

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA AEROESPACIAL

Kenedy Matiasso Portella

**DESENVOLVIMENTO E APLICAÇÃO DE UMA BIBLIOTECA PYTHON
PARA MODELAGEM, ANÁLISE E SIMULAÇÃO DE AERONAVES**

Santa Maria, RS
2022

Kenedy Matiasso Portella

**DESENVOLVIMENTO E APLICAÇÃO DE UMA BIBLIOTECA PYTHON PARA
MODELAGEM, ANÁLISE E SIMULAÇÃO DE AERONAVES**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeroespacial, Área de Concentração em Sistemas Dinâmicos e Controle, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Aeroespacial**. Defesa realizada por videoconferência.

ORIENTADOR: Prof. Marcelo Serrano Zanetti

COORIENTADOR: Prof. André Luis da Silva

Santa Maria, RS
2022

Kenedy Matiasso Portella

**DESENVOLVIMENTO E APLICAÇÃO DE UMA BIBLIOTECA PYTHON PARA
MODELAGEM, ANÁLISE E SIMULAÇÃO DE AERONAVES**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Aeroespacial, Área de Concentração em Sistemas Dinâmicos e Controle, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Engenharia Aeroespacial**.

Aprovado em 11 de janeiro de 2022:

Marcelo Serrano Zanetti, Dr. (UFSM)
(Presidente/Orientador)

André Luis da Silva, Dr. (UFSM)
(Coorientador)

Lucas Vizzotto Bellinaso, Dr. (UFSM)

Santa Maria, RS
2022

DEDICATÓRIA

À memória de meu avô, Zeonilde Bordin Matiasso. Eu sei que você estaria orgulhoso.

AGRADECIMENTOS

Primeiramente, agradeço à minha família, em especial à minha mãe, Lorizete Mattoso, por seu apoio contínuo ao longo de minha jornada. Esse caminho teria sido muito mais difícil –quase impossível– sem vocês ao meu lado.

Agradeço ao meu amigo e mentor, Prof. Pedro Paglione, por ter acreditado incondicionalmente na minha capacidade, por vezes, quando nem eu acreditava. O senhor é uma inspiração para todos ao seu redor, e é uma grande honra ter sido seu aluno.

Agradeço aos Professores: Marcelo Serrano Zanetti, Lucas Vizzotto Bellinaso e André Luis da Silva por todos os ensinamentos, orientações e contribuições, não só nesse trabalho, mas ao longo de toda minha vida acadêmica.

Agradeço à minha namorada, Júlia de Mattos, por seu companheirismo, carinho e compreensividade. Nunca imaginei que braços tão pequenos fossem capazes de um abraço tão aconchegante.

Agradeço meu supervisor de estágio, Frederico Capuzzo, por abrir as portas da Siemens A.G. para mim, me dando assim a privacidade de trabalhar ao lado de algumas das pessoas mais talentosas que já conheci.

Agradeço aos todos os Professores por seu empenho e dedicação na missão de transformar cada um de nós em excelentes profissionais.

Emfim, agradeço a todos que, direta ou indiretamente, contribuíram na minha vida acadêmica, profissional ou pessoal.

Don't Panic!

(The Hitchhiker's Guide to the Galaxy)

RESUMO

DESENVOLVIMENTO E APLICAÇÃO DE UMA BIBLIOTECA PYTHON PARA MODELAGEM, ANÁLISE E SIMULAÇÃO DE AERONAVES

AUTOR: Kenedy Matiasso Portella
ORIENTADOR: Marcelo Serrano Zanetti
COORIENTADOR: André Luis da Silva

Nos últimos anos, o uso de modelos computacionais se tornou parte indispensável do projeto de aeronaves, graças a redução de custos, tempo e riscos propiciada. Sua adoção permite a predição de características da aeronave antes mesmo da construção de um protótipo, assim, diversos parâmetros de desempenho podem ser otimizados em fases preliminares de projeto, quando os custos associados são menores. Contudo, os *softwares* comerciais que cumprem esse propósito apresentam algumas limitações, principalmente ligadas ao custo de aquisição, compatibilidade com outras ferramentas e flexibilidade dos modelos. Considerando isso, esse trabalho apresenta o desenvolvimento de uma biblioteca de código aberto para modelagem, análise e simulação de aeronaves. A principal contribuição é a estruturação de uma plataforma modular onde cada um dos blocos construtivos pode ser adequado ao nível de detalhamento requerido pelo usuário. Além disso, diversos desses blocos são pré-implementados e podem ser adaptados a usos específicos. Em soma, a biblioteca também dispõe de diversas ferramentas que automatizam ou auxiliam no processo de geração e análise de características ligadas ao desempenho da aeronave. Finalmente, são apresentadas duas aplicações práticas do uso da biblioteca, o primeiro tratando de uma aeronave de asa fixa de grande porte, enquanto o segundo trata de uma aeronave não tripulada com capacidade de pouso e decolagem vertical. Para o primeiro caso, é apresentada a metodologia de uso dos modelos e métodos da biblioteca, o que inclui a construção do sistema, linearização, cálculo do balaço, separação modal, análise de qualidades de voo e simulações em malha aberta e malha fechada. O segundo exemplo explora a capacidade de modelagem de aeronaves com arquitetura não convencional, bem como o uso desses modelos em simulações fora da biblioteca.

Palavras-chave: Aeronaves. Simulação. Modelagem. Controle. Desempenho. Mecânica de voo. Código aberto. Python.

ABSTRACT

DEVELOPMENT AND APPLICATION OF AN OPEN-SOURCE SOFTWARE FOR AIRCRAFT MODELING, ANALYSIS AND SIMULATION

AUTHOR: Kenedy Matiasso Portella

ADVISOR: Marcelo Serrano Zanetti

CO-ADVISOR: André Luis da Silva

In recent years, the use of computer models has become an indispensable part of aircraft design, thanks to reducing of costs, time and risks provided. Its adoption allows the prediction of aircraft characteristics even before the manufacturing of a prototype, thus, several performance parameters can be optimized in preliminary design phases, when the costs associated are lower. However, the available softwares that fulfills this purpose have some limitations, mainly related to the acquisition cost, compatibility with other tools and flexibility of the models. Considering this, this work presents the development of an open source library for aircraft modeling, analysis and simulation. The main contribution is the structuring of a modular platform where each of the building blocks can be implemented to the level of detail required by the user. In addition, many of these blocks are pre-implemented and can be adapted to specific uses. In sum, the library also has several tools to automate or assist in the process of generating and analyzing characteristics related to aircraft performance. Finally, two practical applications are presented, the first one dealing with a large fixed-wing aircraft, while the second one with an unmanned aircraft with vertical take-off and landing capabilities. For the first case, the methodology for using the models and methods from the library is presented, including model implementation, linearization, trimming, flight quality analysis and open- and closed-loop simulations. The second example explores the library capabilities related to modeling unconventional architecture aircraft, as well as the use of these models in simulations.

Keywords: Aircraft. Simulation. Modeling. Control. Performance. Flight mechanics. Open-source software. Python.

LISTA DE FIGURAS

Figura 2.1 – Sistema de referência inercial e sistema do corpo.	21
Figura 2.2 – Transformação do sistema \mathcal{I} para o sistema \mathcal{B}	22
Figura 2.3 – Sistema de referência aerodinâmico e sistema do corpo.	22
Figura 2.4 – Sistema de referência propulsivo e sistema do corpo.	23
Figura 3.1 – Relação entre a escala <i>Cooper-Harper</i> e os níveis de qualidade de voo.	33
Figura 4.1 – Fluxograma de uso da biblioteca PyAAT.	36
Figura 4.2 – Blocos construtivos e fluxo de informação da biblioteca PyAAT.	37
Figura 5.1 – Variáveis de controle da aeronave de referência.	41
Figura 5.2 – Ações de controle na aeronave em malha aberta.	48
Figura 5.3 – Atitude e velocidades angulares da aeronave em malha aberta.	49
Figura 5.4 – Posição e velocidades linear da aeronave em malha aberta.	49
Figura 5.5 – Qualidade de voo do modo longitudinal de período curto.	53
Figura 5.6 – Qualidade de voo do modo longitudinal de período longo.	54
Figura 5.7 – Qualidade de voo do modo <i>Dutch-roll</i>	54
Figura 5.8 – Qualidades de voo do modo de rolamento puro.	55
Figura 5.9 – Qualidade de voo do modo espiral.	55
Figura 5.10 – Malha de sistema de aumento de estabilidade longitudinal.	56
Figura 5.11 – Qualidades de voo do modo de período longo em malha fechada.	59
Figura 5.12 – Qualidades de voo do modo de período curto em malha fechada.	59
Figura 5.13 – Rajadas de vento aplicadas na simulação não linear.	60
Figura 5.14 – Posição do profundor da aeronave com SAS ligado e desligado.	60
Figura 5.15 – Comportamento dos estados da dinâmica longitudinal com SAS ligado e desligado.	61
Figura 5.16 – Malha do sistema de aumento de estabilidade látero-direcional.	61
Figura 5.17 – Qualidade de voo do modo <i>Dutch-roll</i> do sistema em malha fechada. ..	63
Figura 5.18 – Deflexões de aileron e leme durante um voo em curva sujeito a perturbações.	63
Figura 5.19 – Comportamento das variáveis angulares durante um voo em curva sujeito a perturbações.	64
Figura 5.20 – Trajetória da aeronave com SAS ligado e desligado durante uma manobra de curva coordenada.	65
Figura 5.21 – Malha de rastreamento de ângulo de arfagem.	65
Figura 5.22 – Ângulo e velocidade de arfagem para um voo de subida com PA de arfagem.	66
Figura 5.23 – Deflexão do profundor durante rastreo do ângulo de arfagem.	67
Figura 5.24 – Trajetória da aeronave em um voo de subida com rastreo de arfagem. .	67
Figura 5.25 – Envelope de voo da aeronave de asa fixa.	68
Figura 6.1 – Variáveis de controle da aeronave não tripulada.	69
Figura 6.2 – SAS do tipo MIMO da aeronave não tripulada.	73
Figura 6.3 – Trajetória da aeronave não tripulada com e sem SAS.	74
Figura 6.4 – Ações de controle da aeronave não tripulada com e sem SAS.	74
Figura A.1 – Modelos atmosféricos presentes na biblioteca PyAAT.	81
Figura A.2 – Modelos de rajada de acordo com a norma FAR25341-1.	84
Figura B.1 – Critério da impressão digital para o modo de período curto.	87

LISTA DE TABELAS

Tabela 3.1 – Informações sobre os integradores disponíveis da biblioteca PyAAT.	35
Tabela 5.1 – Nomes atribuídos aos estados.	45
Tabela 5.2 – Nomes atribuídos às derivadas dos estados.	45
Tabela 5.3 – Conjunto de variáveis para cálculo de equilíbrio.	46
Tabela 5.4 – Funções de separação modal presentes na biblioteca PyAAT.	51
Tabela 5.5 – Comparação entre autovalores da matriz completa e matrizes de ordem reduzida.	52
Tabela 5.6 – Funções para análise de qualidade de voo presentes na biblioteca PyAAT.	53
Tabela 5.7 – Autovalores da dinâmica longitudinal antes e após inserção de SAS longitudinal.	58
Tabela 5.8 – Autovalores da dinâmica látero-direcional antes e após inserção do SAS látero-direcional.	62
Tabela 6.1 – Estados para balanço em hover e manobra genérica.	71
Tabela 6.2 – Controles para balanço em hover e manobra genérica.	71
Tabela A.1 – Derivadas de estabilidade da aeronave conceitual.	82
Tabela B.1 – Qualidades de voo do modo de período longo.	88
Tabela B.2 – Qualidades de voo do modo de rolamento puro.	88
Tabela B.3 – Qualidades de voo do modo espiral.	88
Tabela B.4 – Qualidades de voo do modo <i>Dutch-roll</i>	89
Tabela C.1 – Parâmetros globais da aeronave.	90
Tabela C.2 – Derivadas de estabilidade e controle da aeronave de asa fixa de grande porte.	90
Tabela D.1 – Parâmetros globais da aeronave.	91
Tabela D.2 – Derivadas de estabilidade e controle da aeronave não tripulada conceitual.	91

LISTA DE ABREVIATURAS E SIGLAS

<i>6GDL</i>	Seis graus de liberdade
<i>PyAAT</i>	<i>Python Aerospace Analysis Toolbox</i>
<i>ISA</i>	<i>International Standard Atmosphere</i>
<i>COESA</i>	<i>Committee on Extension to the Standard Atmosphere</i>
<i>NED</i>	<i>Nort-East-Down</i>
<i>SAS</i>	Sistema de Aumento de Estabilidade (<i>Stability Augmentation System</i>)
<i>PA</i>	Piloto Automático
<i>EDs</i>	Equações Diferenciais
<i>RK45</i>	Método Runge-Kuta de quarta ordem
<i>BDF</i>	<i>Backward Differentiation Formula</i>
<i>LQR</i>	Regulador Linear Quadrático
<i>CG</i>	Centro de gravidade
<i>SISO</i>	<i>Single Input Single Output</i>
<i>MIMO</i>	<i>Multiple Input Multiple Output</i>

LISTA DE SÍMBOLOS

$(\cdot) _{\square}$	Variável escrita no sistema de referências \square
$(\cdot)_a$	Variável aumentada
$(\cdot)_{dr}$	Variável referente ao modo de oscilação <i>Dutch-roll</i>
$(\cdot)_e$	Valor da variável na condição de equilíbrio
$(\cdot)_i$	Valor da variável em uma condição de referência especificada
$(\cdot)_{ph}$	Variável referente ao modo de oscilação de período longo
$(\cdot)_{ref}$	Valor de referência da variável
$(\cdot)_r$	Variável referente ao modo de oscilação de rolamento puro
$(\cdot)_s$	Variável referente ao modo de oscilação espiral
$(\cdot)_{sp}$	Variável referente ao modo de oscilação de período curto
A	Matriz de estados do sistema linearizado
B	Matriz de entradas do sistema linearizado
b	Envergadura da asa
$C_{\square\triangle}$	Matriz de transformação do sistema \square para o sistema \triangle
C	Matriz de saídas do sistema linearizado
\bar{c}	Corda média aerodinâmica
C_D	Coeficiente de arrasto
c_d	Coeficiente de torque
C_L	Coeficiente de sustentação
C_l	Coeficiente de momento em torno do eixo \hat{a}_x do sistema aerodinâmico
C_m	Coeficiente de momento em torno do eixo \hat{a}_y do sistema aerodinâmico
C_n	Coeficiente de momento em torno do eixo \hat{a}_z do sistema aerodinâmico
c_t	Coeficiente de tração
C_Y	Coeficiente de força látero-direcional
D	Força de arrasto
\vec{F}	Vetor de forças
\vec{F}_g	Vetor de forças gravitacionais

\vec{F}_a	Vetor de forças aerodinâmicas
\vec{F}_p	Vetor de forças propulsivas
\vec{F}_p	Vetor de momentos
g	Módulo da aceleração gravitacional
H_w	Comprimento de rajada de vento
\mathbf{I}	Matriz de inércias
L	Força de sustentação
\vec{M}	Vetor de momentos
\vec{M}_a	Vetor de momentos aerodinâmicos
\vec{M}_p	Vetor de momentos propulsivos
p	Velocidade angular da aeronave em torno do eixo \hat{b}_x do sistema do corpo
Q	Módulo torque de um propulsor apontando na direção \hat{p}_x do sistema propulsivo
q	Velocidade angular da aeronave em torno do eixo \hat{b}_y do sistema do corpo
r	Velocidade angular da aeronave em torno do eixo \hat{b}_z do sistema do corpo
S	Área em planta da asa
T_{\square}	Constante de tempo do modo de oscilação \square
T	Módulo da tração de um propulsor apontando na direção \hat{p}_x do sistema propulsivo
T_K	Temperatura do ar
\vec{U}	Vetor de controles
u	Velocidade linear da aeronave na direção na direção \hat{b}_x do sistema do corpo
\vec{V}	Vetor de velocidades da aeronave
v	Velocidade linear da aeronave na direção na direção \hat{b}_y do sistema do corpo
V_o	Velocidade máxima do vento
V_w	Velocidade vertical do vento
\vec{V}_{wind}	Vetor de velocidades do vento
V_{inf}	Velocidade do escoamento não perturbado
w	Velocidade linear da aeronave na direção na direção \hat{b}_z do sistema do corpo

\vec{X}	Vetor de estado
x_o	Posição cartesiana da aeronave na direção \hat{n}_x do sistema inercial
x_{ow}	Coordenada x de início da rajada de vento
x_{fw}	Coordenada x de fim da rajada de vento
y_o	Posição cartesiana da aeronave na direção \hat{n}_y do sistema inercial
y_{ow}	Coordenada y de início da rajada de vento
y_{fw}	Coordenada y de fim da rajada de vento
z_o	Posição cartesiana da aeronave na direção \hat{n}_z do sistema inercial
α	Ângulo de ataque
β	ângulo de derrapagem
ϵ	Perturbação em torno da condição de equilíbrio
ϕ	Ângulo de rolamento
ω_{\square}	Frequência natural não amortecida do modo de oscilação \square
$\vec{\omega}$	Vetor velocidade angular da aeronave
ψ	Ângulo de guinada
ρ	Densidade do ar
θ	Ângulo de arfagem
τ_{\square}	Constante de tempo do componente \square
ζ_{\square}	Razão de amortecimento do modo de oscilação \square

SUMÁRIO

1	INTRODUÇÃO	16
1.1	MOTIVAÇÃO	17
1.2	OBJETIVOS	18
1.3	ORGANIZAÇÃO DO TRABALHO	19
2	MODELAGEM DA DINÂMICA DE VOO	20
2.1	SISTEMAS DE REFERÊNCIA.....	20
2.2	EQUAÇÕES DO MOVIMENTO DE UM CORPO RÍGIDO	23
2.2.1	Cinemática de translação	24
2.2.2	Cinemática de rotação	24
2.2.3	Dinâmica de translação	25
2.2.4	Dinâmica de rotação	25
2.3	DINÂMICA AUMENTADA	26
2.4	MODELO AERODINÂMICO	26
2.5	MODELOS PROPULSIVOS	28
2.5.1	Motor a jato	28
2.5.2	Pequenos propulsores a hélice	28
3	FUNÇÕES DA BIBLIOTECA	30
3.1	CÁLCULO DO BALANÇO	30
3.2	LINEARIZAÇÃO	31
3.3	SEPARAÇÃO MODAL.....	32
3.4	QUALIDADES DE VOO	33
3.5	SIMULAÇÃO NÃO LINEAR	34
3.6	ENVELOPE DE VOO.....	35
4	PYTHON AEROSPACE ANALYSIS TOOLBOX	36
4.1	DESCRIÇÃO DOS SUBMODELOS	38
4.2	DESCRIÇÃO DOS MÉTODOS	39
5	APLICAÇÃO EM AERONAVE COMERCIAL DE GRANDE PORTE.....	41
5.1	CONSTRUÇÃO DO SISTEMA	41
5.1.1	Criação do bloco aeronave	42
5.2	CRIAÇÃO DO BLOCO PROPULSIVO	43
5.3	INTEGRAÇÃO DOS SUBMODELOS	44
5.4	CÁLCULO DO BALANÇO	44
5.4.1	Condições de balanço pré-implementadas	46
5.5	SIMULAÇÕES EM MALHA ABERTA.....	47
5.6	MODELOS LINEARES.....	50
5.7	QUALIDADES DE VOO DA AERONAVE EM MALHA ABERTA	52
5.8	SISTEMA DE AUMENTO DE ESTABILIDADE LONGITUDINAL	54
5.8.1	Controladores em malha fechada	57
5.9	SISTEMA DE AUMENTO DE ESTABILIDADE LÁTERO-DIRECIONAL.....	61
5.10	PILOTO AUTOMÁTICO DE ÂNGULO DE ARFAGEM.....	64
5.11	ENVELOPE DE VOO.....	68
6	AERONAVE DO TIPO ASA VOADORA COM CAPACIDADE DE POUSO E DECOLAGEM VERTICAL	69
6.1	CRIAÇÃO DO SISTEMA	70
6.2	CÁLCULO DO BALANÇO	71

6.3	SISTEMA DE AUMENTO DE ESTABILIDADE	72
7	CONCLUSÕES.....	76
7.1	TRABALHOS FUTUROS	77
	REFERÊNCIAS BIBLIOGRÁFICAS	78
	APÊNDICE A – MODELOS MATEMÁTICOS	80
	APÊNDICE B – CRITÉRIOS DE QUALIDADE DE VOO.....	87
	APÊNDICE C – DADOS DA AERONAVE DE ASA FIXA DE GRANDE PORTE .	90
	APÊNDICE D – DADOS DA AERONAVE NÃO TRIPULADA COM CAPACI- DADE DE POUSO E DECOLAGEM VERTICAL	91
	APÊNDICE E – LINK PAR CÓDIGOS E EXEMPLOS	92

1 INTRODUÇÃO

Projetos aeronáuticos envolvem análises de sistemas complexos e interdisciplinares. Antes de se obter uma aeronave funcional, centenas de componentes e subsistemas devem ser validados individualmente, o que torna o processo custoso e demorado. De modo geral, cada subsistema é especialmente desenvolvido para uma aeronave – ou família de aeronaves – e deve buscar um balanço entre vários parâmetros (incluindo peso, desempenho, custo, entre outros). Contudo, para se obter esse balanço, deve-se conhecer a aeronave completa a priori, o que é impossível sem o conhecimento de cada componente individualmente. Nesse contexto, o uso de modelos se torna indispensável, pois permite a predição de características da aeronave antes que essa seja fabricada.

De modo simplificado, pode-se dizer que modelos são representações matemáticas que aproximam o comportamento de um certo sistema, permitindo que algumas predições sejam feitas sem a necessidade de um protótipo (NEELAMKAVIL, 1987). Como a acurácia dessas predições está fortemente vinculada ao nível de detalhamento e confiabilidade dos modelos aplicados, geralmente busca-se incrementar a complexidade dos modelos ao passo que o projeto avança e os requisitos são refinados. Essa metodologia viabiliza o uso de modelos em todas as fases de desenvolvimento, validação e utilização da aeronave, além de evitar custos computacionais prematuros e permitir que o projeto avance mais rapidamente.

Outra vantagem é que o uso de modelos permite a aplicação de diferentes níveis de detalhamento a cada subsistema, permitindo que estudos dedicados a cada um sejam conduzidos em paralelo. Além disso, podem ser realizados estudos acerca de condições onde a execução de experimentos é inviável, seja por custos, riscos ou limitações.

Em sistemas complexos, o desempenho individual de cada subsistema não necessariamente reflete o do sistema em sua totalidade. De fato, diversas limitações e simbioses só podem ser percebidas quando a aeronave é analisada de forma integral e considerando as condições de operação a que ela será submetida. Nesse sentido, análise de critérios interdisciplinares como: estabilidade, qualidade de voo e desempenho de voo são necessárias, tanto para validação de decisões, quanto para o direcionamento de eventuais mudanças de projeto. Vale destacar que o custo para modificações em fases avançadas de projeto é consideravelmente superior (RAYMER, 1992), o que torna esse tipo de análise importante desde as fases preliminares de projeto.

Em paralelo, destaca-se que diversas características de aeronaves podem ser alteradas artificialmente. Por exemplo, uma aeronave pode possuir naturalmente uma margem de estabilidade relaxada – o que permite economia de combustível graças às menores deflexões de superfícies aerodinâmicas – e altera-la artificialmente através do uso de controladores em malha fechada. Tendo em vista que essa é prática comum em sistemas

aeronáuticos, a análise de aeronaves nessas condições se torna parte indissociável do projeto.

1.1 MOTIVAÇÃO

O uso de ferramentas para modelagem e simulação é uma realidade do setor aeroespacial. Devido a isso, diversos *softwares* comerciais possuem funcionalidades específicas para esse fim, a exemplo dos *softwares* MATLAB/Simulink (MATHWORKS, 2021), Dimola (Dassault Systemes, 2021), Simcenter Amesim (Siemens Industry Software, 2021), entre outros. Essas soluções são extensivamente utilizadas na indústria e academia, e dispõem de funcionalidades que cobrem diversos aspectos do projeto e operação de aeronaves. Contudo, por se tratarem de produtos proprietários, muitas vezes são inacessíveis a certos públicos, como pequenas empresas, estudantes e grupos de competição. Além disso, o usuário por vezes não tem conhecimento das formulações utilizadas nem permissão para modifica-las, o que limita sua aplicabilidade em problemas específicos. Finalmente, muitas empresas não fornecem suporte para comunicação entre seus produtos com os de outras marcas, exigindo que todo o ciclo de desenvolvimento ocorra em uma única plataforma, impedindo o uso de ferramentas otimizadas para determinada aplicação e/ou reaproveitamento de soluções previamente desenvolvidas.

