n3*n0), 2*n0^2+2*n2^2-1, 2*(n2*n3-n1*n0)
    2*(n1*n3-n2*n0), 2*(n2*n3+n1*n0), 2*n0^2+2*n3^2-1];

omega=[p;q;r]; Vt=[u;v;w];

dVt=cross(-omega,Vt)+F/m;
du=dVt(1);dv=dVt(2);dw=dVt(3);

domega=inv(J)*(cross(-omega,J*omega)+M);
dp=domega(1); dq=domega(2); dr=domega(3);

```

```
OMEGA=(1/2)*[0 -p -q -r;p 0 r -q; q -r 0 p;r q -p 0];  
dquat=OMEGA*[n0;n1;n2;n3];  
dn0=dquat(1);dn1=dquat(2);dn2=dquat(3);dn3=dquat(4);  
  
dpos=R*[u;v;w];  
dpn=dpos(1);dpe=dpos(2);dh=-dpos(3);  
  
Y=du^2+dv^2+dw^2+dp^2+dq^2+dr^2+dh^2  
end
```