Além de ferramentas proprietárias, existem diversas alternativas de código aberto, como listado no catálogo de *softwares* de domínio público da Agência Espacial Americana (National Aeronautics and Space Administration, 2021). Entretanto, a maioria deles são desenvolvidos em linguagem C/C++ ou FORTRAN e não contam com nenhum tipo interface gráfica, o que dificulta a utilização por usuários iniciantes. Ainda, geralmente são desenvolvidas como plataformas *standalone* e disponibilizadas apenas em formato compilado, o que dificulta a adaptação para usos específicos ou integração com outras soluções.

Diante disso, o desenvolvimento de uma ferramenta de código aberto e implementada em Python mostra-se uma solução aos problemas apresentados. A linguagem Python vem ganhando espaço nas áreas da engenharia nos últimos anos, sendo que, de acordo com um estudo da Universidade de Colorado Berkeley (EXTENSION, 2021), é a segunda mais utilizada no mundo. Ademais, de acordo com o índice PYPL (GITHUB, 2021), Python é a linguagem com a comunidade mais ativa do mundo em 2021, correspondendo a mais de 30% do total de projetos no *github*. Como consequência, já dispõe de diversos *plugins* para comunicação com outros *softwares*. Em soma, o fato de ser uma linguagem aberta e multiparadigma a torna acessível aos mais variados públicos. Finalmente, a ferramenta já possui diversas bibliotecas científicas (e.g. *scipy*, *pandas*, *numpy*) e de engenharia (e.g. *python-control*, *CFDPython*, *SfePy*, *AstroPy*, *PyTornado*), o que facilita a integração entre múltiplas disciplinas.

Um grande desafio da descrição de sistemas aeronáuticos é a integração entre subsistemas, considerando isso, a plataforma proposta é desenvolvida com arquitetura modular e capaz de realizar essa tarefa de forma automática. Para tal, a biblioteca padroniza o formato de entradas e saídas de cada módulo (correspondente a cada subsistema), e efetua verificações para garantir que esses padrões são seguidos. Esse método torna o processo de modelagem mais intuitivo e flexível, visto que o usuário é livre para descrever cada submodelo, desde que seja garantido o formato das interfaces.

Além disso, a biblioteca conta com uma série de modelos nativos, tornando possível a rápida construção de um sistema pronto para simulação. Quando esse é o caso, o usuário precisa apenas definir um conjunto de parâmetros, os quais podem ser obtidos com outras ferramentas ou experimentalmente.

Finalmente, são disponibilizadas uma série de funções que auxiliam na geração e análise de resultados referentes aos modelos. Cada método é construído buscando conciliar simplicidade de uso e flexibilidade. Por exemplo, a função de balanço pode ser utilizada com controle total sobre cada uma das condições de contorno, ou, alternativamente, ser utilizada considerando uma condição de voo padrão, e a biblioteca se encarrega de criar as condições de contorno.

1.2 OBJETIVOS

O objetivo principal do trabalho é a implementação e demonstração de uma biblioteca Python para modelagem, simulação e análise de aeronaves. Para cumprir o objetivo principal, os objetivos específicos são:

- Arquitetar e implementar uma plataforma modular para modelagem e simulação de aeronaves;
- Implementar modelos de atmosfera, gravidade e vento;
- Desenvolver o modelo de mecânica de voo de uma aeronave genérica em seis graus de liberdade (6GDL);
- Implementar modelos propulsivos e aerodinâmico;
- Implementar funções para cálculo de balanço, linearização, análise de qualidade de voo, construção do envelope de voo e simulação não linear da aeronave;
- Implementar modelos de atuadores, sensores, filtros e controladores;
- Validar e demonstrar a ferramenta em uma aplicação considerando uma aeronave de asa fixa;

- Demonstrar o uso da ferramenta em uma aeronave de arquitetura não convencional.

A modelagem aplicada nesse trabalho é majoritariamente fundamentada nos conceitos apresentados em Cook (2007) e Stevens, Lewis e Johnson (2016), dada a ampla utilização dessas referências nos meios científico e acadêmico. A configuração de asa fixa é a mais utilizada na aviação atual, por esta razão, tanto as bibliografias quanto os métodos implementados nesse trabalho dão ênfase a essa arquitetura, contudo, extensões são realizadas com o intuito de estender a aplicabilidade para aeronaves de asa rotativa.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em 7 capítulos. Nos **capítulos 2 e 3** é apresentada a fundamentação teórica referente aos modelos e métodos implementados na biblioteca. O **capítulo 4** apresenta uma descrição geral da ferramenta desenvolvida. O **capítulo 5** é dedicado a uma aplicação prática da biblioteca no estudo de uma aeronave de asa fixa, a qual tem por objetivo a validação da biblioteca e demonstração de seu uso em uma aeronave de configuração canônica. Esse capítulo é escrito na forma de manual, onde é apresentada a implementação (código) acompanhada de comentários e informações referentes a cada método e/ou modelo aplicado. Além disso, também são incluídos comentários e interpretações acerca dos resultados gerados. O **capítulo 6** considera uma segunda aplicação, dessa vez tratando de uma aeronave do tipo asa voadora com capacidade de pouso e decolagem vertical. Essa etapa explora as capacidades da biblioteca quanto à modelagem de aeronaves de asa rotativa e demonstra o uso de um modelo gerado pela biblioteca em simulações fora dela. Finalmente, o **capítulo 7** conclui esse trabalho.

2 MODELAGEM DA DINÂMICA DE VOO

A modelagem da dinâmica de voo de uma aeronave é composta por dois blocos fundamentais: O modelo ambiental, que descreve o meio no qual a aeronave se desloca, e o modelo da aeronave em si, que descreve o comportamento dinâmico da mesma e como essa interage com o ambiente. A descrição de ambos é bem estabelecida na literatura, especialmente para os casos de aeronaves de asa fixa e corpo rígido (caso considerado nesse capítulo). A abordagem apresentada nesse capítulo é majoritariamente baseada em Cook (2007), Tewari (2007) e Stevens, Lewis e Johnson (2016).

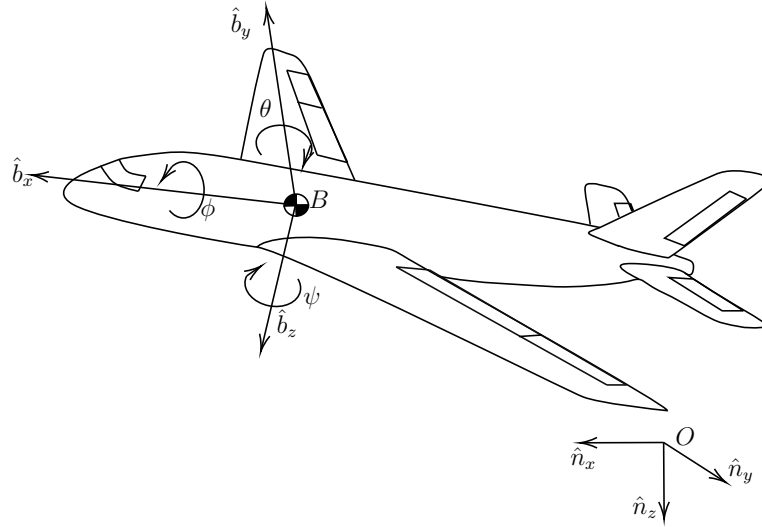
2.1 SISTEMAS DE REFERÊNCIA

Movimentos são relativos, assim sendo, a descrição do movimento de um corpo só é possível com a adoção de sistemas de referência. A descrição do movimento de corpos rígidos requer no mínimo dois sistemas independentes, contudo, por questões práticas, geralmente são utilizados sistemas complementares. Nesse trabalho são adotados 4 sistemas de referência, sendo um sistema inercial (aceleração absoluta desprezível na região de estudo) e outros 3 sistemas móveis. Todos os sistemas adotados são compostos por uma tríade de eixos cartesianos e ortogonais entre si. Assim têm-se:

- O sistema de coordenadas inercial $\mathcal{I} : \{O, \hat{n}_x, \hat{n}_y, \hat{n}_z\}$, com origem O fixa na superfície da terra, \hat{n}_x apontando na direção norte, \hat{n}_y apontando na direção leste e \hat{n}_z apontando para baixo. Essa definição é conhecida como NED (*north, east, down*) e é utilizada tanto para descrever movimento de aeronaves quanto espaçonaves. No contexto da mecânica de voo, tem-se interesse em pequenos intervalos de tempo e pequenos deslocamentos, de modo que a curvatura da Terra pode ser desprezada e a aproximação de Terra Plana passa a ser válida. Nesse caso, deslocamentos no sistema NED são diretamente expressos como deslocamentos cartesianos x_o , y_o e z_o .
- Um sistemas de coordenadas móvel $\mathcal{B} : \{B, \hat{b}_x, \hat{b}_y, \hat{b}_z\}$, com origem B fixa no centro de massa da aeronave, sendo que \hat{b}_x aponta para frente (na direção do nariz da aeronave), \hat{b}_y aponta na direção da semi-asa direita e \hat{b}_z aponta na direção da barriga da aeronave. É convenção chamar \mathcal{B} de sistema do corpo, uma vez que esse sistema fica rigidamente fixo à aeronave, tando durante a translação quanto a rotação. A Figura 2.1 apresenta o sistema inercial e sistema do corpo.

O apontamento do sistema do corpo em relação ao sistema inercial é representado por uma matriz de atitude $C_B^{\mathcal{I}}$, a qual é construída por três rotações elementares sucessivas

Figura 2.1 – Sistema de referência inercial e sistema do corpo.



Fonte: Autor

em torno de eixos não consecutivos. Essa matriz é lida como: "Matriz de transformação do sistema \mathcal{I} para o sistema \mathcal{B} ", dado que a transformação de um vetor qualquer \vec{V} representado no sistema \mathcal{I} para o sistema \mathcal{B} é dada pela simples multiplicação por essa matriz. Como apresentado na Equação 2.1.

$$\vec{V}|_{\mathcal{B}} = C_{\mathcal{B}}^{\mathcal{I}} \vec{V}|_{\mathcal{I}} \quad (2.1)$$

Similarmente, a conversão de um vetor representado no sistema \mathcal{B} para o sistema \mathcal{I} se dá multiplicando o vetor pela inversa de $C_{\mathcal{B}}^{\mathcal{I}}$. Contudo, $C_{\mathcal{B}}^{\mathcal{I}}$ possui a propriedade de ortogonalidade, assim sendo, a inversa é igual a transposta, gerando o resultado apresentado na Equação 2.2.

$$\vec{V}|_{\mathcal{I}} = C_{\mathcal{I}}^{\mathcal{B}} \vec{V}|_{\mathcal{B}} = C_{\mathcal{B}}^{\mathcal{I}T} \vec{V}|_{\mathcal{B}} \quad (2.2)$$

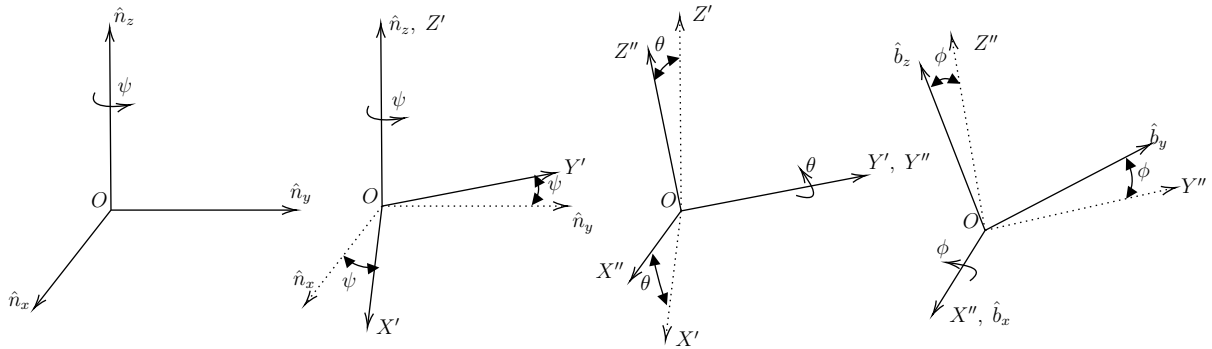
Em aeronáutica, a sequência de rotações mais utilizada é a sequencia 3-2-1, pelas quantidades ψ , θ e ϕ (guinada, arfagem e rolamento), como representado na Figura 2.2. Nota-se que a primeira rotação ocorre em torno de um eixo no sistema inercial \hat{n}_z , enquanto a ultima rotação ocorre em torno do eixo \hat{b}_x .

Adotando-se a sequência de rotações descrita, a matriz de transformação toma a forma apresentada na Equação 2.3. Mais detalhes sobre a obtenção são apresentados no Apêndice A.5.

$$C_{\mathcal{B}}^{\mathcal{I}} = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ \sin \phi \cos \psi \sin \theta - \cos \theta \sin \psi & \sin \phi \sin \psi \sin \theta + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi & \cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.3)$$

Os demais sistemas são definidos com respeito ao sistema \mathcal{B} . Assim, tem-se:

Figura 2.2 – Transformação do sistema \mathcal{I} para o sistema \mathcal{B} .

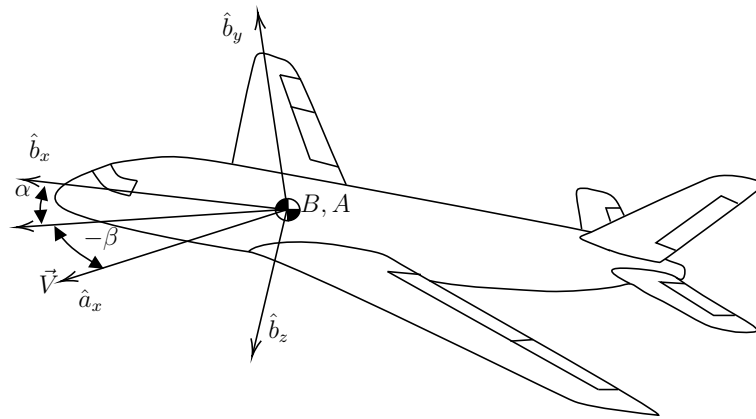


Fonte: Autor

- Um sistema de coordenadas móvel $\mathcal{A} : \{B, \hat{a}_x, \hat{a}_y, \hat{a}_z\}$, chamado de sistema aerodinâmico (ou sistema do vento). O qual possui origem no centro de massa da aeronave, \hat{a}_x aponta na direção da velocidade aerodinâmica, \hat{a}_y aponta para o plano da semi-asa direita, com um desvio de um ângulo β em relação a \hat{b}_y e, por fim, \hat{a}_z completa a tríade de apontamentos.

Esse sistema é bastante útil por ser onde as forças e momentos aerodinâmicos são expressos, além de estar relacionado às medidas do sistema de anemometria. A Figura 2.3 ilustra os sistemas \mathcal{A} e \mathcal{B} .

Figura 2.3 – Sistema de referência aerodinâmico e sistema do corpo.



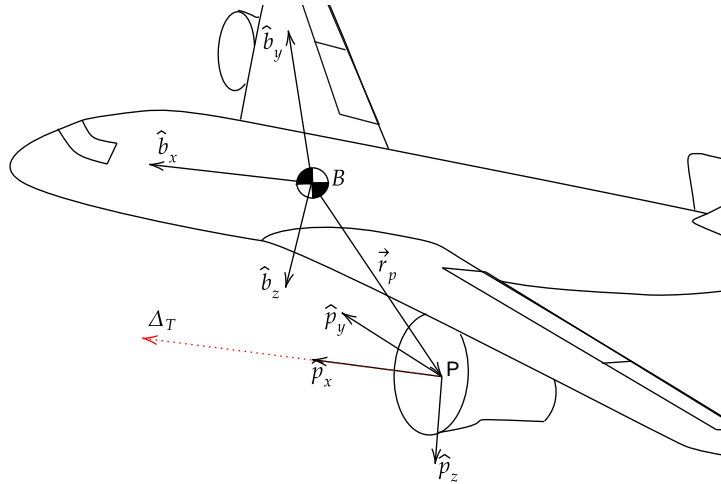
Fonte: Autor

Novamente, esses estão relacionados por uma matriz de rotação, sendo a conversão de um vetor representado no sistema aerodinâmico para o sistema do corpo dada pela matriz C_B^A , apresentada na Equação 2.4.

$$C_B^A = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \quad (2.4)$$

- Finalmente, tem-se o sistema propulsivo $\mathcal{P} : \{P, \hat{p}_x, \hat{p}_y, \hat{p}_z\}$, o qual expressa a orientação de um vetor propulsão em relação ao sistema do corpo, como apresentado na Figura 2.4. Esse sistema tem origem no centro de tração do propulsor, \hat{p}_x aponta na direção da linha de tração, \hat{p}_y aponta para o plano da semi-asa direita e, por fim, \hat{p}_z completa a tríade de apontamentos do sistema dextrogiro.

Figura 2.4 – Sistema de referência propulsivo e sistema do corpo.



Fonte: Autor

Cada propulsor possui seu próprio sistema de referências, o qual é orientado de forma independente dos demais. Em aeronaves convencionais, no entanto, é comum a definição de um único sistema com origem no centro de empuxo da aeronave e com eixo \hat{p}_x apontando na direção do empuxo líquido gerado pela combinação dos propulsores. Essa simplificação é equivalente a assumir que todos os propulsores possuem apontamento e posição fixas e são controlados proporcionalmente.

A transformação de um vetor no sistema propulsivo para o sistema do corpo se dá através das rotações ψ_p e θ_p , na sequência 3-2. De forma matemática, é dada pela multiplicação pela matriz C_B^P , apresentada na Equação 2.5.

$$C_B^P = \begin{bmatrix} \cos(\psi_p) \cos(\theta_p) & \sin(\psi_p) \cos(\theta_p) & -\sin(\theta_p) \\ -\sin(\psi_p) & \cos(\psi_p) & 0 \\ \sin(\theta_p) \cos(\psi_p) & \sin(\psi_p) \sin(\theta_p) & \cos(\theta_p) \end{bmatrix} \quad (2.5)$$

2.2 EQUAÇÕES DO MOVIMENTO DE UM CORPO RÍGIDO

Equações do movimento são expressões matemáticas que descrevem o comportamento espacial da aeronave em função do tempo e dos estados. Um conjunto mínimo

composto por 12 equações diferenciais ordinárias de primeira ordem é necessário para a descrição do movimento de um corpo rígido em 6 graus de liberdade. Cada uma dessas equações possui sentido físico podendo ser classificada em cinemática (translação e rotação) ou dinâmica (translação e rotação). Essas equações são acopladas e geralmente não dispõem de uma solução analítica, sendo necessário recorrer a métodos numéricos para obtenção da resposta temporal. As equações podem ser compactadas de forma vetorial em um vetor de estado \vec{X} , organizado como apresentado na Equação 2.6.

$$\vec{X} = [x_o \ y_o \ z_o \ u \ v \ w \ \phi \ \theta \ \psi \ p \ q \ r]^T \quad (2.6)$$

2.2.1 Cinemática de translação

A cinemática de translação relaciona a velocidade linear do centro de gravidade (CG) da aeronave em relação ao referencial inercial representada no sistema inercial $\vec{V}|_{\mathcal{I}} = [\dot{x}_o \ \dot{y}_o \ \dot{z}_o]^T$ às velocidades lineares da aeronave no sistema do corpo $\vec{V}|_{\mathcal{B}} = [u \ v \ w]^T$, como descrito pela Equação 2.7.

$$\begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{z}_o \end{bmatrix} = C_{\mathcal{I}}^{\mathcal{B}} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.7)$$

2.2.2 Cinemática de rotação

A cinemática de atitude relaciona a variação dos parâmetros de atitude ($\dot{\psi}$, $\dot{\theta}$ e $\dot{\phi}$) com o vetor velocidade angular do corpo com respeito ao sistema inercial escrito no sistema do corpo $\vec{\omega}|_{\mathcal{B}} = [p \ q \ r]^T$. Nesse caso, as velocidades p , q e r estão escritas no sistema do corpo, enquanto a taxa de variação dos ângulos de Euler são expressas no referencial em que a rotação instantânea ocorre, como ilustrado na Figura 2.2 para o caso rotações na sequência 3-2-1. Nesse contexto, a conversão de sistemas deve ser realizada considerando cada um dos sistemas intermediários, como apresentado na Equação 2.8.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = C_1(\phi)C_2(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + C_1(\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (2.8)$$

onde $C_1(\phi)$ e $C_2(\theta)$ são rotações elementares em torno dos eixos Z'' e Z' da Figura 2.2, respectivamente, como apresentado no Apêndice A.5. Efetuando a multiplicação matricial,

obtém-se a Equação 2.9.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.9)$$

Finalmente, isolando em termos das derivadas dos ângulos de Euler, tem-se a Equação 2.10.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.10)$$

2.2.3 Dinâmica de translação

A dinâmica de translação relaciona as acelerações lineares do CG às forças \vec{F} aplicadas sobre este. Aplicando a Segunda Lei de Newton para referenciais girantes, obtém-se a Equação 2.11.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \sum \vec{F} \Big|_{\mathcal{B}} - \vec{\omega} \times \vec{V} \quad (2.11)$$

onde as forças são expressas no sistema do corpo e m é a massa do corpo.

Essas forças podem ser de três naturezas: gravitacional \vec{F}_g , propulsiva \vec{F}_p ou aerodinâmica \vec{F}_a . Como cada uma está expressa em um sistema distinto, elas devem ser convertidas para o sistema \mathcal{B} antes da aplicação na Equação 2.11, como apresentado na Equação 2.12.

$$\sum \vec{F} \Big|_{\mathcal{B}} = C_{\mathcal{B}}^{\mathcal{A}} \vec{F}_a \Big|_{\mathcal{A}} + C_{\mathcal{B}}^{\mathcal{P}} \vec{F}_p \Big|_{\mathcal{P}} + C_{\mathcal{B}}^{\mathcal{I}} \vec{F}_g \Big|_{\mathcal{I}} \quad (2.12)$$

2.2.4 Dinâmica de rotação

A dinâmica de rotação relaciona as acelerações angulares aos momentos aplicados em torno do seu CG. Aplicando a Segunda Lei de Newton para rotações, obtemos a Equação 2.13.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}^{-1} \left(\sum \vec{M} \Big|_{\mathcal{B}} - \vec{\omega} \times \mathbf{I} \vec{\omega} \right) \quad (2.13)$$

onde \mathbf{I} é o tensor de inércias, descrito pela Equação 2.14.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (2.14)$$

Os momentos podem ser de origem propulsiva \vec{M}_p ou aerodinâmica \vec{M}_a e, assim como as forças, devem ser expressos no referencial \mathcal{B} , como apresentado na Equação 2.15.

$$\sum \vec{M} \Big|_{\mathcal{B}} = C_{\mathcal{B}}^A \vec{M}_a \Big|_{\mathcal{A}} + C_{\mathcal{B}}^P \vec{M}_p \Big|_{\mathcal{P}} \quad (2.15)$$

2.3 DINÂMICA AUMENTADA

Além da dinâmica do movimento em 6GDL descrita na Seção 2.2, outras dinâmicas podem gerar efeitos significativos no comportamento aeronave, como é o caso de atuadores e controladores. Geralmente esses componentes estão associados a equações diferenciais, e, portanto, devem ser incluídos no conjunto de equações a ser integrada pelo sistema. Nesse caso, o vetor de estado \vec{X} se converte em um vetor de estado aumentado \vec{X}_a , como apresentado na Equação 2.16.

$$\vec{X}_a = \left[\vec{X}^T \quad at_1 \quad \dots \quad at_I \quad sns_1 \quad \dots \quad sns_J \quad cnt_1 \quad \dots \quad cnt_K \right]^T \quad (2.16)$$

Na Equação 2.16, at_i é o i -ésima estado relacionado aos atuadores, sns_j é o j -ésimo estado dos sensores, e cnt_k é o k -ésimo relacionado aos controladores.

2.4 MODELO AERODINÂMICO

Os efeitos aerodinâmicos são os principais geradores de forças e momentos do sistema, por isso, um modelo representativo do comportamento real da aeronave é indispensável para a obtenção de resultados coerentes. Por outro lado, modelos sofisticados costumam ser complexos e custosos do ponto de vista computacional, o que inviabiliza certas aplicações.

Uma das técnicas mais utilizadas para o caso de aeronaves de asa fixa é o uso de derivadas de estabilidade e controle. Essa abordagem consiste em uma aproximação linear do comportamento aerodinâmico em torno de um determinado ponto de operação, a qual é compactada em coeficientes aerodinâmicos, os quais são dependentes dos esta-

dos e deflexão das controles. O equacionamento e convenção de sinais segue o padrão desenvolvido em (McDonnell Douglas Astronautics Company, 1999), contudo, são desconsiderados os termos não-estacionários presentes no modelo original. Os coeficiente são dados pelas Equações 2.17 - 2.22.

$$\underline{C_L} = C_{L_0} + C_{L_\alpha} \alpha + \frac{\bar{c}}{2V_\infty} C_{L_q} q + C_{L_{\delta_e}} \delta_e \quad (2.17)$$

$$\underline{C_D} = C_{D_0} + C_{D_\alpha} \alpha + \frac{\bar{c}}{2V_\infty} C_{D_q} q + C_{D_{\delta_e}} \delta_e \quad (2.18)$$

$$\underline{C_Y} = C_{Y_0} + C_{Y_\beta} \beta + \frac{b}{2V_\infty} C_{Y_p} p + \frac{b}{2V_\infty} C_{Y_r} r + C_{Y_{\delta_a}} \delta_a + C_{Y_{\delta_r}} \delta_r \quad (2.19)$$

$$\underline{C_m} = C_{m_0} + C_{m_\alpha} \alpha + \frac{\bar{c}}{2V_\infty} C_{m_q} q + C_{m_{\delta_e}} \delta_e \quad (2.20)$$

$$\underline{C_l} = C_{l_0} + C_{l_\beta} \beta + \frac{b}{2V_\infty} C_{l_p} p + \frac{b}{2V_\infty} C_{l_r} r + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r \quad (2.21)$$

$$\underline{C_n} = C_{n_0} + C_{n_\beta} \beta + \frac{b}{2V_\infty} C_{n_p} p + \frac{b}{2V_\infty} C_{n_r} r + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r \quad (2.22)$$

Nas Equações 2.17 - 2.22, \bar{c} representa a corda média aerodinâmica, V_∞ é a velocidade do escoamento não perturbado e b é envergadura da aeronave. Esses coeficientes são convertidos em forças e momentos através das Equações 2.23 e 2.24.

$$\vec{F}_a|_A = \frac{1}{2} \rho V^2 S \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \quad (2.23)$$

$$\vec{M}_a|_A = \frac{1}{2} \rho V^2 S \begin{bmatrix} bC_l \\ \bar{c}C_m \\ bC_n \end{bmatrix} \quad (2.24)$$

onde V é a velocidade aerodinâmica e S é a área em planta da asa.

Destaca-se que essa abordagem é majoritariamente aplicável em fases de projeto preliminar, sendo que, em fases mais avançadas é comum a adoção de técnicas mais sofisticadas, como métodos aerodinâmicos não-estacionários e dinâmica de fluidos computacional (CFD).

2.5 MODELOS PROPULSIVOS

2.5.1 Motor a jato

De acordo com Sadraey (2017), a tração máxima que um motor turbo-jato ou turbo-fan pode gerar a uma certa altitude pode ser aproximada pela Expressão 2.25,

$$T_{max_h} = T_{max} \left(\frac{\rho}{\rho_i} \right)^{n_\rho} \quad (2.25)$$

onde T_{max} é a tração máxima gerada na altitude de referência, ρ_0 é uma densidade da altitude de referência e n_ρ é um coeficiente empírico que relaciona a perda (ou ganho) de tração com a altitude. Essa expressão pode ser modificada para incluir a relação da tração com a velocidade e posição da manete de combustível, como apresentado na Equação 2.26,

$$T = \delta_t T_{max} \left(\frac{V}{V_i} \right)^{n_V} \left(\frac{\rho}{\rho_i} \right)^{n_\rho} \quad (2.26)$$

onde δ_p é a deflexão da manete de combustível (entre 0 e 1), V_i é a velocidade utilizada no momento obtenção dos coeficientes, e n_V é um coeficiente que modela o comportamento da tração em função da velocidade da aeronave.

O vetor força propulsiva é dado pela Equação 2.27.

$$\vec{F}_p|_{\mathcal{P}} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \quad (2.27)$$

Os momentos propulsivos, por sua vez, dependem da posição de cada um dos motores, como apresentado na Equação 2.28.

$$\vec{M}_p|_{\mathcal{B}} = \vec{r}_p|_{\mathcal{B}} \times \vec{F}_p|_{\mathcal{B}} \quad (2.28)$$

2.5.2 Pequenos propulsores a hélice

A tração T e torque Q gerados por uma hélice dependem de diversos parâmetros, como geometria da hélice, condição de voo e velocidade de rotação. Ainda assim, para hélices com geometria fixa e sujeitas a pequenas variações de altitude, esses valores podem ser aproximados pelas Equações 2.29 e 2.30 (POLIVANOV; SIDORENKO, 2019),

(KUANTAMA; TARCA, 2017), (ISAAC et al., 2021).

$$T = c_t \omega_p^2 \quad (2.29)$$

$$Q = \text{sign}(\omega_p) c_d \omega_p^2 \quad (2.30)$$

onde c_t e c_d são os coeficientes de tração e torque, respectivamente, ω_p é a velocidade de rotação do rotor, em rad/s, e $\text{sign}(\omega_p)$ é o sinal de ω_p .

A vetor força propulsiva é dado pela Equação 2.27, enquanto o momento propulsivo é dado pela Equação 2.31.

$$\vec{M}_p|_B = \vec{r}_p|_B \times \vec{F}_p|_B + C_B^P \begin{bmatrix} -Q \\ 0 \\ 0 \end{bmatrix} \quad (2.31)$$

3 FUNÇÕES DA BIBLIOTECA

3.1 CÁLCULO DO BALANÇO

O cálculo do balanço, também chamado de equilíbrio ou trimagem, permite encontrar a combinação de controles e estados que geram um certo regime de voo desejado. Essas variáveis podem ser divididas em quatro categorias, dependendo da sua função:

- Variáveis fixas: São variáveis e controles com valores definidos, os quais são fornecidos pelo usuário como entrada da função de equilíbrio.
- Variáveis desejadas: São as condições de contorno que definem o regime de voo, sendo o objetivo da função de equilíbrio satisfazer essas condições. É composto por um conjunto de derivadas de estados. Recebem o nome "desejadas" porque não existe uma garantia de que serão completamente satisfeitas.
- Variáveis livres: É o conjunto de estados e controles que permite satisfazer condições de contorno, sendo, portanto, a solução do problema.
- Variáveis ignoráveis: Não geram impacto na condição de voo.

O balanço considera uma condição estacionária, ou seja, há interesse em um determinado ponto de operação. Por esta razão, não há interesse em dinâmicas auxiliares, os quais afetam apenas regime transiente. Qualquer combinação de variáveis pode ser solicitada, porém sem garantia de existência de solução. Nesse sentido, três situações são possíveis:

- Solução única: Quando o número de variáveis livres é o mesmo de equações independentes.
- Múltiplas soluções: Quando existem mais variáveis livres que equações independentes. Nesse caso a solução é a primeira combinação que satisfaz as condições de contorno. Sendo assim, é sensível aos chutes iniciais para variáveis livres.
- Não há solução exata: Por vezes pode não existir uma combinação de estados e controles que geram o regime de voo desejado. Nesse caso, a solução encontrada é aquela que minimiza os desvios das variáveis desejadas em relação ao valor solicitado, como descrito pela Equação 3.1.

$$\min_{u \in U, x \in X} \sum \Delta \dot{x}_{1,2,3} + 10 \Delta \dot{x}_{4,5,3} + 50 \Delta \dot{x}_{7,8,9} + 1000 \Delta \dot{x}_{10,11,12} \quad (3.1)$$

A ponderação é baseada no fato de que as variáveis angulares geram um impacto maior na estabilidade do sistema e, por isso, devem sofrer menores desvios. Além disso, busca-se para compensar a certo nível as diferenças dimensionais entre variáveis. Essas ponderações foram obtidas por tentativa e erro, e podem ser ajustadas pelo usuário.

A biblioteca implementa nativamente as combinações de variáveis mais representativas do voo de aeronaves comerciais. Nominalmente: voo de cruzeiro, voo de subida, voo de descida, voo em curva e manobra *pull-up*. Nesses casos, é requerido apenas uma pequena quantidade de entradas por parte do usuário.

3.2 LINEARIZAÇÃO

Dada a complexidade atrelada a dinâmica de aeronaves, é comum que diversas análises sejam realizadas considerando o modelo linearizado em torno de um ponto de operação. Essa aproximação permite análises no domínio da frequência e a aplicação de estratégias de controle clássico, ambas técnicas bastante utilizadas em aeronáutica.

De forma simplificada, o processo de linearização permite a obtenção de duas matrizes \mathbf{A} e \mathbf{B} tal que as equações diferenciais do sistema $\dot{\vec{X}} = f(\vec{X}, \vec{U})$ possam ser aproximadas de acordo com a Equação 3.2 para pequenas perturbações em torno do ponto de operação.

$$\dot{\vec{X}} = f(\vec{X}, \vec{U}) \approx f(\vec{X}_e, \vec{U}_e) + \mathbf{A}\delta\vec{X} + \mathbf{B}\delta\vec{U} \quad (3.2)$$

Na Equação 3.2, \vec{X} é o vetor de estados, \vec{U} é o vetor de controles, \vec{X}_e é o vetor de estados no ponto de operação, \vec{U}_e é o vetor de controles no ponto de estudo, $\delta\vec{X}$ é a perturbação dos estados em torno do ponto de operação e, finalmente, $\delta\vec{U}$ é a perturbação dos controles em torno do ponto de operação.

Cada termo das matrizes \mathbf{A} e \mathbf{B} é obtido a partir do truncamento no primeiro valor da expansão da função $f(\vec{X}, \vec{U})$ em termos de uma série de Taylor, como apresentado nas Equações 3.3 e 3.4. Sendo que cada caluna das matrizes \mathbf{A} e \mathbf{B} é obtida numericamente a partir das Expressões 3.5 e 3.6,

$$\mathbf{A} = \begin{bmatrix} \frac{\partial \dot{X}_1}{\partial X_1} & \frac{\partial \dot{X}_1}{\partial X_2} & \cdots & \frac{\partial \dot{X}_1}{\partial X_n} \\ \frac{\partial \dot{X}_2}{\partial X_1} & \frac{\partial \dot{X}_2}{\partial X_2} & \cdots & \frac{\partial \dot{X}_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{X}_n}{\partial X_1} & \frac{\partial \dot{X}_n}{\partial X_2} & \cdots & \frac{\partial \dot{X}_n}{\partial X_n} \end{bmatrix} \quad (3.3)$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial \dot{X}_1}{\partial U_1} & \frac{\partial \dot{X}_1}{\partial U_2} & \cdots & \frac{\partial \dot{X}_1}{\partial U_n} \\ \frac{\partial \dot{X}_2}{\partial U_1} & \frac{\partial \dot{X}_2}{\partial U_2} & \cdots & \frac{\partial \dot{X}_2}{\partial U_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{X}_n}{\partial U_1} & \frac{\partial \dot{X}_n}{\partial U_2} & \cdots & \frac{\partial \dot{X}_n}{\partial U_n} \end{bmatrix} \quad (3.4)$$

$$\mathbf{A}_i = \frac{f(\vec{X}_e + \epsilon, \vec{U}_e) - f(\vec{X}_e - \epsilon, \vec{U}_e)}{2\epsilon} \quad (3.5)$$

$$\mathbf{B}_i = \frac{f(\vec{X}_e, \vec{U}_e + \epsilon) - f(\vec{X}_e, \vec{U}_e - \epsilon)}{2\epsilon} \quad (3.6)$$

onde ϵ é uma perturbação em torno do ponto de operação do i -ésimo estado ou controle. Esse valor é definido por padrão como $\epsilon = 0,01$, podendo ser alterado pelo usuário.

3.3 SEPARAÇÃO MODAL

A matriz de estados possui dimensão 12×12 , o que representa uma quantidade significativa de informação a ser trabalhada. Por isso, é comum a sua divisão em submatrizes representando dinâmicas localizadas, as quais são conhecidas como dinâmicas de ordem reduzida. Essas novas matrizes buscam conservar as frequências naturais e razão de amortecimento do modelo linearizado original e, portanto, podem ser utilizadas para propósitos de análise e controle.

A divisão mais clássica é quanto ao plano de movimento da aeronave. Ela parte da observação de que, para aeronaves de asa fixa em voo reto nivelado, os estados relacionados ao movimento longitudinal (subíndice " l ") são praticamente desacoplados daqueles relacionados ao movimento látero-direcional (subíndice " ld "). Isso permite que esses movimentos sejam tratados individualmente sem muitas preocupações de que outras dinâmicas sejam impactadas. A divisão consiste basicamente do adequado recorte das matrizes \mathbf{A} e \mathbf{B} de acordo com os estados de interesse, como apresentado nas Equações 3.7 e 3.8.

$$\delta \vec{X}_{ld} = \begin{bmatrix} \delta v & \delta \phi & \delta p & \delta r & \delta \psi \end{bmatrix}^T \quad (3.7)$$

$$\delta \vec{X}_l = \begin{bmatrix} \delta u & \delta w & \delta \theta & \delta q & \delta z \end{bmatrix}^T \quad (3.8)$$

As questões de estabilidade e controle, por outro lado, geralmente são abordadas separando as dinâmicas de acordo com as frequências naturais e amortecimento, o que pode ser descrito em termos dos modos naturais de oscilação. Essa decisão se dá principalmente pelas diferenças entre os impactos gerados por cada modo no comportamento global da aeronave. Nesse contexto, modos de período curto geralmente são mais críticos, pois estão diretamente relacionadas à manobrabilidade e indicam rápida divergência em caso de instabilidade. Modos de período longo, por outro lado, estão relacionadas a manutenção do voo em um regime estacionário a longo termo, permitindo margens maiores de tempo de resposta.

Usualmente são considerados 5 modos de oscilação, os quais são descritos em termos de estados específicos. Esses modos são: período curto (subíndice sp , Equação 3.9), período longo (subíndice ph , Equação 3.10), rolamento puro (subíndice r , Equação 3.11), espiral (subíndice s , Equação 3.12) e *dutch-roll* (subíndice dr , Equação 3.13).

$$\delta \vec{X}_{sp} = \begin{bmatrix} \delta w & \delta q \end{bmatrix}^T \quad (3.9)$$

$$\delta \vec{X}_{ph} = \begin{bmatrix} \delta u & \delta \theta & \delta z \end{bmatrix}^T \quad (3.10)$$

$$\delta \vec{X}_r = \begin{bmatrix} \delta q \end{bmatrix} \quad (3.11)$$

$$\delta \vec{X}_s = \begin{bmatrix} \delta \phi \end{bmatrix} \quad (3.12)$$

$$\delta \vec{X}_{dr} = \begin{bmatrix} \delta v & \delta r \end{bmatrix}^T \quad (3.13)$$

Como essas dinâmicas não são completamente desacopladas, algumas delas precisam de correções para compensar a ausência de certos estados, como apresentado em Cook (2007).

3.4 QUALIDADES DE VOO

Os parâmetros de qualidade de voo relacionam o comportamento da aeronave às sensações do piloto. Como opiniões de pilotos são questões subjetivas, alguns estudos foram conduzidos de modo a estabelecer uma escala padrão, a qual é conhecida como escala *Cooper-Harper* (COOK, 2007). Essa escala atribui às aeronaves um valor de 1 a 10, que podem ser diretamente relacionados a 3 níveis de qualidade de voo, como apresentado na Figura 3.1.

Figura 3.1 – Relação entre a escala *Cooper-Harper* e os níveis de qualidade de voo.

Nível de qualidade de voo	Nível 1			Nível 2			Nível 3			Inaceitável
Escala Cooper-Harper	1	2	3	4	5	6	7	8	9	10

Fonte: Adaptado de (COOK, 2007)

De forma resumida, cada nível pode ser interpretado como apresentado abaixo:

- Nível 1: Qualidades de voo adequadas para a missão;

- Nível 2: Qualidades de voo suficientes para completar a missão, porém com carga de trabalho aumentada e/ou degradação da efetividade da missão;
- Nível 3: Qualidades de voo degradadas, porém a aeronave pode ser controlada. A efetividade da missão é inadequada e/ou a carga de trabalho do piloto é excessiva.
- Acima de 3 (inaceitável): Aeronave não pode ser controlada.

Os níveis de qualidade de voo dependem da fase de voo e tipo de aeronave. As fases de voo são separadas em 3 categorias, como apresentado abaixo:

- Categoria A: Fases de voo não terminal que requerem manobras rápidas ou rastreamento preciso de trajetória. Por exemplo: combate ar-ar, ataque ao solo, missão de reconhecimento, reabastecimento em voo, voo em formação próxima.
- Categoria B: Fases de voo não terminal que requerem manobras graduais e menor precisão de rastreamento. São exemplos os voos de cruzeiro, subida ou loiter.
- Categoria C: Fases de voo terminal que requerem manobras graduais e trajetórias precisas. São exemplos: fase de decolagem, pouso, arredondamento.

As aeronaves, por sua vez, são classificadas de acordo com o tamanho e massa, como segue:

- Classe I: Aeronaves pequenas e leves;
- Classe II: Aeronaves de peso médio com manobrabilidade média ou baixa;
- Classe III: Aeronaves grandes e pesadas com baixa manobrabilidade;
- Classe IV: Aeronaves de alta manobrabilidade.

Como destacado na Sessão 3.3, cada modo de oscilação da aeronave gera um impacto mais ou menos crítico em seu desempenho. Em razão disso, os requisitos para cada modo são distintos entre si, como apresentado no Apêndice B.

3.5 SIMULAÇÃO NÃO LINEAR

A simulação não-linear consiste na integração das equações diferenciais do sistema ao longo do tempo. A biblioteca conta com 3 tipos de integradores, sendo que um deles é escolhido automaticamente pela plataforma dependendo do formato do sistema construído. Como especificado na Tabela 3.1.

No método 2, todas as *flags* são interpretadas pelo sistema antes do início da simulação e em seguida os instantes de tempo críticos são informados ao integrador. Portanto,

Tabela 3.1 – Informações sobre os integradores disponíveis da biblioteca PyAAT.

Nome	Método	Descrição
Método 1	Implicit multi-step variable-order (BDF)	Método padrão. Possui passo de tempo variável.
Método 2	Implicit multi-step variable-order (BDF)	Mesmo algoritmo aplicado no Método 1, porém dotado de um sistema de <i>flags</i> que informa ao integrador quando reduzir o passo de tempo da simulação. Essa capacidade é necessária para capturar perturbações que ocorrem de forma abrupta e durante pequenos intervalos de tempo, as quais poderiam passar despercebidas.
Método 3	Runge-Kuta 5(4) (RK45)	Integrador de passo fixo.

Fonte: Autor

não pode ser esperado que esse sistema seja capaz de prevenir que sinais abruptos passem despercebidos em sistemas em malha fechada. Ainda, nenhuma atenção foi dada a controladores de ação abrupta, como estratégias bang-bang, por exemplo.

3.6 ENVELOPE DE VOO

O envelope de voo, também chamado de envelope de operação, é um gráfico de altitude em função da velocidade que apresenta as condições limite de operação de uma aeronave. Esse gráfico pode ser construído para qualquer condição de voo, porém, neste trabalho é tratada apenas a condição de voo horizontal permanente.

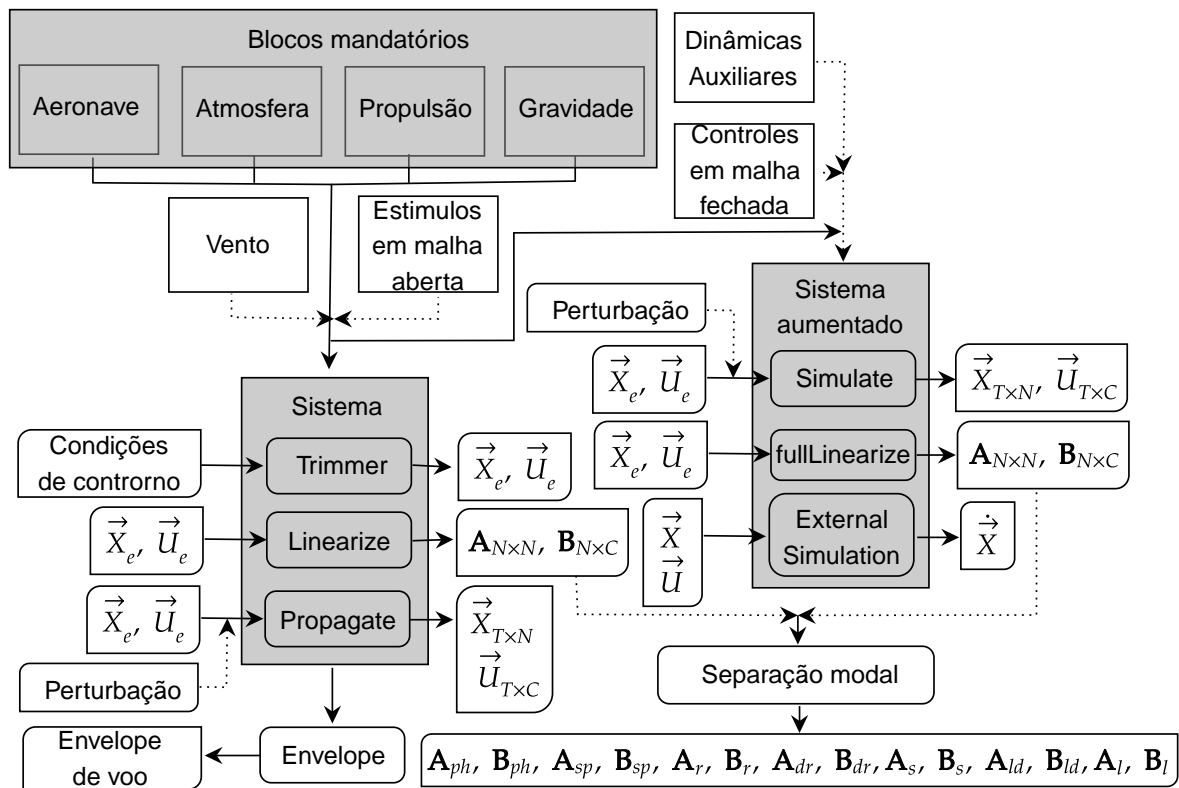
Do ponto de vista físico, os limites de uma aeronave podem ser de caráter propulsivo, aerodinâmico ou estrutural, como especificado abaixo:

- Limite aerodinâmico: É dada pela condição de mínima velocidade capaz de manter um voo sustentado. Do ponto de vista físico, é a condição em que a força de sustentação máxima se iguala à força peso.
- Limite estrutural: Corresponde a máxima pressão dinâmica que a estrutura da aeronave pode suportar, o que, indiretamente, reflete a condição de tensões da aeronave.
- Limite propulsivo: É a condição onde a tração máxima gerada pelo conjunto propulsivo se iguala ao arrasto provocado pela aeronave.

4 PYTHON AEROSPACE ANALYSIS TOOLBOX

O *software* desenvolvido como parte desse trabalho é a *Python Aerospace Analysis Toolbox (PyAAT)*, a qual tem como principal objetivo simplificar a modelagem e análise da mecânica de voo de aeronaves. De forma resumida, a PyAAT funciona como uma plataforma para descrição e integração de subsistemas. Para tal, a biblioteca padroniza a forma de construção dos subsistemas de modo garantir um fluxo adequado de informação entre estes. A Figura 4.1 apresenta um fluxograma geral de uso da biblioteca.

Figura 4.1 – Fluxograma de uso da biblioteca PyAAT.



Fonte: Autor

Na Figura 4.1, os blocos retangulares indicam objetos, os com 4 cantos arredondados são funções, e os com 2 cantos arredondados indicam entradas ou saídas numéricas de uma função. Por exemplo, 'Aeronave' é um objeto, 'Trimmer' é uma função e 'Perturbações' são entradas (geralmente um conjunto de valores numéricos). Ainda, as linhas contínuas indicam que o bloco é obrigatório no uso ou construção do bloco para o qual ele aponta, enquanto linhas pontilhadas indicam que é opcional. Por exemplo, os blocos 'Aeronave', 'Atmosfera', 'Propulsão' e 'Gravidade' são obrigatórios para a construção do bloco de sistema, enquanto 'Vento' é opcional.

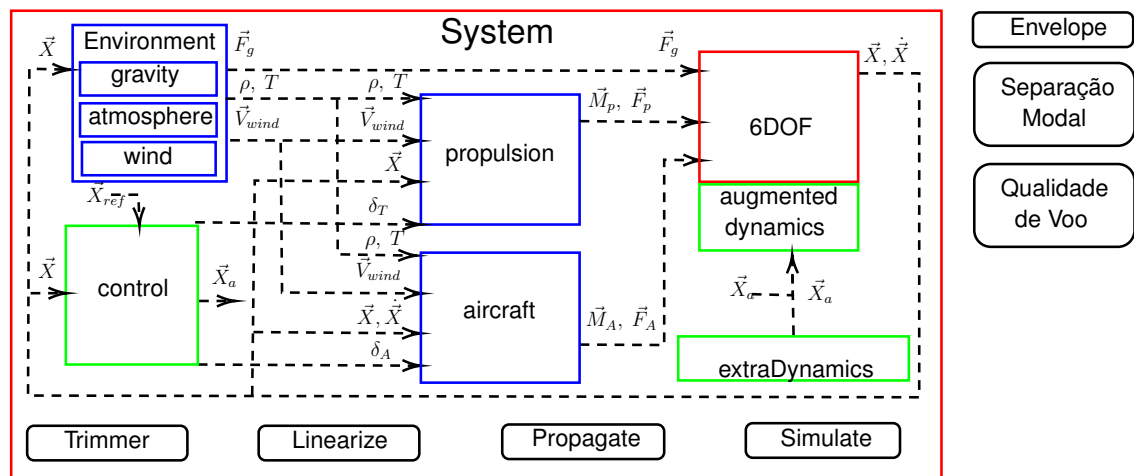
De modo geral, o uso da biblioteca gira em torno do bloco de sistema (sistema

aumentado é equivalente, porém com mais informações). Esse bloco é a plataforma integradora em si, a qual é responsável por gerenciar os demais blocos e implementar as equações diferenciais (EDs) do sistema. Esse bloco não pode ser diretamente modificado pelo usuário, contudo, alguns blocos são capazes de alterar as EDs.

Para garantir que todos os blocos estejam funcionando corretamente e assegurar a comunicação entre eles, a plataforma conta com um sistema de detecção e indicação de erros, o qual faz a verificação de entradas e saídas de cada bloco comparando ao tipo de informação esperada, e, caso as informações não sejam compatíveis, a ferramenta notifica ao usuário informando o tipo de erro e em qual bloco ocorreu. Como a verificação é centralizada, essa funcionalidade se aplica também aos blocos implementados pelos usuários.

Quanto ao fluxo de informação, por padrão, todos os blocos têm acesso às propriedades dos demais, porém, com permissão limitada para modifica-las. Além disso, também são acessíveis todas as informações referentes a simulação, como instante de tempo, passo de tempo, estados e derivadas de estados. A Figura 4.2 apresenta o fluxo de informação entre os blocos.

Figura 4.2 – Blocos construtivos e fluxo de informação da biblioteca PyAAT.



Fonte: Autor

Os objetos em azul da Figura 4.2 são chamados submodelos. Esses blocos implementam equações estáticas, as quais recebem um conjunto de parâmetros de entrada e fornecem outro conjunto como saída. Por exemplo, o bloco de gravidade recebe a posição do corpo e gera o vetor força gravitacional.

Os objetos em verde também são submodelos, contudo, esses blocos podem conter dinâmicas internas, as quais são acrescentadas às equações diferenciais do sistema. Alguns exemplos de blocos dinâmicos são atuadores, filtros e alguns tipos de controladores. Nesse sentido, o bloco *augmented dynamic* é criado quando algum objeto desse tipo é adicionado ao sistema.

4.1 DESCRIÇÃO DOS SUBMODELOS

Cada um dos submodelos pode ser implementado pelo usuário ou definido a partir dos modelos nativos da biblioteca. Abaixo é apresentada uma descrição de cada bloco juntamente com os modelos já disponíveis.

- *Gravity* (obrigatório): O modelo de gravidade fornece o vetor campo gravitacional em função da posição. Os modelos disponíveis são apresentados abaixo e equacionados no Apêndice A.1.
 - *NewtonGravity*: Modelo gravitacional de Newton.
 - *HighOrder*: Modelo de alta ordem, descrito em termos de harmônicos esféricos.
 - *VerticalConstant*: Aceleração independente da altitude e apontando na vertical.
- *Atmosphere* (obrigatório): O modelo atmosférico fornece a densidade do ar, temperatura, pressão e número de Mach para uma determinada altitude, como apresentado no Apêndice A.2. Os modelos disponíveis na biblioteca são:
 - *atmosISA*: *International Standard Atmosphere* (ISA).
 - *atmosCOESA*: *U.S Standard Atmosphere 1976 - COESA*.
 - *simplifiedModel*: Simplificação do modelo ISA apresentada em Cook (2007).
 - *windTunnel*: Aplica condições definidas pelo usuário. Por padrão usa o modelo ISA ao nível do mar.
- *Propulsion* (obrigatório): O bloco de propulsão é composto por um ou mais propulsores independentes entre si, cada um possuindo três graus de liberdade de controle, sendo eles: a posição da manete e 2 ângulos de apontamento. Por padrão, a posição da manete é normalizada entre 0 e 1, correspondendo a mínima e máxima ação de controle. Essa ação está relacionada a variáveis diferentes dependendo do tipo de motor e.g. velocidade de rotação para pequenos propulsores a hélice, manete de combustível em motores a jato, etc. As saídas do bloco são as forças e momentos líquidos gerados pelo conjunto dos propulsores. Detalhes sobre os modelos propulsivos são apresentados no Capítulo 2. A biblioteca dispõe dos seguintes modelos:
 - *JetModel*: Modelo genérico para motores a turbo-jato e turbo-fan.
 - *Propeller*: Modelo simplificado para pequenos propulsor a hélice.
 - *NoEngine*: Token para simulação de veículos sem propulsores.
- *Wind* (Opcional): Modelos de vento, os quais fornecem um vetor contendo as velocidades de vento em função dos estados.
 - *gust*: Modelo de rajada FAR 25.341-1, como apresentado no Apêndice A.3.

- *Aircraft* (obrigatório): É o modelo da aeronave em si, o qual possui duas funções principais: A primeira é armazenar as informações referentes à aeronave, incluindo geometria, propriedades de inércia e características aerodinâmicas. A segunda é computar as forças e momentos aerodinâmicos em função dos estados, ambiente e deflexão das superfícies de controle (quando existentes). As informações armazenadas e modelo aerodinâmico aplicado dependem da arquitetura da aeronave.
 - *Aircraft*: Implementa um modelo aerodinâmico baseado em derivadas de estabilidade e controle, conforme apresentado no Capítulo 2.
- *ExtraDynamics* (opcional): Todas as dinâmicas extras da aeronave que são implementadas em caráter global, i.e, dinâmicas que modificam permanentemente o comportamento de uma variável. Por exemplo, caso seja incluído um modelo de atuador, todos os sinais de controle aplicados à superfície em questão serão modificados por essa dinâmica. Detalhes sobre os modelos são apresentados na Seção A.4.
 - *sensor*: Sensor modelado como filtro passa-baixa;
 - *actuator*: Atuador formulado na forma de um atraso de primeira ordem;
- *Control* (opcional): Inclui as estratégias de controle em malha aberta e malha fechada, as quais podem ou não conter dinâmica. Todas as dinâmicas incluídas com esse bloco são locais, visto que são processadas a nível de controlador. Por exemplo, ao incluir-se o modelo de sensor em uma malha de controle, o valor modificado estará disponível unicamente para a malha em questão. Por outro lado, se o mesmo modelo for incluído no bloco *ExtraDynamics*, o sinal estará disponível para todas malhas (como se fosse fornecido pelo barramento digital da aeronave).
 - *loop*: Malha de controle para uma entrada e uma saída.
 - *P, PD, PI, PID, leadLag*: Estratégias de controle clássico.
 - *LowPassFilter, HighPassFilter*: Filtros passa-baixa e passa-alta.
 - *doublet, step*: Ações de controle em malha aberta.

4.2 DESCRIÇÃO DOS MÉTODOS

A biblioteca conta com alguns métodos que auxiliam na extração e análise de resultados, os quais são ilustrados em preto na Figura 4.2. Como pode ser observado, alguns métodos são aplicados diretamente ao sistema, enquanto os outros se aplicam a resultados gerados pelo sistema. Os métodos presentes na biblioteca são:

- Trimmer: Função para cálculo do balanço da aeronave em uma condição de voo desejada.
- Linearize/FullLinearize: Funções de linearização do sistema em torno de um ponto de equilíbrio. A primeira considera o sistema em malha aberta e condições padrão, i.e, atmosfera estacionária, sem dinâmicas auxiliares ou controles em malha fechada. A segunda considera o sistema completo e em malha fechada.
- Propagate/Simulate: Servem para resolver as equações diferenciais do sistema. A primeira se aplica somente a sistemas em malha aberta, enquanto a segunda implementa funcionalidades extras para incluir dinâmicas auxiliares e controles em malha fechada.
- Envelope: Constrói o envelope de voo da aeronave e calcula o teto de serviço.
- Separação modal: Divide as matrizes do sistema linearizado de acordo com os modos de oscilação padrão de aeronaves de asa fixa.
- Qualidade de voo: Gera os mapas de qualidade de voo para cada modo de oscilação.
- ExternalSimulation: Método que gera um modelo de aeronave pronto para simulação fora da biblioteca.

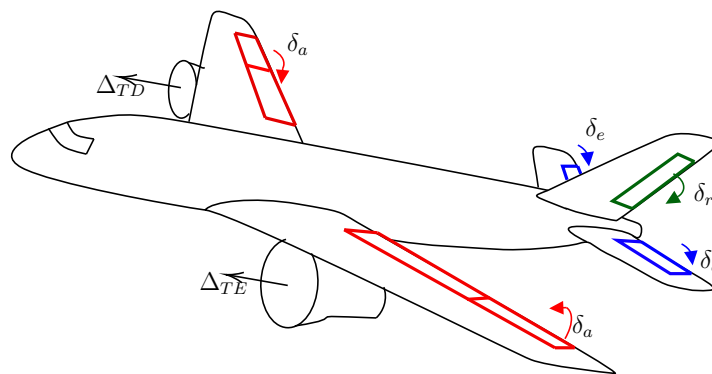
Embora a biblioteca conte com funções para análise de sistemas espaciais, nesse trabalho será abordado apenas o módulo de sistemas aeronáuticos.

5 APLICAÇÃO EM AERONAVE COMERCIAL DE GRANDE PORTE

A aeronave adotada nesse capítulo é uma adaptação do avião conceitual apresentado por Silva, Paglione e Yoneyama (2010). Valores de coeficientes aerodinâmicos, propulsivos e propriedades geométricas são apresentadas no Apêndice C.

Do ponto de vista de controles, o veículo possui três superfícies aerodinâmicas, sendo elas: aileron, com deflexão anti-simétrica δ_a , profundor, com deflexão δ_e , e leme, com deflexão δ_r . A propulsão é realizada por dois motores a jato instalados abaixo da asa, como ilustrado na Figura 5.1. Ambos os motores possuem apontamento fixo de $\theta_p=3^\circ$ e $\psi_p=0^\circ$, e podem gerar 35kN de força cada. O controle dos motores é realizado através das posições das manetes de combustível do motor esquerdo δ_{te} e direito δ_{td} , os quais geram as trações Δ_{TE} e Δ_{TD} , respectivamente.

Figura 5.1 – Variáveis de controle da aeronave de referência.



Fonte: Autor.

5.1 CONSTRUÇÃO DO SISTEMA

De modo geral, a criação de cada bloco é realizada pela instanciação do modelo desejado enquanto fornecendo um conjunto de parâmetros de configuração, como apresentado abaixo para os modelos gravitacional e atmosférico:

```
from pyaat.gravity import NewtonGravity
minhaGravidade = NewtonGravity()

from pyaat.atmosphere import atmosISA
minhaAtmosfera = atmosISA(deltaT = 288.15)
```

O modelo gravitacional não demanda nenhum parâmetro de configuração, enquanto o atmosférico pode ser inicializado com a temperatura a nível do mar ΔT em K, valor esse definido por padrão como 288.15K.

5.1.1 Criação do bloco aeronave

As arquiteturas de aeronaves ficam armazenados no pacote `aircraft`. A escolha de um modelo implica na escolha de uma dada geometria e método aerodinâmico associado. Nesse caso, a escolha do objeto do tipo `Aircraft` implica no uso de uma geometria de asa fixa e método aerodinâmico baseado em derivadas de estabilidade. A criação é indicada abaixo:

```
from pyaat.aircraft import Aircraft
minhaAeronave = Aircraft(mass=45e3, Ixx=0.554e6, Iyy=2.53e6, Izz =3.01e6,
Izx=0.106e6, Cbarw =3.666, bw=28.42, Sw=95.0)
```

onde os parâmetros de criação são: a corda média aerodinâmica C_{barw} , a envergadura da asa bw , a área em planta da asa Sw e as propriedades de massa e inércia.

Os coeficientes aerodinâmicos são informados atribuindo o valor ao diretamente ao atributo, como apresentado abaixo:

```
minhaAeronave.coef0 = [0.0252, 0.0, 0.382, 0.0, 0.0622, 0.0]
minhaAeronave.coefALPHA = [0.2010, 0.0, 6.29, 0.0, -3.63, 0.0]
minhaAeronave.coefBETA = [0.0, -0.785, 0.0, -0.121, 0.0, 0.174]
minhaAeronave.coefP = [0.0, 0.0794, 0.0, -0.522, 0.0, -0.0587]
minhaAeronave.coefQ = [0.281, 0.0, 14.6, 0.0, -45.5, 0.0]
minhaAeronave.coefR = [0.0, -0.572, 0.0, 0.254, 0.0, -0.277]
minhaAeronave.CLmax = 2.4
minhaAeronave.Vinf = 200
minhaAeronave.qmax = 23052.05
```

Esses são organizados em uma lista para cada estado. Por exemplo, a variável `coefALPHA` armazena os coeficientes aerodinâmicos dependentes de α , como apresentado abaixo:

$$\text{coefALPHA} = \left[\underbrace{0.2010}_{C_{D\alpha}} \quad \underbrace{0.0}_{C_{Y\alpha}} \quad \underbrace{6.29}_{C_{L\alpha}} \quad \underbrace{0.0}_{C_{l\alpha}} \quad \underbrace{-3.63}_{C_{m\alpha}} \quad \underbrace{0.0}_{C_{n\alpha}} \right]$$

Finalmente, as superfícies de controle podem ser incluídas por meio do método `set_control_surface`, o qual recebe como entradas: o nome da superfície de controle, a lista das derivadas de controle e uma tupla com batentes de deflexão da superfície em questão, como segue:

```

minhaAeronave.set_control_surface('Profundor',
[0.0126, 0.0, 0.3891, 0.0, -1.5980, 0.0], (radians(-30), radians(30)))
minhaAeronave.set_control_surface('Aileron',
[0.0, 0.0094, 0.0, -0.1784, 0.0, 0.0080], (radians(-20), radians(30)))
minhaAeronave.set_control_surface('Leme',
[0.0, -0.3030, 0.0, -0.0464, 0.0, -0.1594], (radians(-40), radians(40)))

```

Cada nova superfície incluída automaticamente cria uma variável de controle, a qual é referenciada pelo nome atribuído. Nesse sentido, cada objeto criado só pode receber um único nome, o qual não pode ser repetido em nenhum outro objeto.

Ainda, os batentes são utilizados apenas na função de balanço. Para se incluir limites de deflexões durante as simulações devem ser incluídos modelos de atuadores.

5.2 CRIAÇÃO DO BLOCO PROPULSIVO

A criação do bloco propulsivo consiste em definir cada motor individualmente e agrupá-los em uma lista. Para o caso de motores a jato, os parâmetros de configuração são: Nome, tração máxima, n_ρ , ρ_i , V_i e n_V . Posição e apontamento devem ser informados na forma de um array ou lista, do contrário o motor é posicionado sobre o centro de gravidade (C.G), e alinhado com sistema de referência \mathcal{B} . Abaixo é apresentada a criação do conjunto propulsivo da aeronave em estudo:

```

from pyaat.propulsion import JetModel
MotorEsquerdo = JetModel('Motor esquerdo', Fmaxi = 35e3, nrho = 0.775,
rhoi = 0.41271, Vi = 200.0, nv = 0)
MotorEsquerdo.position = np.array([0, -5, 1.42])
MotorEsquerdo.attitude = np.array([0., radians(3.), 0.])

MotorDireito = JetModel('Motor direito', Fmaxi = 35e3, nrho = 0.775,
rhoi = 0.41271, Vi = 200.0, nv = 0)
MotorDireito.position = np.array([0, 5.0, 1.42])
MotorDireito.attitude = np.array([0., radians(3.), 0.])

minhaPropulsao = [MotorEsquerdo, MotorDireito]

```

Como pode ser observado, o modelo consiste de dois motores idênticos posicionados em locais diferentes.

A inclusão de cada motor implica na criação de três variáveis de controle (uma de potência e duas de apontamento), as quais recebem o nome do motor seguida pela gran-

deza controlada. Por exemplo, o propulsor MotorEsquerdo cria as variáveis: deflexão de manete do motor esquerdo δ_{te} , com o nome 'Motor esquerdo thrust', arfagem do motor esquerdo θ_{pe} , com o nome 'Motor esquerdo thetap', e guinada do motor esquerdo ψ_{pe} , com o nome 'Motor esquerdo psip'. Note que, como esse processo é automático, o apontamento sempre é variável de controle, mesmo nos casos onde a aeronave possui motores com apontamento fixo.

5.3 INTEGRAÇÃO DOS SUBMODELOS

A integração do sistema é realizada utilizando o objeto `system`, o qual pode receber até 8 entradas, sendo as entradas mínimas: `aircraft`, `atmosphere`, `propulsion` e `gravity`. Como apresentado abaixo:

```
from pyaat.system import system
meuSistema = system(atmosphere = minhaAtmosfera, propulsion = minhaPropulsao,
aircraft = minhaAeronave, gravity = minhaGravidade)
```

5.4 CÁLCULO DO BALAÇO

O cálculo do balanço consiste em encontrar uma combinação de controles e estados que garantam a condição de voo desejada. Na biblioteca PyAAT, o método `trimmer` desempenha essa função. Como apresentado abaixo:

```
Xe, Ue = meuSistema.trimmer(condition, **kwargs)
```

O método tem como parâmetros obrigatórios o regime de voo (`condition`) e as condições de contorno, as quais são informadas através da variável `kwargs`. O caso mais geral é quando são fornecidas diretamente as variáveis livres (`free`), fixas (`fixed`) e desejadas (`desired`). Nesse caso, `condition` recebe o valor 'general', e as variáveis são informadas na forma de dicionários, como demonstrado abaixo o voo reto nivelado com velocidade $u = 224.6$ m/s e altitude de 10000m.

```
VF = {'Motor esquerdo psip':0.0, 'Motor direito psip': 0.0, 'zo':-10000.0,
'Motor esquerdo thetap':radians(3.0), 'u': 224.6, 'phi':0.0, 'v': 0.0,
'Motor direito thetap': radians(3.0)}
VD = {'zod':0.0, 'ud':0.0, 'vd':0.0, 'wd':0.0, 'phid':0.0, 'psid':0.0,
'thetad':0.0, 'pd':0.0, 'qd':0.0, 'rd':0.0}
VL = {'Motor esquerdo thrust': 0.5, 'Leme': radians(0.5), 'Profundor': 0.0,
```

```
'Aileron': radians(1.0), 'w':1.0, 'theta':0.0, 'p': 1.0, 'q':1.0, 'r':1.0}
VR = {'Motor direito thrust': 'Motor esquerdo thrust'}
Xe, Ue = meuSistema.trimmer(condition = 'general', fixed = VF, desired = VD,
free = VL, dependent = VD)
```

Além disso, restrições de proporcionalidade entre variáveis podem ser incluídas através do parâmetro `dependent`. Por exemplo, na condição em estudo, pode-se assumir que ambos os motores desempenham a mesma tração. Nesse caso, inclui-se um dos motores entre as variáveis livres, e o outro entre as restritas.

Ressalta-se também que todas as variáveis são referenciadas através do nome. Para o caso das variáveis de controles, esse é o valor é definido durante a criação das superfícies e propulsores, já para os estados e suas derivadas, é adotado o padrão da biblioteca, como apresentado nas tabelas 5.1 e 5.2.

Tabela 5.1 – Nomes atribuídos aos estados.

Coef.	x_o	y_o	z_o	u	v	w	ϕ	θ	ψ	p	q	r	α	β
Nome	xo	yo	zo	u	v	w	phi	theta	psi	p	q	r	alpha	beta

Fonte: Autor

Tabela 5.2 – Nomes atribuídos às derivadas dos estados.

Coef.	\dot{x}_o	\dot{y}_o	\dot{z}_o	\dot{u}	\dot{v}	\dot{w}	$\dot{\phi}$	$\dot{\theta}$	$\dot{\psi}$	\dot{p}	\dot{q}	\dot{r}
Nome	xod	yod	zod	ud	vd	wd	phid	thetad	psid	pd	qd	rd

Fonte: Autor

O método `trimmer` exige uma definição completa de variáveis, visto que todas as não especificadas são consideradas ignoráveis. Nesse contexto, o apontamento dos motores deve ser incluído entre os parâmetros fixos, do contrário ele é considerado alinhado ao sistema \mathcal{B} . Para o caso das variáveis livres, os valores informados são usados como chute inicial.

Os resultados gerados são o vetor de estado e de controles na condição de equilíbrio, como apresentado abaixo:

$$\vec{X}_e = [\underbrace{0.0}_{x_o} \quad \underbrace{0.0}_{y_o} \quad \underbrace{-10000.0}_{z_o} \quad \underbrace{224.5}_u \quad \underbrace{0.0}_v \quad \underbrace{1.5}_w \quad \underbrace{0.0}_\phi \quad \underbrace{0.0}_\theta \quad \underbrace{0.0}_\psi \quad \underbrace{0.0}_p \quad \underbrace{0.0}_q \quad \underbrace{0.0}_r]^T$$

$$\vec{U}_e = [\underbrace{0.0439}_{\delta_e} \quad \underbrace{0.0}_{\delta_a} \quad \underbrace{0.0}_{\delta_r} \quad \underbrace{0.3837}_{\delta_{pe}} \quad \underbrace{0.0}_{\psi_{pe}} \quad \underbrace{0.0524}_{\theta_{pe}} \quad \underbrace{0.3837}_{\delta_{pd}} \quad \underbrace{0.0}_{\psi_{pd}} \quad \underbrace{0.0523}_{\theta_{pd}}]^T$$

Os resultados são sintetizados abaixo:

- Ângulo de ataque α : 0.39° ;
- Ângulo de arfagem θ : 0.39° ;

- Manete de combustível esquerda δ_{te} : 38.39%;
- Manete de combustível direita δ_{td} : 38.39%;
- Deflexão do profundor δ_e : 2.52°;

Como pode ser observado, os ângulos α e θ são idênticos, o que indica a condição de voo reto nivelado. Como se trata de uma aeronave simétrica e esse tipo de manobra ocorre no plano longitudinal, nenhuma ação de controle de leme ou aileron é requerida. A deflexão do profundor é positiva, o que serve para contrabalancear o momento aerodinâmico gerado pela asa. Ambas as manetes de combustível são defletidas a pouco mais de 38%, havendo uma boa margem para manobras e gerenciamento de criticalidades. Isso também demonstra que a restrição de igualdade na tração foi corretamente aplicada.

Os controles são organizados na mesma ordem que são inseridos no sistema. Como consequência, pode ser difícil rastrear cada variável nos casos onde o sistema possui muitas superfícies aerodinâmicas ou propulsores. Considerando isso, o sistema disponibiliza essa informação através do método `controlsNames`, como demonstrado abaixo, que nesse caso gera a saída:

```
['Profundor', 'Aileron', 'Leme', 'Motor esquerdo thrust',
'Motor esquerdo phip', 'Motor esquerdo thetap', 'Motor direito thrust',
'Motor direito phip', 'Motor direito thetap']
```

5.4.1 Condições de balanço pré-implementadas

A biblioteca possui algumas condições de balanço pré-definidas. Para sua aplicação deve-se alterar o valor do parâmetro `condition`, bem como os valores a serem balanceados, como apresentado na Tabela 5.3. Em todos os casos é considerada a hipótese de que os motores possuem apontamento fixo e são controlados proporcionalmente.

Tabela 5.3 – Conjunto de variáveis para cálculo de equilíbrio.

Condição de voo	condition	**kargs
Voo reto nivelado	'cruise'	zo, u
Voo de subida ou descida	'climb'	zo, u, zod
Manobra <i>Pull-up</i>	'pullUp'	zo, u, thetap
voo em curva	'turn'	zo, u, psip, beta

Fonte: Autor

Esse caso geralmente só se aplica a modelos que incluem unicamente superfícies primárias, visto que não há nenhum controle sobre quais superfícies serão mais demandadas no balanço. Por exemplo, se forem incluídos flaps, o método pode utilizá-los para

garantir um voo de subida, porém, usualmente, deseja-se utilizar o profundor para esse propósito, visto que o arrasto gerado é muito menor.

5.5 SIMULAÇÕES EM MALHA ABERTA

A propagação de estados consiste na integração das equações do movimento considerando uma determinada condição inicial. Essa é uma maneira rápida de se estudar características relacionadas a estabilidade dinâmica da aeronave, e verificar se o processo de balanço foi bem sucedido. O método é apresentado abaixo:

```
estados, controles = meuSistema.propagate(Xe, Ue, T0, TF, dt, state, control)
```

O conjunto mínimo de entradas são: as condições iniciais de estados X_e , as condições iniciais de controles, U_e , e o tempo final de simulação TF , em segundos. Enquanto as entradas opcionais são: o passo de tempo dt e perturbações de estados (*state*) ou controles (*control*). Geralmente as condições iniciais X_e e U_e são valores de equilíbrio computados com a função de balanço.

Os resultados gerados são duas matrizes, as quais armazenam os valores de estados e controles ao longo do tempo. Essas matrizes possuem dimensão $12 \times T$ e $C \times T$, respectivamente, onde T é o número de passos de tempo e C é o número de variáveis de controle.

O vetor de instantes de tempo, por sua vez, é uma propriedade do sistema e pode ser acessada a qualquer momento após a simulação, como apresentado abaixo.

```
tempo = meuSistema.time
```

Esse vetor não corresponde aos instantes de tempo utilizados na simulação, os quais são definidos automaticamente pelo integrador, mas sim um vetor de tempos linearmente espaçados que servem ao propósito de análise dos resultados.

As perturbações de estados são informadas através de um dicionário. Como apresentado abaixo para o caso de uma perturbação de 2° no ângulo de derrapagem β e 5° no ângulo de ataque α .

```
state = {'beta': np.radians(2.), 'alpha': np.radians(5.)}
```

No caso de perturbações nos controles, um controlador em malha aberta deve ser incluído na simulação. Essa formalidade é necessária para garantir o padrão de entradas e saídas e a correta criação das *flags* que sinalizam os pontos críticos de simulação.

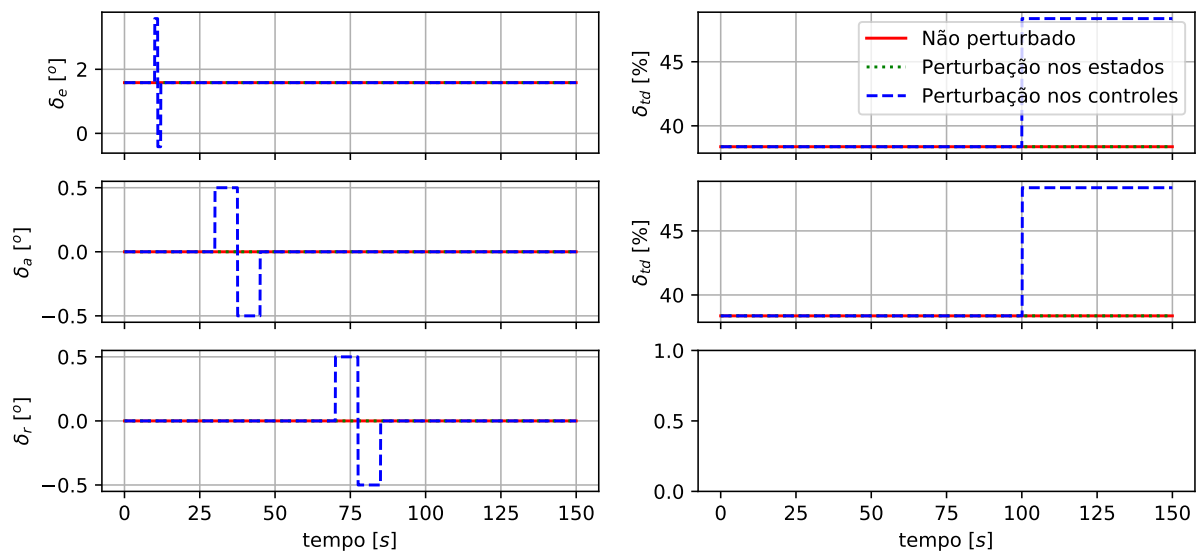
A biblioteca implementa estímulos do tipo *doublet* e degrau (*step*). As informações mínimas para cada estímulo são: O controle a ser perturbado, a amplitude da perturbação (*amplitude*), o tempo de início (*t_init*) e, para o caso do *doublet*, o período de duração

do estímulo (T). Abaixo é apresentada a aplicação de perturbações nas 5 variáveis de controle não fixas.

```
from pyaatcontrol import doublet, step
st0 = doublet('Profundor', amplitude = np.radians(2.), t_init = 10, T = 2)
st1 = doublet('Aileron', amplitude = np.radians(0.5), t_init = 30, T = 15)
st2 = doublet('Leme', amplitude = np.radians(0.5), t_init = 70, T = 15)
st3 = step('Motor esquerdo thrust', amplitude = 0.1, t_init = 100)
st4 = step('Motor direito thrust', amplitude = 0.1, t_init = 100.1)
control = [st0, st1, st2, st3, st4]
```

Todas as ações de controle criadas ser agrupados em uma lista, mesmo quando é aplicada apenas uma perturbação. Além disso, é uma limitação conhecida da biblioteca o fato de que estímulos devem ser incluídos em ordem cronológica. Ainda, eventos que ocorrem simultaneamente devem ser criados com alguns milissegundos de intervalo, como ocorre nesse caso para a tração dos motores. A Figura 5.2 ilustra o comportamento temporal das ações de controle aplicadas ao sistema.

Figura 5.2 – Ações de controle na aeronave em malha aberta.



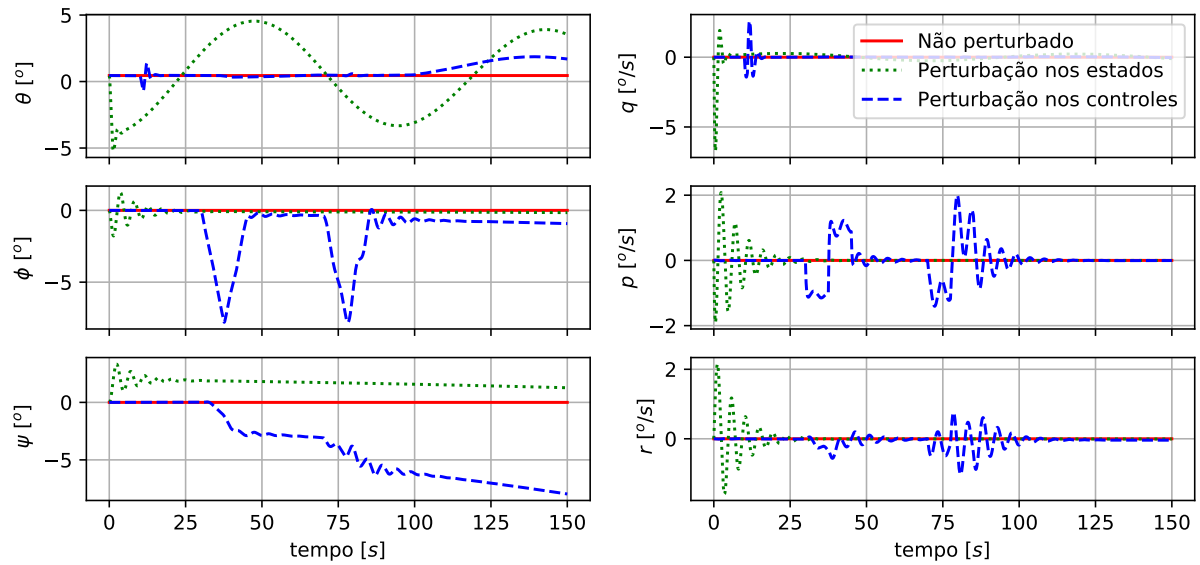
Fonte: Autor

Como o comando propagate considera atuadores ideais, cada ação é aplicada instantaneamente. Ainda, para os casos não perturbado e com aplicação de perturbações nos estados, as ações de controle permanecem em seus valores de equilíbrio, como esperado.

A atitude e velocidade angular da aeronave ao longo do tempo são apresentados na Figura 5.3, enquanto posições e velocidades lineares são apresentadas na Figura 5.4.

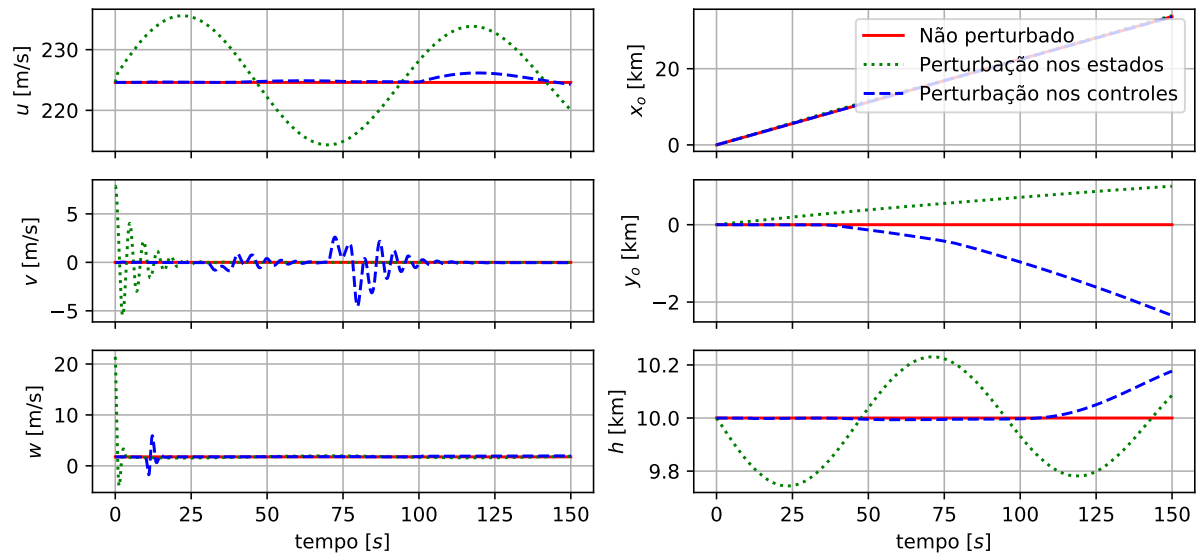
O primeiro ponto a ser observado é o comportamento das linhas vermelhas, as

Figura 5.3 – Atitude e velocidades angulares da aeronave em malha aberta.



Fonte: Autor

Figura 5.4 – Posição e velocidades linear da aeronave em malha aberta.



Fonte: Autor

quais correspondem a resposta a uma aplicação e manutenção de todos os parâmetros calculados no balanço. Como pode ser observado, essa condição é mantida ao longo de todo tempo simulado, o que indica que essa é, de fato, uma condição de equilíbrio.

As variáveis q e w sofrem pequenas oscilações após a aplicação da perturbação nos estados e de profundor, e rapidamente retornam ao equilíbrio. Isso demonstra que a aeronave é estável em sua dinâmica longitudinal de período curto.

A perturbação nos estados também gera uma oscilação, porém de menor frequência, nas variáveis u , θ e z_o , as quais não retornam ao equilíbrio no período simulado. Pode-se observar que as variáveis z_o e u estão defasadas por aproximadamente 180° , o que é condicionado ao fato de que essa oscilação corresponde a uma conversão cíclica de energia potencial em cinética, e vice versa. Esse movimento está associado ao modo de oscilação de período longo, também chamado de fugóide.

Quando aplicada uma perturbação no profundor, o ângulo θ sofre uma pequena oscilação, enquanto u e z_o não são afetados. Isso indica que essa variável gera pouca influência no modo de período longo. Por outro lado, ao se perturbar a tração, imediatamente percebe-se um ganho de altitude, aumento de θ e alteração da velocidade u .

A perturbação nos estados afeta severamente as variáveis létero-direcionais, o que pode ser visto pela amplitude e tempo de duração das oscilações em p e q . Ainda assim, o fato de que ambas variáveis retornam ao equilíbrio indica que a aeronave é estável. O ângulo de guinada ψ atinge um equilíbrio após as oscilações iniciais, contudo, não retorna a condição inicial. Ao mesmo tempo o ângulo de rolamento ϕ retorna ao equilíbrio após oscilações com amplitude de aproximadamente 2° .

De modo geral, perturbações no leme e aileron afetam os mesmos estados, e em ambos os casos a dinâmica é oscilatória. Ainda, esses controles não geram efeitos visíveis na dinâmica longitudinal, o que indica um desacoplamento dinâmico nessa condição de voo.

5.6 MODELOS LINEARES

A linearização é aplicada através do método `Linearize`, o qual requer como entradas os estados e controles no ponto de operação, e gera as matrizes **A** e **B** do sistema linearizado, como apresentado abaixo:

```
A, B = meuSistema.Linearize(Xe, Ue)
```

Novamente, os valores de X_e e U_e geralmente são obtidos na etapa de balanço, mas isso não é uma regra. As matrizes **A** e **B** para o caso de voo reto nivelado são

apresentadas nas Equações 5.1 e 5.2, respectivamente.

$$\begin{bmatrix} x_o & y_o & z_o & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}
 \begin{bmatrix} x_o & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0068 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 y_o & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & -1.5323 & 0.0 & 224.5903 & 0.0 & 0.0 \\
 z_o & 0.0 & 0.0 & 0.0 & -0.0068 & 0.0 & 1.0 & 0.0 & -224.5903 & 0.0 & 0.0 & 0.0 \\
 u & 0.0 & 0.0 & -0.0 & -0.0049 & -0.0 & 0.0278 & 0.0 & -9.7672 & 0.0 & 0.0 & -1.6056 \\
 v & 0.0 & 0.0 & 0.0 & 0.0 & 0.0743 & -0.0 & 9.7672 & 0.0 & 0.0 & 1.5004 & 0.0 \\
 w & 0.0 & 0.0 & -0.0012 & -0.0825 & 0.0 & -0.6197 & 0.0 & -0.0666 & 0.0 & 0.0 & 218.7067 \\
 \phi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\
 \theta & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
 \psi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\
 p & 0.0 & 0.0 & 0.0 & 0.0 & -0.0261 & -0.0 & 0.0 & 0.0 & 0.0 & -3.8089 & -0.0 \\
 q & 0.0 & 0.0 & -0.0 & -0.0001 & 0.0 & -0.0023 & 0.0 & 0.0 & 0.0 & 0.0 & -1.1974 \\
 r & 0.0 & 0.0 & 0.0 & 0.0 & 0.0063 & -0.0 & 0.0 & 0.0 & 0.0 & -0.2169 & -0.0 \end{bmatrix}
 \quad (5.1)$$

$$\begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_{te} & \delta_{\psi_{pe}} & \delta_{\theta_{pe}} & \delta_{td} & \delta_{\psi_{pd}} & \delta_{\theta_{pd}} \end{bmatrix}
 \begin{bmatrix} x_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 y_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 z_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 u & -0.2212 & 0.0 & 0.0 & 0.7779 & 0.0 & -0.0157 & 0.7779 & 0.0 \\
 v & 0.0 & 0.207 & -6.6717 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 w & -8.5692 & 0.0 & 0.0 & -0.0408 & 0.0 & -0.2987 & -0.0408 & 0.0 \\
 \phi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 \theta & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 \psi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 p & 0.0 & -9.1199 & -2.6066 & 0.0279 & 0.0 & 0.1219 & -0.0279 & 0.0 \\
 q & -2.2943 & 0.0 & 0.0 & 0.0196 & 0.0 & -0.0004 & 0.0196 & 0.0 \\
 r & 0.0 & -0.2577 & -1.586 & 0.0591 & 0.0 & 0.0031 & -0.0591 & 0.0 \end{bmatrix}
 \quad (5.2)$$

Embora muita informação possa ser extraída de modelos linearizados, essa é uma tarefa complexa para matrizes com grandes dimensões. Nesse sentido, é bastante usual a aplicação de técnicas de redução de ordem, as quais podem ser aplicadas de acordo com o plano do movimento, ou com o modo de oscilação. Essas funcionalidades estão implementadas no pacote de mecânica de voo da PyAAT, de acordo com o apresentado na Tabela 5.4.

Tabela 5.4 – Funções de separação modal presentes na biblioteca PyAAT.

Modo de oscilação	Função	Saídas
Longitudinal	longMatrix	\mathbf{A}_l e \mathbf{B}_l
Látero-direcional	lateroMatrix	\mathbf{A}_{ld} e \mathbf{B}_{ld}
Período longo	phugoidMatrix	\mathbf{A}_{ph} e \mathbf{B}_{ph}
Período curto	shortPeriodMatrix	\mathbf{A}_{sp} e \mathbf{B}_{sp}
Rolamento puro	rollMatrix	\mathbf{A}_r e \mathbf{B}_r
<i>Dutch-roll</i>	dutchRollMatrix	\mathbf{A}_{dr} e \mathbf{B}_{dr}
Espiral	spirallMatrix	\mathbf{A}_s e \mathbf{B}_s

Fonte: Autor

Cada um dos métodos recebe como entrada as matrizes A e B do sistema completo, e retorna as matrizes reduzidas, como demonstrado abaixo para o caso do rolamento puro:

```
from pyaat.flightmechanics import rollMatrix
Ar, Br = rollMatrix(A, B)
```

É importante ressaltar que, como o impacto de cada controle nos modos de oscilação depende da arquitetura da aeronave, todos os controles são considerados nas matrizes de controle de ordem reduzida.

A comparação entre os autovalores da matriz A completa e dos modelos de ordem reduzida é apresentada na Tabela 5.5.

Tabela 5.5 – Comparação entre autovalores da matriz completa e matrizes de ordem reduzida.

Modo	Completa	Longitudinal	Ordem reduzida
Período curto	$-0.9111 \pm 2.2369j$	$-0.9111 \pm 2.2369j$	$-0.9110 \pm 2.2370j$
Período longo	$-0.0018 \pm 0.0657j$	$-0.0018 \pm 0.0657j$	$-0.0024 \pm 0.06572j$
Período longo	-0.0010	-0.0010	-0.0012
Látero-direcional			
<i>Dutch-roll</i>	$-0.1218 \pm 1.3339j$	$-0.1218 \pm 1.3339j$	$-0.1144 \pm 1.1742j$
Rolamento puro	-3.8090	-3.8090	-3.8187
Espiral	0.0051	0.0051	0.0053

Fonte: Autor

Como pode ser observado, os autovalores da dinâmica completa e das dinâmicas longitudinal e látero-direcional são idênticos até a quarta casa decimal. Os autovalores relacionados as matrizes de ordem reduzida, por outro lado, apresentam algumas diferenças significativas principalmente nos modos de período longo e *Dutch-roll*.

Todos os autovalores relacionados à dinâmica longitudinal possuem parte real negativa, o que indica estabilidade da aeronave nesse plano. Já no plano látero-direcional, apenas o modo espiral é instável, o que geralmente não representa um problema, dado o pequeno impacto desse modo na dinâmica geral da aeronave. Em soma, o valor do modo espiral reflete uma grande constante de tempo, e, portanto, essa instabilidade pode ser facilmente corrigida pelo piloto.

5.7 QUALIDADES DE VOO DA AERONAVE EM MALHA ABERTA

Uma forma mais objetiva de se avaliar a aeronave é através dos parâmetros de qualidade de voo, os quais são especificados em termos da frequência natural ω , razão de amortecimento ζ , e constante de tempo T . Como cada modo de oscilação possui

critérios específicos, os quais variam de acordo com o tipo de aeronave e regime de voo, a estratégia aplicada na PyAAT consiste em criar mapas de qualidade de voo em função desses parâmetros. A tabela 5.6 apresenta essas funções, bem como suas entradas,

Tabela 5.6 – Funções para análise de qualidade de voo presentes na biblioteca PyAAT.

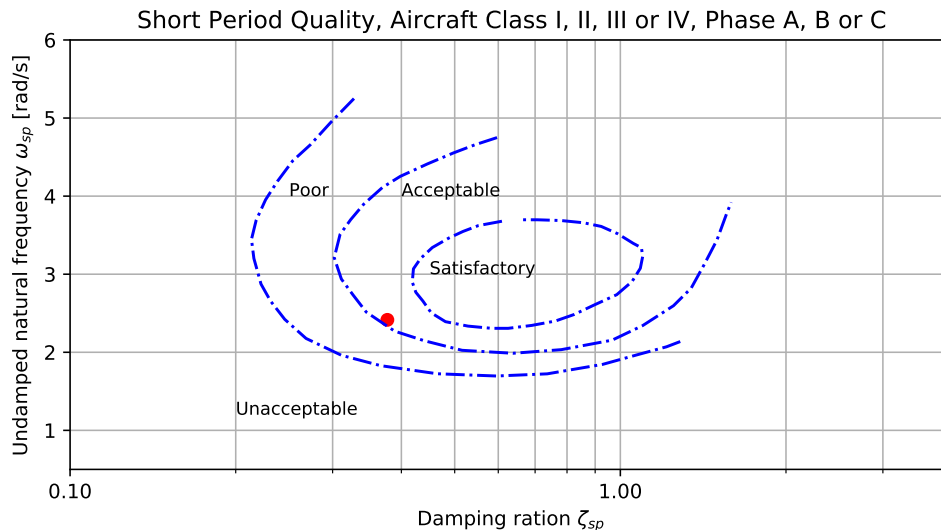
Modo de oscilação	Função	Entradas
Período longo	phugoidQuality	ω_{ph} , ζ_{ph} , aircraft, phase
Período curto	shortPeriodQuality	ω_{sp} , ζ_{sp} , aircraft, phase
Rolamento puro	rollQuality	T_r , aircraft, phase
<i>Dutch-roll</i>	dutchRollQuality	ω_{dr} , ζ_{dr} , aircraft, phase
Espiral	spiralQuality	T_s , aircraft, phase

Fonte: Autor

onde phase é a fase de voo, e aircraft é o tipo de aeronave, ambos definidos na Seção 3.4. As frequências, razões de amortecimento e constantes de tempo não são geradas pela biblioteca, e podem ser obtidas a partir das matrizes **A** do sistema linearizado.

As Figuras 5.5 - 5.9 apresentam os mapas de qualidade de voo para a aeronave em estudo (aeronave do tipo III) durante a fase de voo de cruzeiro (fase B).

Figura 5.5 – Qualidade de voo do modo longitudinal de período curto.

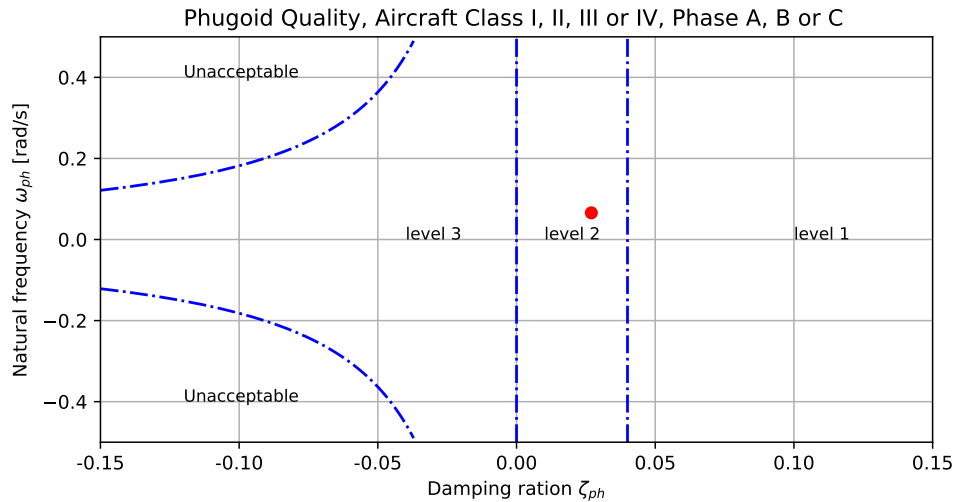


Fonte: Autor

Como mencionado anteriormente, cada modo de oscilação possui requisitos específicos, o que dificulta qualquer interpretação direta dos autovalores. Essa afirmação pode ser verificada no fato de que o modo espiral (instável) é classificado com nível 1 de qualidade de voo (nível máximo), enquanto algumas dinâmicas estáveis são classificadas em níveis inferiores.

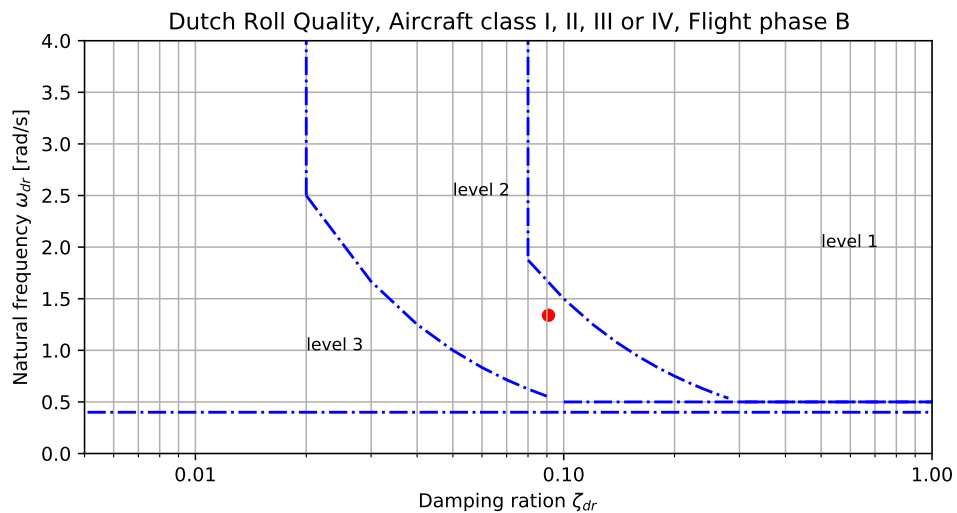
Pontualmente, pode-se observar que o modo *Dutch-roll* é classificado com qualidade de voo nível 2. Como essa dinâmica gera um grande impacto no comportamento

Figura 5.6 – Qualidade de voo do modo longitudinal de período longo.



Fonte: Autor

Figura 5.7 – Qualidade de voo do modo *Dutch-roll*.



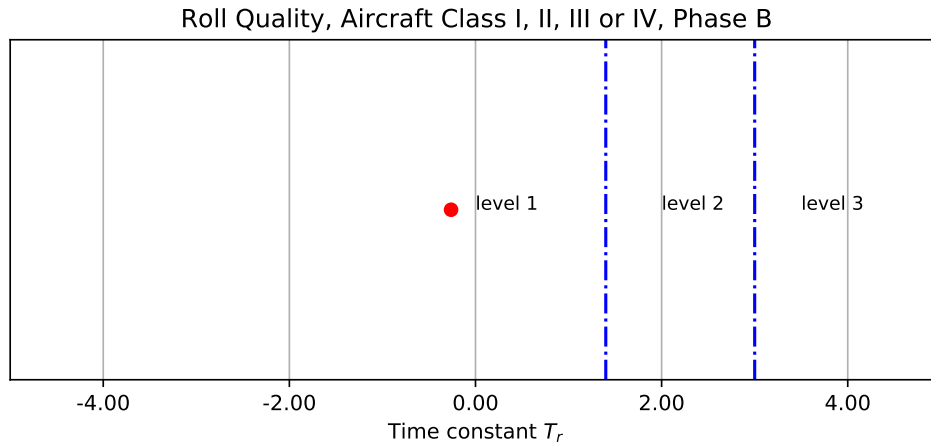
Fonte: Autor

da aeronave, essa característica deve ser melhorada, seja com reprojeção da aeronave, ou com uso de compensadores de malha fechada. Já na dinâmica longitudinal, ambos os períodos curto e longo são classificados com qualidade de voo nível 2, indicando que essas dinâmicas também precisam ser melhoradas.

5.8 SISTEMA DE AUMENTO DE ESTABILIDADE LONGITUDINAL

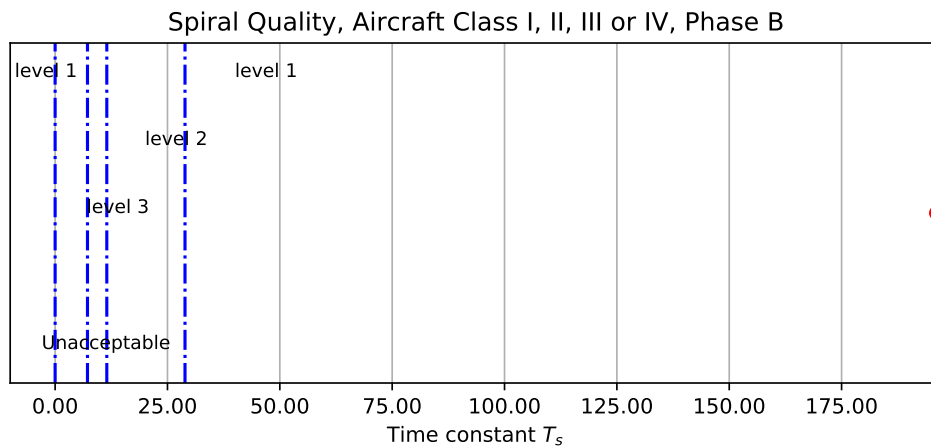
A estratégia utilizada na maioria das aeronaves modernas para a melhoria da resposta dinâmica é o uso de sistemas de aumento de estabilidade (SAS) (STEVENSON; LEWIS;

Figura 5.8 – Qualidades de voo do modo de rolamento puro.



Fonte: Autor

Figura 5.9 – Qualidade de voo do modo espiral.



Fonte: Autor

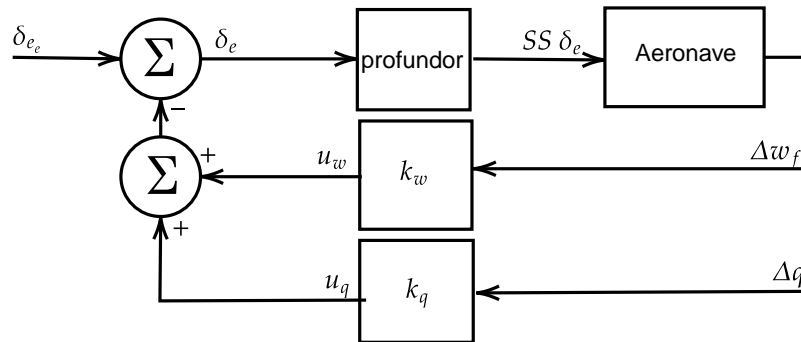
JOHNNSON, 2016). Essa técnica consiste no uso de controladores em malha fechada – geralmente realimentados pelas velocidades angulares – que geram um efeito de amortecimento artificial.

O principal propósito do SAS longitudinal é a melhoria das qualidades de voo de período curto. A Figura 5.10 apresenta uma malha com esse propósito, onde são medidas a velocidade w e velocidade angular q , e uma ação de controle proporcional ao desvio de cada medida em relação ao equilíbrio é aplicada ao profundor.

Na Figura 5.10, δ_{ee} é a deflexão do profundor no equilíbrio, δ_e é a ação comandada ao profundor, $SS\delta_e$ é a deflexão do profundor, Δw e Δq são os desvios de w e q em relação ao equilíbrio, u_w e u_q são as ações de controle geradas pelos erros de w e q , k_w e k_q são os ganhos de realimentação.

Como essa técnica depende inevitavelmente da medida de sinais e aplicação de controles, o comportamento de sensores e atuadores passa a desempenhar um papel

Figura 5.10 – Malha de sistema de aumento de estabilidade longitudinal.



Fonte: Autor

importante na dinâmica do sistema. Nesse sentido, as análises e projetos devem considerar tais dinâmicas. Nesse caso, a dinâmica do profundor é aproximada a um atraso de primeira ordem, enquanto o sensor de w é aproximado pela dinâmica de um filtro passa-baixa, como apresentado no Apêndice A.4. Nesse caso, w_f é o valor medido de w , e é esse sinal que é utilizado na realimentação. Abaixo é apresentada a criação do modelo do atuador de profundor e sensor de w .

```
from pyaat.pyaatcontrol import actuator, sensor
act2 = actuator(name = 'ss profundor', command = 'Profundor', tau = 0.2)
sens3 = sensor(name = 'wf', state = 'w', tau = 1/10)
actuatorsAndFilters = [act2, sens3]
```

Os parâmetros de criação são: o nome do componente, o estado medido ou superfície de controle atuada e a constante de tempo associada (nesse são considerados $\tau_e = 0.2s$ para o profundor, e $\tau_{wf} = 0.1s$ para o do sensor de w). Todas as dinâmicas são agrupadas em uma lista para então serem incluídas no sistema. Nesse caso, `wf` corresponde ao valor medido w_f , enquanto `ss profundor` ao valor real da deflexão do profundor $SS\delta_e$.

Os estados do sistema podem ser acessados a qualquer momento através do método `statesNames`, o qual é especialmente importante na interpretação dos resultados de simulações, uma vez que sistemas aumentados podem conter dezenas de estados extras.

Uma vez que novas dinâmicas são inseridas, as matrizes **A** e **B** do sistema linearizado são alteradas. Nesse contexto, as novas matrizes **A_a** e **B_a** do sistema aumentado podem ser geradas utilizando o método `FullLinearize`. A aplicação dessa função é semelhante a função `Linearize`, contudo, como a dinâmica possui estados extras, a dimensão de **X_e** também deve refletir esse acréscimo dimensional. Para a condição de voo reto

nivelado, as matrizes aumentadas ficam de acordo com as Equações 5.3 e 5.4.

$$\begin{bmatrix}
 & x_o & y_o & z_o & u & v & w & \phi & \theta & \psi & p & q & r & SS\delta_e & w_f \\
 x_o & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0076 & 0.0 & -0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 y_o & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & -1.7085 & 0.0 & 224.5915 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 z_o & 0.0 & 0.0 & 0.0 & -0.0076 & 0.0 & 1.0 & 0.0 & -224.5915 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 u & 0.0 & 0.0 & -0.0 & -0.0049 & -0.0 & 0.0283 & 0.0 & -9.7671 & 0.0 & 0.0 & -1.7774 & 0.0 & -0.2128 & 0.0 \\
 v & 0.0 & 0.0 & 0.0 & 0.0 & 0.0745 & -0.0 & 9.7671 & 0.0 & 0.0 & 1.6765 & 0.0 & -224.3685 & 0.0 & 0.0 \\
 w & 0.0 & 0.0 & -0.0012 & -0.082 & 0.0 & -0.6214 & 0.0 & -0.0743 & 0.0 & 0.0 & 218.6911 & 0.0 & -8.5921 & 0.0 \\
 \phi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0076 & 0.0 & 0.0 \\
 \theta & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 \\
 \psi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\
 p & 0.0 & 0.0 & 0.0 & 0.0 & -0.0262 & -0.0 & 0.0 & 0.0 & 0.0 & -3.8187 & -0.0 & 1.7966 & 0.0 & 0.0 \\
 q & 0.0 & 0.0 & -0.0 & 0.0 & 0.0 & -0.0233 & 0.0 & 0.0 & 0.0 & 0.0 & -1.2006 & 0.0 & -2.3004 & 0.0 \\
 r & 0.0 & 0.0 & 0.0 & 0.0 & 0.0063 & -0.0 & 0.0 & 0.0 & 0.0 & -0.218 & -0.0 & -0.3034 & 0.0 & 0.0 \\
 SS\delta_e & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -5.0 & 0.0 \\
 w_f & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 10.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & -10.0
 \end{bmatrix} \quad (5.3)$$

$$\begin{bmatrix}
 & \delta_e & \delta_a & \delta_r & \delta_{te} & \delta_{\psi_{pe}} & \delta_{\theta_{pe}} & \delta_{td} & \delta_{\psi_{pd}} & \delta_{\theta_{pd}} \\
 x_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 y_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 z_o & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 u & 0.0 & 0.0 & 0.0 & 0.7794 & 0.0 & -0.0157 & 0.7794 & 0.0 & -0.0157 \\
 v & 0.0 & 0.2075 & -6.6894 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 w & 0.0 & 0.0 & 0.0 & -0.0408 & 0.0 & -0.299 & -0.0408 & 0.0 & -0.2992 \\
 \phi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 \theta & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 \psi & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 p & 0.0 & -9.1446 & -2.6072 & 0.0279 & 0.0 & 0.122 & -0.0279 & 0.0 & -0.1221 \\
 q & 0.0 & 0.0 & 0.0 & 0.0197 & 0.0 & -0.0004 & 0.0197 & 0.0 & -0.0004 \\
 r & 0.0 & -0.2597 & -1.5903 & 0.0592 & 0.0 & 0.0031 & -0.0592 & 0.0 & -0.0031 \\
 SS\delta_e & 5.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
 w_f & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0
 \end{bmatrix} \quad (5.4)$$

Como o profundor é o único modelo de atuador incluído, a matriz B_a só tem a primeira coluna alterada em relação a matriz B . Nesse caso, o profundor passa a não desempenhar influência direta nas equações do movimento, mas afeta o estado do profundor $SS\delta_e$, o qual desempenha.

Além disso, as primeiras 12 linhas da coluna $SS\delta_e$ da matriz B_a são, na verdade, a própria coluna de δ_e da matriz B . Isso quer dizer que $SS\delta_e$ gera o mesmo efeito que seria gerado por δ_e caso o atuador fosse ideal, como é esperado.

5.8.1 Controladores em malha fechada

A inserção de controladores em malha fechada pode ser realizada de três maneiras: Diretamente na simulação, através da variável `control`, durante a criação do sistema, também através da variável `control`, ou em uma simulação fora da biblioteca. O segundo caso é ilustrado abaixo:

```

from pyaatcontrol import P, loop
P1 = P(name = 'P1', gain = 0.001)
loop1 = loop(name='L1', Xmeasure='wf', Xreference='w', command='Profundor')
loop1.loopControls = [P1]

P2 = P(name = 'P2', gain = 0.85)
loop2 = loop(name='L2', Xmeasure='q', Xreference='q', command='Profundor')
loop2.loopControls = [P2]

closedloopcontrols = [loop1, loop2]

```

A primeira etapa é criação de cada componente, sejam eles filtros, compensadores, inversores de fase, etc. Para esse caso, os únicos componentes são os controladores proporcionais P1 e P2, com ganhos $k_w = 0.001$ e $k_q = 0.65$, respectivamente.

Em seguida, cada malha pode ser construída utilizando o objeto `loop`, o qual é inicializado com: nome, estado de referencia, estado medido e superfície de controle que recebera o comando. Feito isso, os componentes podem ser incluídos na malha através do atributo `loopControls`.

Por exemplo, na malha `loop1` a medida do estado `'wf'` é comparada com o valor de referência `'w'` (condição inicial), e a diferença entre eles é a entrada do primeiro componente da malha (P1, nesse caso). A saída do primeiro componente é a entrada do segundo, e assim sucessivamente, até que o último componente aplica a ação de controle. Como nesse caso a malha só contém o componente P1, a saída é diretamente o comando aplicado ao profundor.

Os valores de referência são os fornecidos como condição inicial da simulação (geralmente uma condição de equilíbrio), e os valores medidos são os estados em si (ou medida de existe um modelo de sensor).

Finalmente, todas as malhas devem ser agrupadas em uma lista para então serem adicionadas ao sistema.

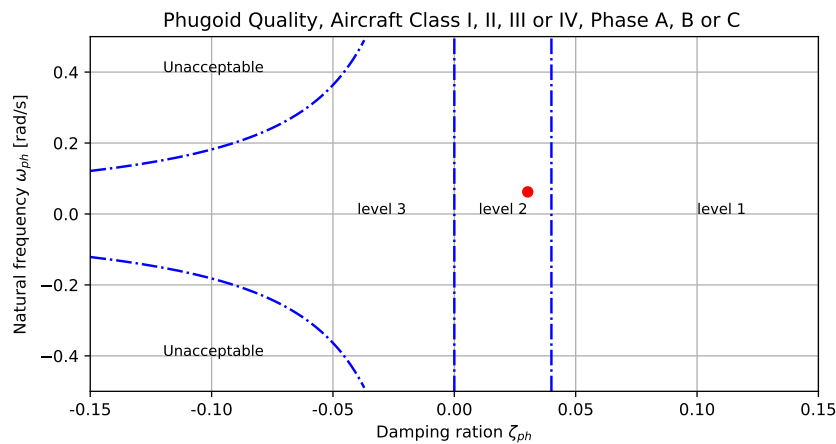
Os autovalores para os modos de período longo e período curto do sistema em malha fechada são apresentados na tabela 5.7, enquanto as qualidades de voo para esses modos são apresentadas nas Figuras 5.11 e 5.12, respectivamente.

Tabela 5.7 – Autovalores da dinâmica longitudinal antes e após inserção de SAS longitudinal.

Modo	Autovalores com SAS	Autovalores sem SAS
Período longo	-0.0025 ± 0.05119	$-0.0018 \pm 0.0657j$
Período curto	-2.0258 ± 3.0162	$-0.9111 \pm 2.2369j$

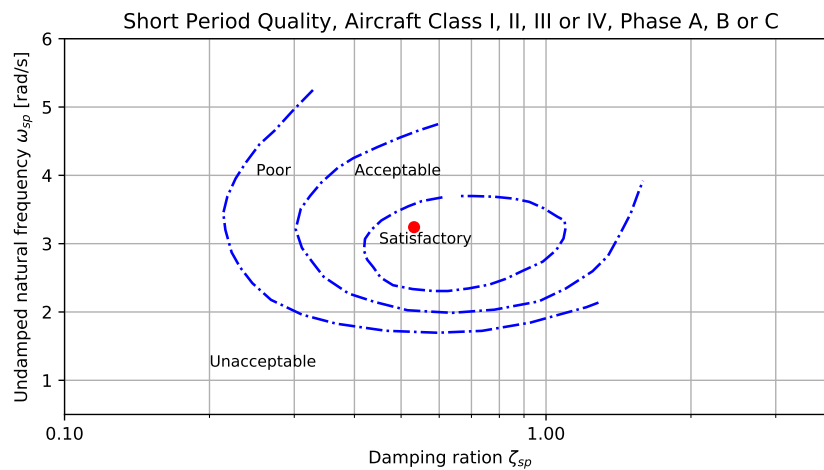
Fonte: Autor

Figura 5.11 – Qualidades de voo do modo de período longo em malha fechada.



Fonte: Autor

Figura 5.12 – Qualidades de voo do modo de período curto em malha fechada.



Fonte: Autor

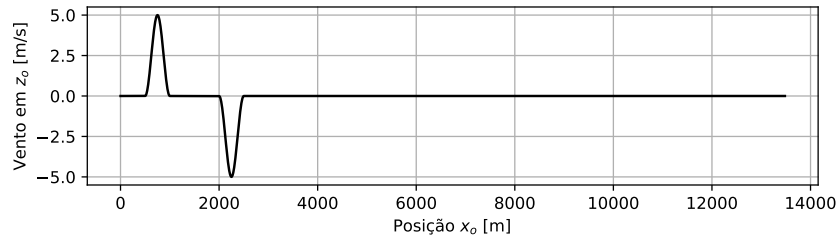
Como apresentado nas Figuras 5.11 e 5.12, o modo de período curto passa a ter qualidade de voo Nível 1, enquanto o modo de período longo mantém a qualidade de voo Nível 2. Essa condição pode ser melhorada com uso de técnicas mais sofisticadas de controle, como compensadores.

A efetividade da malha escolhida pode ser testada através de simulações não-lineares. Para esse caso, a simulação é perturbada com duas rajadas, as quais seguem as especificações da norma FAR 25.341-1, como especificado no Apêndice A.3. Ambas possuem velocidade máxima de 5 m/s e extensão de 500m (porém atuando em direções opostas), como apresentada na figura 5.13.

A criação das rajadas é apresentada abaixo,

```
from pyaat.wind import gust
rajada = gust(MaxSpeed = 5, xo = 500, xf = 1000, yo = -1000, yf = 1000)
```

Figura 5.13 – Rajadas de vento aplicadas na simulação não linear.



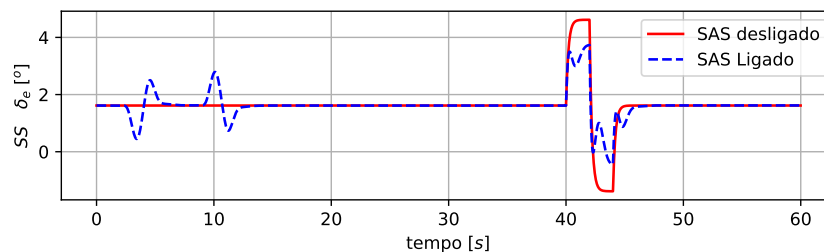
Fonte: Autor

```
rajada2 = gust(MaxSpeed = -5, xo = 2000, xf = 2500, yo = -1000, yf = 1000)
meusVentos = [rajada, rajada2]
```

onde MaxSpeed é a velocidade máxima do vento, x_o e y_o são as posições x_o e y_o de início da rajada, e x_f e y_f são as posições x_o e y_o de término da rajada.

Além de vento, é considerada uma deflexão de profundor em formato *doublet*, com amplitude de 3° e duração de 4s. A Figura 5.14 apresenta a deflexão $SS\delta_e$ do profundor para os casos de SAS ligado e desligado.

Figura 5.14 – Posição do profundor da aeronave com SAS ligado e desligado.

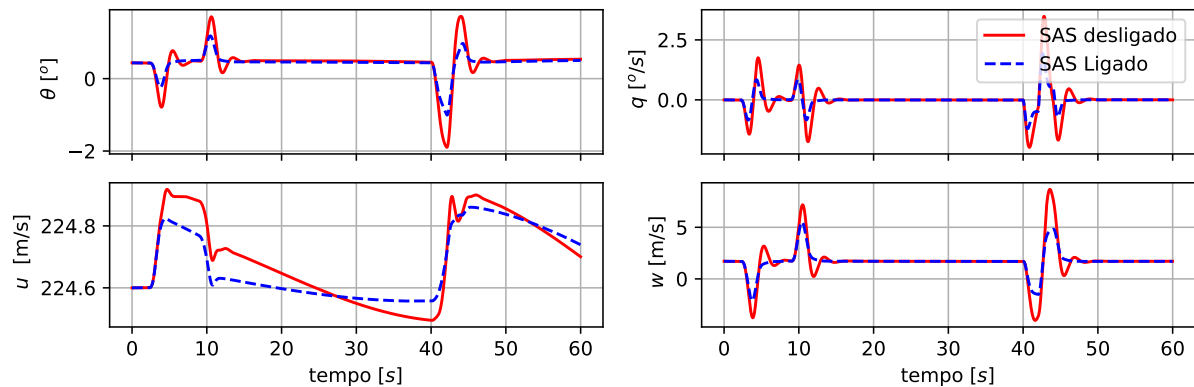


Fonte: Autor

Como pode ser observado, a dinâmica do atuador faz com que a ação de controle não seja aplicada instantaneamente. Além disso, para o caso onde o SAS está ligado, uma ação $SS\delta_e$ é requerida para compensar os efeitos dos ventos (próximo aos instantes 5s e 10s). Ainda, quando perturbação de profundor é aplicada, o SAS gera uma compensação, tornando a amplitude da deflexão consideravelmente menor. O comportamento dos estados da dinâmica longitudinal é apresentado na Figura 5.15.

Como pode ser observado, a velocidade angular q possui amplitude menor, o que, por consequência, gera variações menores de θ em torno da sua posição de equilíbrio. Além disso, a amplitude e duração das oscilações de w e u também são atenuadas, o que demonstra a funcionalidade da malha proposta.

Figura 5.15 – Comportamento dos estados da dinâmica longitudinal com SAS ligado e desligado.

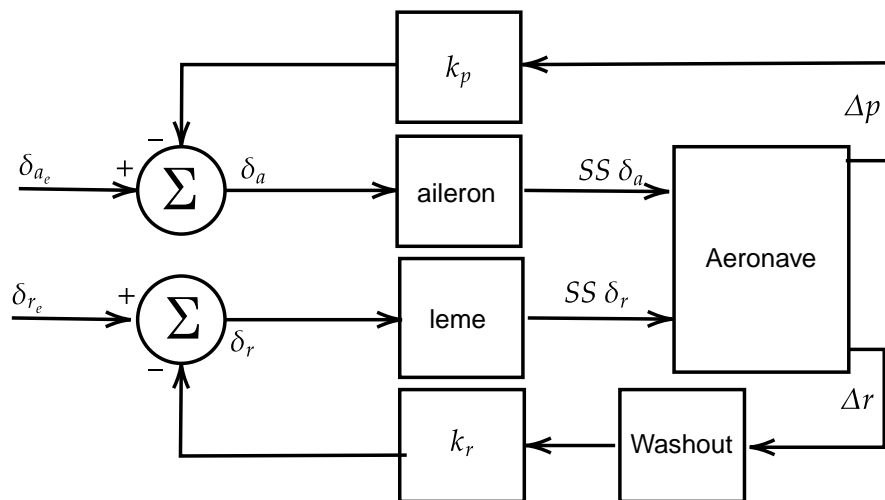


Fonte: Autor

5.9 SISTEMA DE AUMENTO DE ESTABILIDADE LÁTERO-DIRECIONAL

O modo *Dutch-roll* possui impacto significativo no conforto dos passageiros e tripulação, por isso, é desejável que atinja qualidade de voo nível 1. Para tal, Stevens, Lewis e Johnson (2016) propõe a malha de controle apresentada na Figura 5.16, onde Δp e Δr

Figura 5.16 – Malha do sistema de aumento de estabilidade látero-direcional.



Fonte: Adaptado de (STEVENS; LEWIS; JOHNNSON, 2016)

são os desvios de p e r em relação ao equilíbrio, δ_{a_e} e δ_{r_e} são as deflexões de aileron e leme na condição de equilíbrio, k_r e k_p são os ganhos de realimentação de Δr e Δp , e $SS\delta_a$ e $SS\delta_r$ são as deflexões de aileron e leme, respectivamente.

Além disso, também é incluído um filtro *washout* na realimentação da velocidade de guinada, o qual tem como intuito reduzir o efeito do SAS durante manobras em curva.

Esse filtro é do tipo passa-alta, fazendo com que oscilações de baixa frequência não sejam compensadas. A função de transferência do filtro é apresentada na Equação 5.5.

$$\text{washout} = \frac{s\tau_{wo}}{1 + s\tau_{wo}} \quad (5.5)$$

Na aplicação, utilizou-se uma constante de tempo τ_{wo} de 1s, como sugerido por Stevens, Lewis e Johnson (2016). Além disso, foi adotado o modelo de atraso simples com constantes de tempo τ_a e τ_r de 0.2s para ambos os atuadores, além de ganhos $k_r=0.2$ e $k_p=0.1$. A implementação é apresentada abaixo:

```
from pyaat.pyaatcontrol import P, loop, HighPassFilter
P1 = P(name = 'P1', gain = 0.1)
loop1 = loop(name='L1', Xmeasure='p', Xreference='p', command='Aileron')
loop1.loopControls = [P1]

washout = HighPassFilter(name = 'washout', tau = 1)
P2 = P(name = 'P2', gain = 0.2)
loop2 = loop(name='L2', Xmeasure='r', Xreference='r', command='Leme')
loop2.loopControls = [washout, P2]
closedloopcontrols = [loop1, loop2]
```

Nesse caso, loop2 contém dois componentes (filtro e o compensador proporcional). Quando isso ocorre, a saída de um componente é a entrada do seguinte, e a saída do ultimo componente é a ação de controle em si. Por consequência, a ordem de alocação dentro da lista afeta o resultado final.

Os autovalores da dinâmica látero-direcional com SAS assumem os valores apresentados na Tabela 5.8, enquanto a qualidade de voo do modo *Dutch-roll* é apresentado na Figura 5.17.

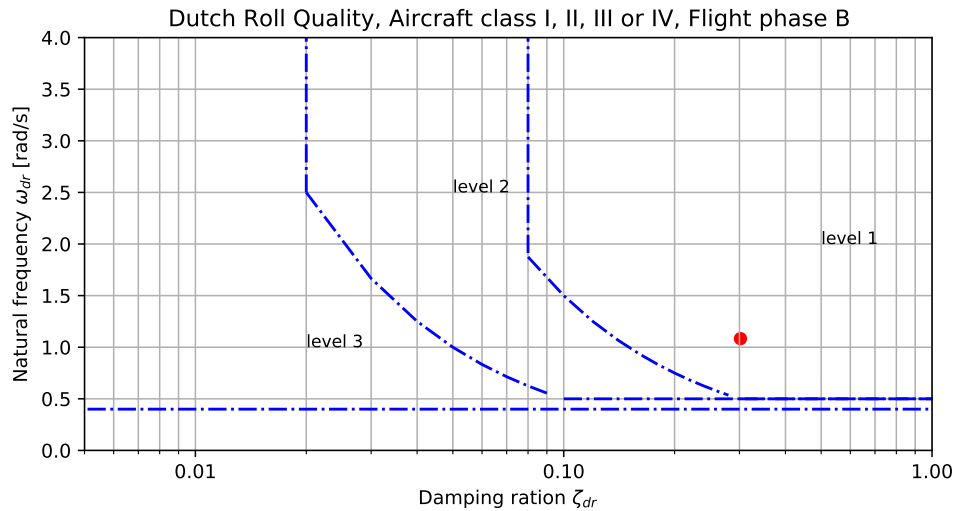
Tabela 5.8 – Autovalores da dinâmica látero-direcional antes e após inserção do SAS látero-direcional.

Modo	Autovalores com SAS	Autovalores sem SAS
Espiral	0.0026	0.0051
Rolamento puro	-3.6279	-3.8090
<i>Dutch-roll</i>	-0.3269 ± 1.0325	$-0.1218 \pm 1.3339j$

Fonte: Autor

Com o uso do SAS, o modo *Dutch-roll* passa ao nível 1 de qualidade de voo, enquanto os demais conservam suas qualidades no nível 1. Contudo, é observado por (STEVENS; LEWIS; JOHNNSON, 2016) que a presença do filtro *washout* gera um grande impacto na dinâmica do sistema, e qualquer melhoria (ou piora) só pode ser observada através de simulações não-lineares.

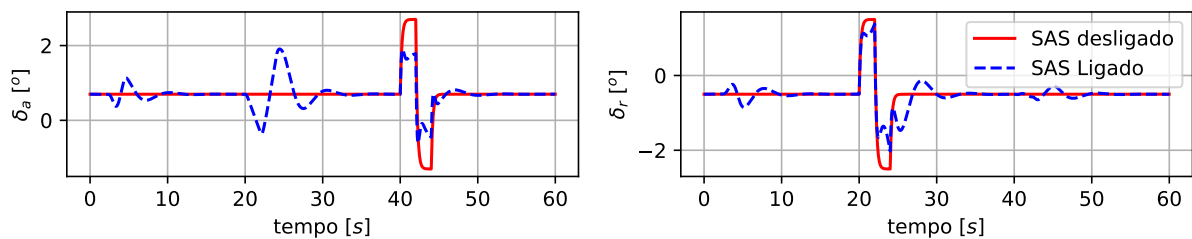
Figura 5.17 – Qualidade de voo do modo *Dutch-roll* do sistema em malha fechada.



Fonte: autor

Para esse caso, é considerada a condição de voo em curva coordenada com velocidade $\dot{\psi} = 6^\circ/\text{s}$, o que serve para verificar a efetividade do filtro *washsout*. Além disso, são consideradas perturbações de vento (idênticas às da seção 5.8) e são aplicadas perturbações do tipo *doublet* no aileron e leme, ambas com amplitude de 2° e duração de 4s, como apresentado na Figura 5.18.

Figura 5.18 – Deflexões de aileron e leme durante um voo em curva sujeito a perturbações.



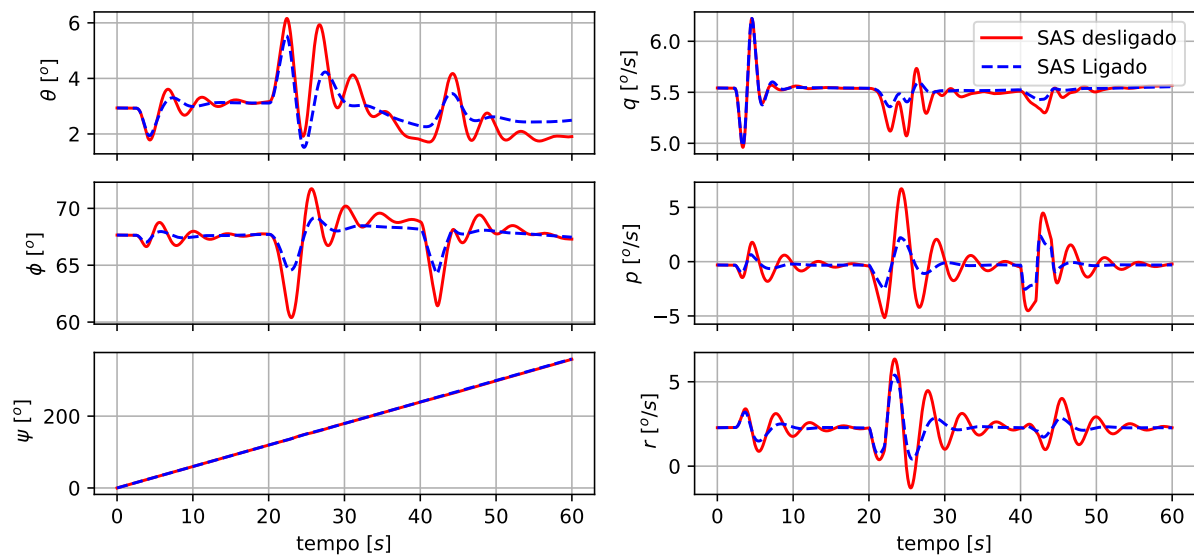
Fonte: autor

Como apresentado, ações de controle de leme e aileron são demandadas para compensar os efeitos dos ventos. Esse efeito é esperado visto que nessa condição de voo a aeronave possui um ângulo de rolamento não nulo e, por consequência, os ventos acabam gerando uma componente de velocidade lateral.

Pode ser visto ainda que, quando são aplicadas as perturbações em cada superfície de controle, a superfície não perturbada aplica uma ação de compensação, evidenciando que rolamento e guinada são acoplados nessa condição de voo. Ainda, devido à presença do SAS, as amplitudes das perturbações também são atenuadas.

A Figura 5.19 apresenta o comportamento temporal das variáveis angulares.

Figura 5.19 – Comportamento das variáveis angulares durante um voo em curva sujeito a perturbações.



Fonte: Autor

Como pode ser observado, as velocidades angulares p , q e r possuem amplitude de oscilação menor para o caso onde o SAS está ligado. Por consequência, as oscilações dos ângulos ϕ e θ também possuem amplitudes menores. Já para o caso de ψ , como o seu valor absoluto cresce continuamente, não é possível visualizar oscilações.

É interessante notar que, embora nenhuma compensação de q ou θ seja aplicada, esses estados são bastante afetados, o que, mais uma vez, demonstra o acoplamento das dinâmicas nessa condição de voo.

A posição da aeronave durante as simulações é apresentada na Figura 5.20.

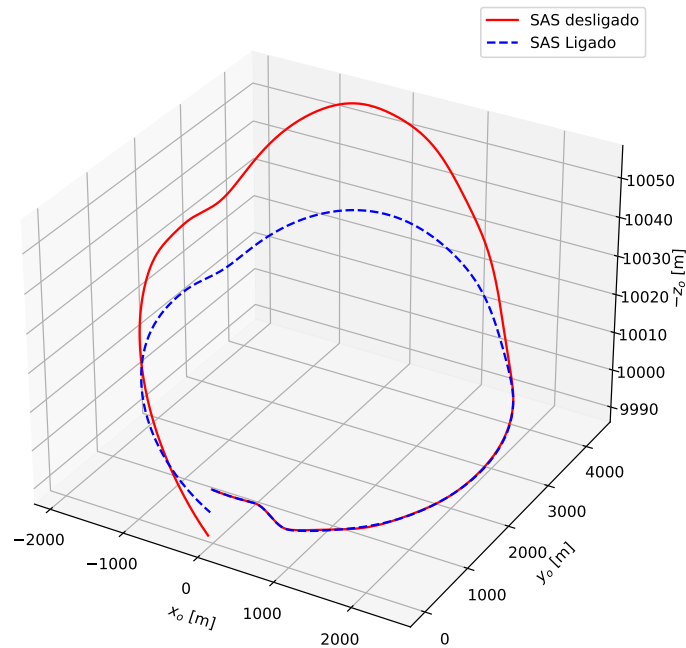
Como pode ser observado, em ambos os casos a aeronave completa a curva no período simulado, contudo, as amplitudes de oscilação na vertical são maiores para o caso de SAS desligado.

5.10 PILOTO AUTOMÁTICO DE ÂNGULO DE ARFAGEM

Pilotos automáticos (PAs) são categorias de controladores que dispensam a interferência de pilotos humanos i.e. uma vez engajados, são capazes de executar a tarefa demandada de forma completamente autônoma. Esse tipo de controlador é usual em atividades que demandam controle contínuo sobre determinados parâmetros, como por exemplo: manutenção de altitude, nivelamento de asas, entre outras.

Nessa seção é apresentado o projeto de um piloto automático de ângulo de arfa-

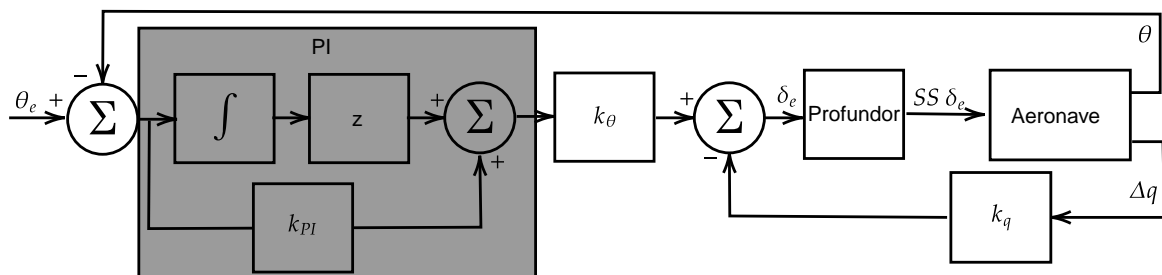
Figura 5.20 – Trajetória da aeronave com SAS ligado e desligado durante uma manobra de curva coordenada.



Fonte: Autor

gem, o qual tem como principal objetivo rastrear um determinado valor de θ . A malha de controle proposta é apresentada na Figura 5.21.

Figura 5.21 – Malha de rastreamento de ângulo de arfagem.



Fonte: Autor

Essa malha é composta por duas malhas com fechamentos sucessivos. A malha externa, aqui chamada de malha de θ , é composta por controlador proporcional integral (PI) seguido por um controle proporcional com ganho k_θ . A malha interna, por sua vez, é um SAS longitudinal com realimentação de q . A implementação da malha é apresentada abaixo:

```
from pyaatcontrol import P, loop, PI
```

```

PI_theta = PI(name = 'PI_theta', Kp = 1, zero = -0.1)
P_theta = P(name = 'P_theta', gain = 2.9)
loop_theta = loop(name='loop_theta', Xmeasure='theta', Xreference = 'theta',
command = 'Profundor')
loop_theta.loopControls = [PI_theta, P_theta]

Pq = P(name = 'Pq', gain = 0.4)
loop_q= loop(name='Lq', Xmeasure='q', Xreference='q', command='Profundor')
loop_q.loopControls = [Pq]

closedloopcontrols = [loop_theta, loop_q]

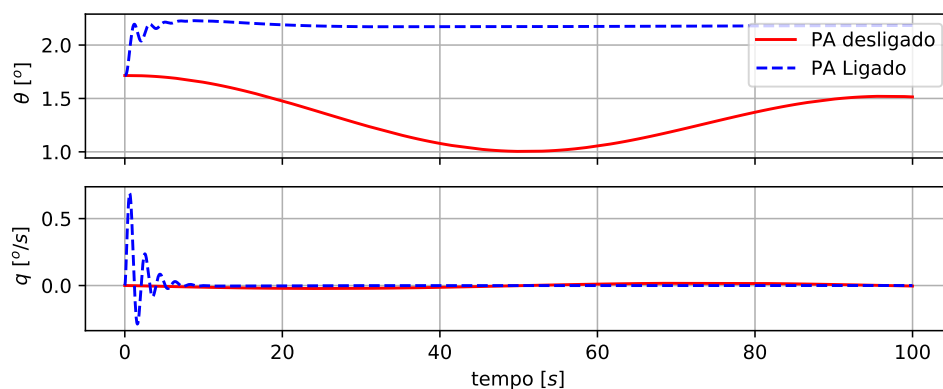
```

O controlador PI é disponibilizado no pacote de controle da biblioteca. Os parâmetros de inicialização são: nome, ganho proporcional e posição do zero. Para essa aplicação são considerados os valores $K_{PI}=1$ para o ganho proporcional e -0.1 para a posição do zero. A saída desse é aplicada em um compensador proporcional com ganho $k_\theta = 2.9$, o qual gera um sinal de comando de profundor. O SAS atua de forma independente, gerando um comando de profundor proporcional a q com ganho $k_q=0.4$.

O desempenho de PAs não está ligado às qualidades de voo, mas sim a critérios de missão, como erro de regime permanente e questões ligadas ao regime transiente. Por isso, o uso de simulações é parte fundamental do processo de análise desses compensadores.

Para essa análise, é considerado o voo de subida com velocidade constante de 5m/s iniciando aos 10000m de altitude e com velocidade u de 224.6m/s. Nesse caso, o ângulo θ de equilíbrio é 1.7° e θ comandado é 2.2° . O comportamento da velocidade q e ângulo θ são apresentados na Figura 5.22.

Figura 5.22 – Ângulo e velocidade de arfagem para um voo de subida com PA de arfagem.



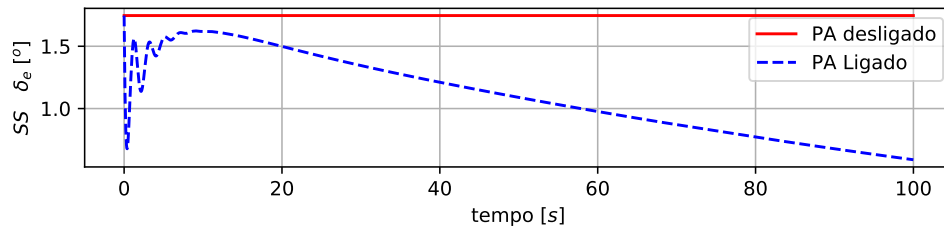
Fonte: Autor

Como pode ser observado, o compensador é capaz de atingir rapidamente o valor

comandado de θ . Ainda, como se trata de um compensador integral, o erro de regime permanente é nulo. Nos instantes iniciais o sinal de q é oscilatório, o que é natural visto que a velocidade de arfagem é naturalmente impactada por oscilações de θ . Essas oscilações atenuam ao passo que θ se aproxima do valor comandado.

O sinal do profundor é apresentado na Figura 5.23.

Figura 5.23 – Deflexão do profundor durante rastreo do ângulo de arfagem.

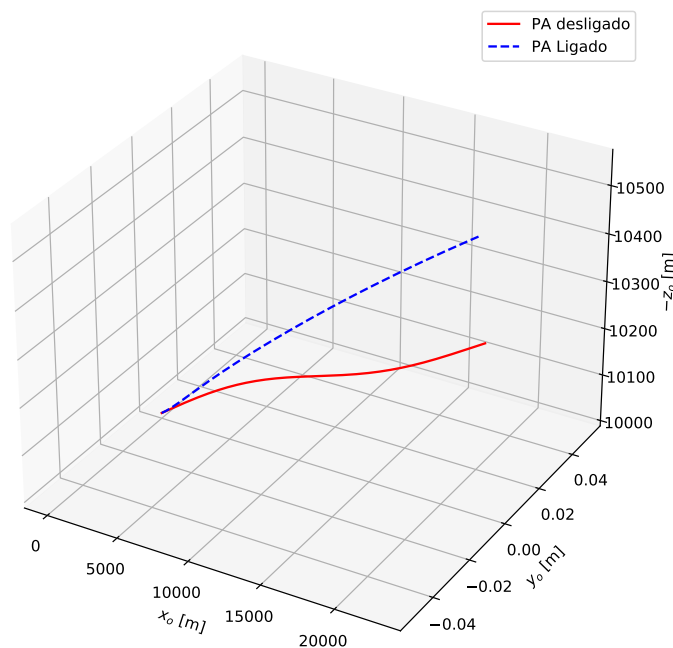


Fonte: Autor

Como pode ser observado, a deflexão é oscilatória nos instantes iniciais e, em seguida, diminui continuamente. O comportamento oscilatório pode ser explicado pela abrupta ação de controle nos instantes iniciais, gerada pelo grande erro de θ nesses instantes. Como a aeronave ganha altitude continuamente, a densidade do ar diminui e, por consequência, os momentos aerodinâmicos também, de tal modo que são necessárias deflexões cada vez menores para manter o ângulo desejado.

Finalmente, a trajetória da aeronave é apresentada na figura 5.24.

Figura 5.24 – Trajetória da aeronave em um voo de subida com rastreo de arfagem.



Fonte: Autor

Como pode ser observado, a trajetória desenvolvida pela aeronave quando auxiliada pelo PA é mais linear, embora não perfeitamente. Os desvios podem ser explicados pela alteração nas condições ambientais, as quais geram alterações no ângulo de ataque α . Para se obter uma trajetória em linha reta, a variável de controle deve ser o ângulo de trajetória $\gamma = \alpha - \theta$.

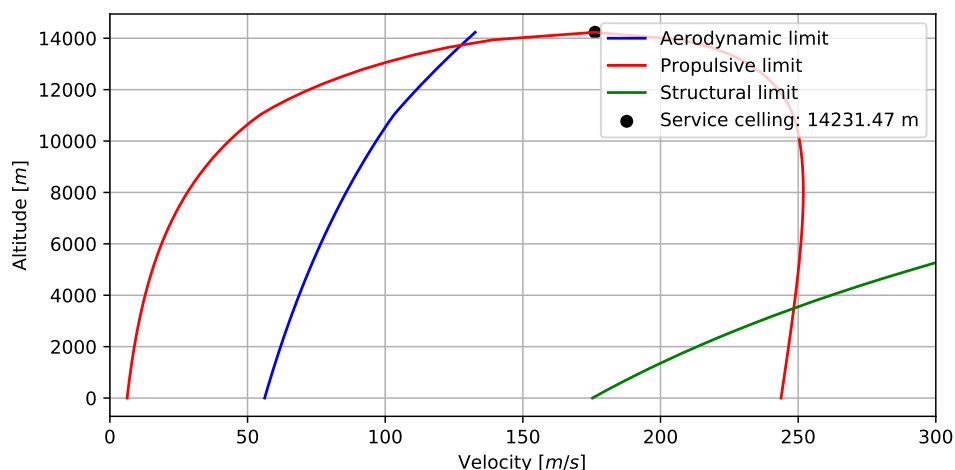
5.11 ENVELOPE DE VOO

O envelope de voo é um dos principais critérios de desempenho de aeronaves, visto que descreve os limites operacionais da mesma, o que, por fim, determina as missões as quais ela pode ser aplicada. Na biblioteca PyAAT, funcionalidade está disponível através da função `envelope` contida no pacote `performance`, a qual calcula os limites estrutural, propulsivo e aerodinâmico, bem como o teto de serviço. A aplicação da função é apresentada abaixo:

```
from pyaat.performance import Envelope
Envelope(meuSistema, limits = [0,400])
```

A entrada da função é o próprio sistema e, opcionalmente, os limites de velocidade do gráfico gerado. O resultado da função é apresentado na Figura 5.25.

Figura 5.25 – Envelope de voo da aeronave de asa fixa.



Fonte: Autor

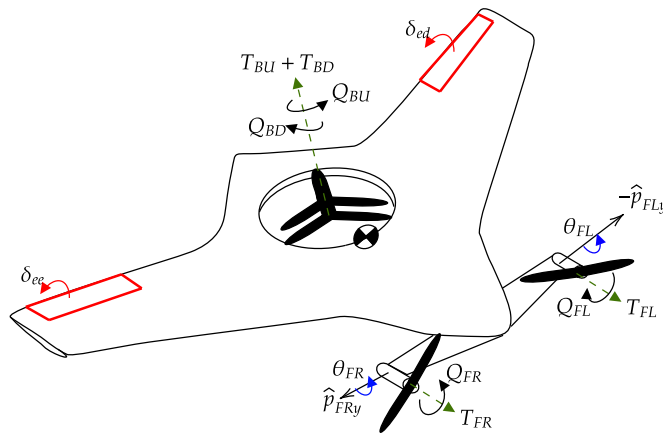
Nesse caso, o teto de serviço é de aproximadamente 14000m, e a velocidade máxima é aproximadamente 250m/s a uma altitude de 8000m.

6 AERONAVE DO TIPO ASA VOADORA COM CAPACIDADE DE POUSO E DECOLAGEM VERTICAL

A aeronave considerada nesse capítulo é uma adaptação do conceito apresentado por Bliamis et al. (2021), o qual consiste de uma asa voadora com capacidade de pouso e decolagem vertical. Suas características geométricas e aerodinâmicas são detalhadas no Apêndice D.

Do ponto de vista de controles, a aeronave conta com dois elevons, os quais ficam localizados nos bordos de fuga das asas. Essas superfícies são controladas individualmente através de deflexões δ_{ee} para o elevon esquerdo e δ_{ed} para o direito. Do ponto de vista propulsivo, a aeronave conta com 4 propulsores a hélice com motores elétricos. Dois deles ficam posicionados de forma coaxial no corpo da aeronave, com apontamento fixo $\theta_{pBU} = \theta_{pBL} = 90^\circ$ e $\psi_{pBU} = \psi_{pBL} = 0^\circ$. Ainda, o motor superior gira no sentido horário, enquanto o motor inferior gira no anti-horário. Os demais propulsores ficam dispostos simetricamente nas extremidades do *canaard* da aeronave, e possuem apontamentos θ_{FR} e θ_{FL} variáveis e independentes entre si. O propulsor esquerdo gira do sentido horário, enquanto o direito gira no sentido anti-horário, como apresentado na Figura 6.1.

Figura 6.1 – Variáveis de controle da aeronave não tripulada.



Fonte: Autor

Na figura 6.1, Q_{BD} e Q_{BU} são os torques gerados pelos propulsores posicionados no corpo da aeronave, enquanto T_{BD} e T_{BU} são as trações. Como os motores giram em sentidos opostos, o torque líquido é a diferença entre Q_{BD} e Q_{BU} , já a tração líquida é a soma de T_{BD} e T_{BU} .

Os motores frontais são idênticos, porém posicionados simetricamente e girando em sentidos opostos. Da Figura 6.1 temos que Q_{FL} e T_{FL} são o torque e tração do motor da esquerda, enquanto Q_{FR} e T_{FR} são as mesmas grandezas para o motor direito.

As velocidades de rotação comandadas para cada motor são normalizadas entre 0 e 1, sendo 0 para velocidade nula e 1 para a máxima. Nesse caso, os comandos são: δ_{BU} para o motor traseiro superior, δ_{BD} para o motor traseiro inferior, δ_{FL} para o motor frontal da esquerda e δ_{FR} para o motor frontal da direita.

Os coeficientes de tração escolhidos para os propulsores centrais são pequenos o suficiente de modo que a aeronave não tem capacidade de executar pouso e decolagem vertical utilizando somente eles. De fato, são considerados propulsores com maior capacidade na posição frontal, o que faz com que as manobras devam ser executadas através de uma combinação de comandos.

6.1 CRIAÇÃO DO SISTEMA

A criação dos modelos aerodinâmico, atmosfera e gravidade segue o mesmo processo detalhado no Capítulo 5. A implementação do modelo propulsivo é apresentada abaixo:

```

from pyaat.propulsion import Electric
FL = Electric(name = 'Motor FL', MaxSpeed =3000.0, ct =1.4341e-4,
cd =2.4086e-6, sense = 'clockwise')
FL.position = np.array([0.6, -0.32, .0])
FL.attitude = np.array([.0, np.radians(-90), .0])

FR = Electric(name = 'Motor FR', MaxSpeed =3000.0, ct =1.4341e-4,
cd =2.4086e-6, sense = 'counter-clockwise')
FR.position = np.array([0.6, 0.32, .0])
FR.attitude = np.array([.0, np.radians(-90), .0])

BU = Electric(name = 'Motor BU', MaxSpeed =2000.0, ct =0.43e-5,
cd =7.2e-7, sense = 'counter-clockwise')
BU.position = np.array([-0.25, 0, .0])
BU.attitude = np.array([.0, np.radians(-90), .0])

BD = Electric(name = 'Motor BD', MaxSpeed =2000.0, ct =0.43e-5,
cd =7.2e-7, sense = 'clockwise')
BD.position = np.array([-0.25, 0, .0])
BD.attitude = np.array([.0, np.radians(-90), .0])

minhaPropulsao = [FL, FR, BD, BU]

```

Como o modelo propulsivo considerado se aplica a aeronaves que voam a baixas altitudes e velocidades, ele é independente de parâmetros atmosféricos e condição de operação. Além disso, a variável de controle em motores elétricos é a velocidade de rotação, a qual controla indiretamente a tração. Os parâmetros de inicialização do modelo são: c_d é o coeficiente de torque, c_t coeficiente de tração, MaxSpeed velocidade máxima de rotação e sense sentido de rotação. A posição e atitude são definidas de maneira semelhante ao caso do motor a jato.

6.2 CÁLCULO DO BALANÇO

Graças a sua configuração não convencional, a aeronave é dotada da capacidade de voo sustentado, bem como voo em hover. As Tabelas 6.1 e 6.2 apresentam as combinações de estados e controles capazes de gerar ambas condições. Para o caso de hover, é considerado o voo pairado sobre um ponto fixo a uma altitude de 10m, enquanto que para o voo sustentado é considerada uma combinação de voo em curva (a uma taxa de $\dot{\psi} = 5^\circ/\text{s}$) com voo de subida (a uma taxa de 5m/s), onde a velocidade u aplicada é de 50m/s.

Tabela 6.1 – Estados para balanço em hover e manobra genérica.

Estado	V [m/s]	θ [°]	ϕ [°]	α [°]	β [°]	z_o [m]	p [°/s]	q [°/s]	r [°/s]
Hover	0.0	84.02	0.0	–	–	-10	0.0	0.0	0.0
Subida em curva	50.23	7.33	24.29	5.54	0.0	-10	-0.63	2.04	4.51

Fonte: Autor

Tabela 6.2 – Controles para balanço em hover e manobra genérica.

Comando	δ_{ee} [°]	δ_{ed} [°]	δ_{BU} [%]	δ_{BD} [%]	δ_{FL} [%]	δ_{FR} [%]	θ_{FR} [°]	θ_{FL} [°]
Hover	–	–	99.71	99.71	42.35	42.35	1.76	1.76
Subida em curva	5.73	-5.11	22.00	22.00	13.05	24.27	-24.67	66.68

Fonte: Autor

Como pode ser observado, para o caso de hover a aeronave se equilibra praticamente na vertical ($\theta = 84^\circ$). Isso ocorre porque essa condição é altamente dependente dos motores frontais. Caso seja utilizado um ângulo menor, os motores centrais não são capazes de contrabalancear o momento de arfagem gerado, como é evidenciado pela ação de controle requerida, que se aproxima da condição limite. Além disso, essa condição totalmente independe da posição dos elevons, o que é esperado.

Para o caso de voo combinado, existe uma completa assimetria nas demandas de ações de controles. Primeiramente, a deflexão dos elevons não é completamente simétrica, isso porque, além de executar a curva, também precisa compensar os momentos de

arfagem gerados pelos motores e momento aerodinâmico da asa (atividade geralmente desempenhada por um profundor).

Em aeronaves convencionais, uma curva coordenada requer comandos combinados de aileron e leme. Como a aeronave só dispõe de uma superfície aerodinâmica, essa função é desempenhada pela combinação entre elevons, apontamento dos motores e tração diferencial entre eles.

6.3 SISTEMA DE AUMENTO DE ESTABILIDADE

Na condição de manobra, os autovalores do sistema são:

- -1.8393
- -0.3202
- -1.8393
- $0.0838 \pm 0.2704i$
- -0.0801
- $0.021 \pm 0.0724i$
- 0.0001
- -0.0019

É interessante observar que o modo de período longo não possui o par complexo-conjugado, diferente do que geralmente é observado em aeronaves convencionais.

Como pode ser observado pelos autovalores, a aeronave é instável e, portanto, precisa de um SAS para ser capaz de desempenhar um voo contínuo. Contudo, como os estados e controles são altamente acoplados, a tarefa de implementar um SAS baseado em malhas de controle do tipo SISO é bastante complexa, visto que cada nova malha pode gerar efeitos indesejados nas outras dinâmicas. Nesse contexto, a implementação de um SAS do tipo MIMO é a alternativa mais viável para se garantir um desempenho adequado.

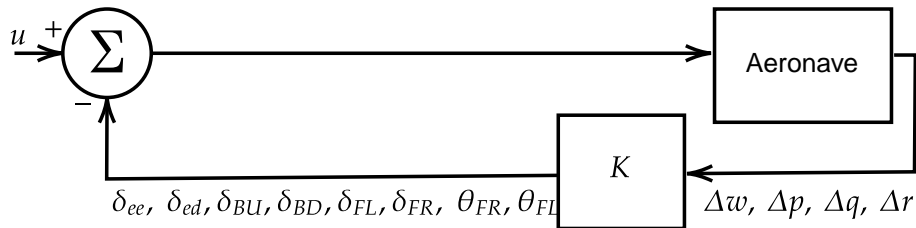
Como a biblioteca PyAAT não oferece suporte para esse tipo de controlador, a metodologia adotada é a utilização do modelo criado pela biblioteca em uma simulação em malha fechada fora da biblioteca. Isso é realizado por meio do método `ExternalSimulation`, como apresentado abaixo:

```
Xed = meuSistema.ExternalSimulation(t, X, U)
```

Esse método recebe como entrada o tempo de simulação, estado atual e ações de controle, e retorna as derivadas dos estados, as quais são integradas fora da biblioteca. Além de flexibilizar a simulação, isso permite a integração dessas equações com quaisquer outras desejadas pelo usuário, além de implementação de estratégias de guiagem e navegação.

A malha de controle implementada é apresentada na Figura 6.2.

Figura 6.2 – SAS do tipo MIMO da aeronave não tripulada.



Fonte: Autor

Como pode ser visto, nessa malha são medidos w , p , q e r , e o desvio de cada variável em relação ao equilíbrio gera uma ação em cada variável de controle.

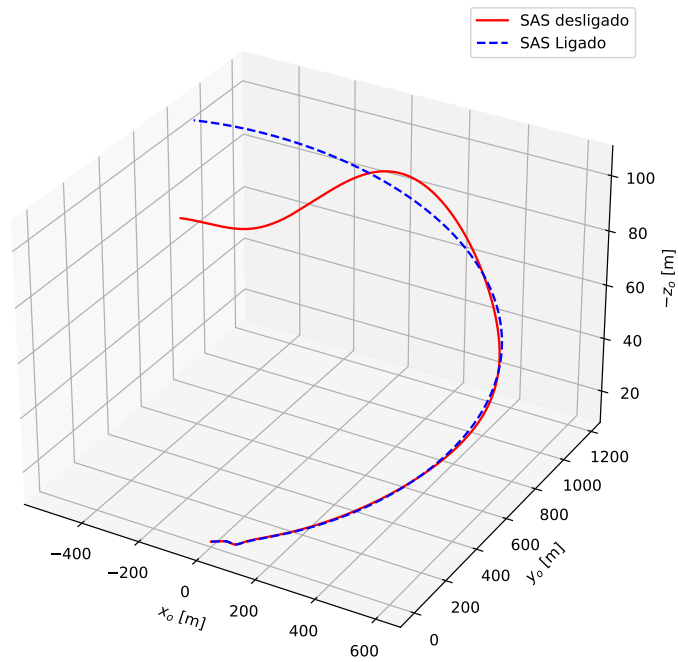
A matriz de ganhos K , apresentada na Equação 6.1, é calculada minimizando um índice de desempenho quadrático, técnica essa conhecida como Regulador Linear Quadrático (LQR). Embora não seja a condição de voo em estudo, os ganhos foram calculados para a condição de voo reto nivelado, tendo como base a maior generalidade dos ganhos gerados nessas condições. Mais detalhes sobre o método são apresentados por Ogata (2010).

$$\mathbf{K} = \begin{bmatrix} & \mathbf{w} & p & q & r \\ \delta_{ee} & 0.013 & -7.825 & 26.8159 & -27.3349 \\ \delta_{ed} & 0.013 & 7.825 & 26.8159 & 27.3349 \\ \delta_{FL} & -2.1074 & 6.6729 & 2.9362 & -68.156 \\ \theta_{FL} & -0.1878 & -0.0047 & -0.0191 & -0.7822 \\ \delta_{FR} & -2.1074 & -6.6729 & 2.9362 & 68.156 \\ \theta_{FR} & -0.1878 & 0.0047 & -0.0191 & 0.7822 \\ \delta_{BU} & -1.1764 & -0.2819 & 1.7823 & -0.3113 \\ \delta_{BL} & -1.1764 & 0.2819 & 1.7823 & 0.3113 \end{bmatrix} \quad (6.1)$$

A verificação da eficácia da malha implementada é feita através de simulações não lineares. A condição considerada é manobra de curva com subida, onde a aeronave também é submetida a vento e perturbações de elevon esquerdo. A rajada considerada tem velocidade máxima de 5m/s, e tem efeito entre as posições $x_o = 40\text{m}$ e 100m. Já a perturbação de elevon é do tipo *doublet*, tem magnitude de 4° e duração de 1s. A trajetória da aeronave é apresentada na Figura 6.3 enquanto as ações de controle são apresentadas

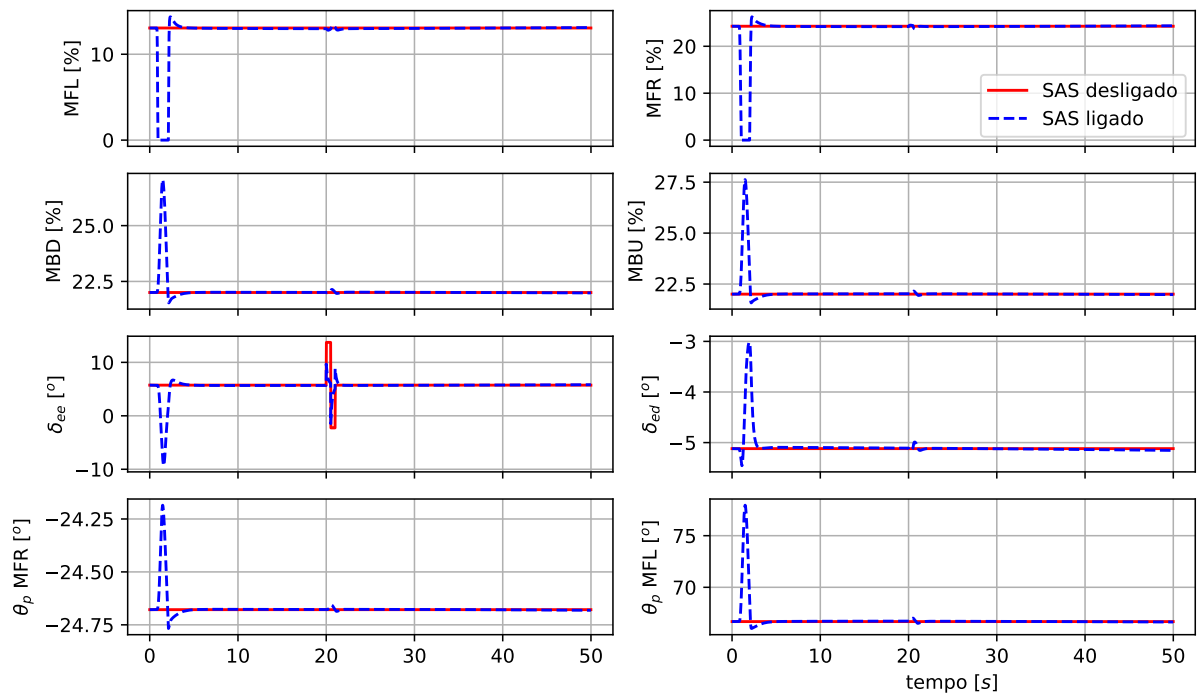
na Figura 6.4.

Figura 6.3 – Trajetória da aeronave não tripulada com e sem SAS.



Fonte: Autor

Figura 6.4 – Ações de controle da aeronave não tripulada com e sem SAS.



Fonte: Autor

Como pode ser observado no gráfico de trajetória, a aeronave não é capaz de manter a condição de voo para o caso onde o SAS está desligado, enquanto o sistema estabilizado sofre apenas pequenas oscilações nos instantes iniciais, coincidindo com a região de aplicação dos ventos.

As ações de controle na condição de equilíbrio são totalmente assimétricas, e mantêm esse comportamento durante a simulação. Quando a aeronave é submetida à rajada, todas as ações de controle atuam conjuntamente para neutralizar os efeitos. Além disso, é interessante observar que, quando é aplicada a perturbação no elevon esquerdo, apenas uma pequena compensação é requerida no elevon direito e apontamento dos motores, visto que a compensação gerada no próprio elevon afetado praticamente neutraliza o efeito da perturbação.

7 CONCLUSÕES

Esse trabalho apresentou o desenvolvimento e aplicação de uma biblioteca Python de código aberto, a qual implementa métodos e modelos que facilitam o processo de modelagem, análise e simulação de aeronaves. Mais precisamente, a biblioteca disponibiliza uma plataforma modular, onde o sistema é construído a partir de blocos construtivos (submodelos). Além disso, a biblioteca dispõe de diversos desses submodelos nativamente implementados, os quais podem ser adaptados pelo usuário a casos específicos. Em soma, a biblioteca implementa diversas funções, as quais aplicam técnicas consolidadas em aeronáutica, como linearização, separação modal, construção do envelope de voo, ferramentas para análise de qualidades de voo e simulações não lineares.

São apresentadas duas aplicações práticas da biblioteca. A primeira aplicação é um caso canônico tratando de uma aeronave de grande porte, a qual tem como objetivo a validação da biblioteca e demonstração das funções relacionadas a esse tipo de aeronave. Nesse sentido, a modelagem é realizada usando os submodelos presentes na biblioteca, onde fica evidenciada a facilidade com que modelos podem ser definidos, visto que, a grosso modo, esse processo demanda apenas a definição de um conjunto de valores e nenhuma função precisa ser explicitamente programada. O cálculo do balanço é realizado em diversas condições (subida, cruzeiro e curva) utilizando método *trimmer*. Em todos os casos são obtidos fisicamente coerentes, sendo que, é observada similaridade até a primeira casa decimal com valores obtidos em MATLAB. A separação modal também é apresentada, onde a eficiência do método é evidenciada na similaridade entre os autovalores do sistema completo e dos modelos de ordem reduzida. São implementadas diferentes estruturas de controle, que vão de sistemas de aumento de estabilidade ao caso de piloto automático, sendo que, nesse processo são adicionadas diversas dinâmicas auxiliares ao sistema (sensores e atuadores). A funcionalidade dessas técnicas é verificada através de simulações não lineares considerando perturbações.

O segundo caso trata de uma aeronave conceitual com arquitetura não convencional, a qual junta características de aeronaves de asa fixa e multirrotores. Esse caso trata de uma aeronave naturalmente instável, sendo necessário um sistema de aumento de estabilidade para compensar essa deficiência. Dadas as características da aeronave, a forma dessa malha é do tipo MIMO, o que requer o uso do modelo criado em uma simulação externa à biblioteca. Essa funcionalidade expande consideravelmente o uso da biblioteca, visto que, usando essa abordagem, a biblioteca passa a ser compatível com qualquer *software* compatível com a linguagem Python.

De modo geral, foi evidenciada a simplicidade de construção de um modelo utilizando a biblioteca. Modificações significativas, como inserção de novas superfícies de controle ou alteração de modelos podem ser realizadas em poucas linhas de código e sem

preocupação com adequação das equações do sistema, visto que a biblioteca faz isso automaticamente. Além disso, graças ao formato modular, todos os módulos implementados pelo usuário podem ser facilmente reaproveitados.

7.1 TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se:

- Embora o sistema possa conter diversas dinâmicas adicionais relacionadas aos estados e controles, isso não se aplica às outras variáveis. Por consequência, não é possível acomodar dinâmicas referentes à geometria, aerodinâmica, propulsão, etc, o que é interessante para propósitos de pesquisa. Nesse sentido, é previsto que parte da biblioteca será reformulada aplicando o conceito de *Input/output systems* disponibilizado pela biblioteca Python-control, o que facilita a junção de diversos sistemas dinâmicos.
- Por hora, existe pouca flexibilidade na implementação do sistema propulsivo. As únicas variáveis de controle são apontamento e tração, enquanto podem existir outras variáveis de interesse (como passo de hélices, por exemplo). Futuras implementações podem considerar controles distintos dependendo do modelo propulsivo adotado. Nesse caso, também é solucionado o problema de variáveis de controle desnecessárias, como por exemplo a presença do controle de apontamento de motores em aeronaves com motores fixos.
- Atualmente os parâmetros dos blocos devem ser fornecidos pelo usuário, enquanto existem diversas ferramentas de código aberto que geram esses valores automaticamente. Nesse sentido, é interessante buscar formas de integração entre essas soluções e a biblioteca. Por exemplo, pode-se integrar as ferramentas Digital DATCOM e Pytornado ao bloco *aircraft*, ou a ferramenta python-propeller ao bloco *propulsion*.
- O número de modelos e métodos disponíveis na biblioteca ainda é bastante limitado. Nesse sentido, sugere-se a implementação de novos modelos de vento, propulsores, etc. Além disso, diversas funcionalidades podem ser acrescentadas, principalmente no que diz respeito às análises de desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

AIRBUS. **Getting to grips with Aircraft Performance**. Blagnac, FR: Airbus Customer Services, 2002. 216 p. (Flight Operations Support and Line Assistance).

BLIAMIS, C. et al. Aerodynamic and stability analysis of a vtol flying wing uav. In: **IOP Conf. Series: Materials Science and Engineering**. [S.l.]: IOP Publishing, 2021. Doi:10.1088/1757-899X/1024/1/012039.

CHUNG, P.-H.; MA, D.-M.; SHIAU, J.-K. Design, manufacturing, and flight testing of an experimental flying wing uav. **Applied Sciences**, v. 9, n. 3043, p. 22, 2019.

COOK, M. V. **Fligh Mechanics Principles: A Linear Systems Approach to Aircraft stability and Control**. Oxford: Butterworth-Heinemann, 2007. Second. 491 p. (Elsevier Aerospace Engineering Series, Second).

Dassault Systemes. **DYMOLA Systems Engineering :Multi-Engineering Modeling and Simulation based on Modelica and FM**. 2021. Disponível em: <<https://www.mathworks.com/products/aerospace-toolbox.html>>.

EXTENSION, B. **11 Most In-Demand Programming Languages in 2021**. 2021. Disponível em: <<https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/>>.

GITHUB. **PYPL PopularitY of Programming Language**. 2021. Disponível em: <<https://pypl.github.io/PYPL.html>>.

ISAAC, M. S. A. et al. Mathematical modeling and designing a heavy hybrid-electric quadcopter, controlled by flaps. **Unmanned Systems**, v. 10, p. 14–18, 2021.

KUANTAMA, E.; TARCA, R. Quadcopter thrust optimization with ducted-propeller. In: **Annual Session of Scientific Papers IMT ORADEA**. [S.l.]: EDP Sciences, 2017. Doi:10.1051/mateconf/201712601002.

MATHWORKS. **Aerospace Toolbox**. 2021. Disponível em: <<https://www.mathworks.com/products/aerospace-toolbox.html>>.

McDonnell Douglas Astronautics Company. **THE USAF STABILITY AND CONTROL DAT-COM:Users Manual**. St. Louis, 1999. AFFDL-TR-79-3032 Volume I, 147 p. Acesso em 28 jan. 2022. Disponível em: <http://wpage.unina.it/agodemar/DSV-DQV/Digital_Datcom_Users_Manual_1.2.pdf>.

National Aeronautics and Space Administration. **U.S. Standard Atmosphere, 1976**. Washington, D.C., 1976. NASA-TM-X-74335, 241 p. Acesso em 02 dez. 2021. Disponível em: <<https://ntrs.nasa.gov/search.jsp?R=19770009539>>.

_____. **Public Domain Aeronautical Software**. 2021. Disponível em: <<https://www.pdas.com/contents16.html>>.

NEELAMKAVIL, F. **Computer Simulation and Modelling**. [S.l.]: Wiley & sons, 1987. 324 p.

OGATA, K. **Modern Control Engineering**. Fifth. New Jersey: Pearson, 2010. 906 p.

POLIVANOV, P. A.; SIDORENKO, A. A. Quadcopter propeller characteristics in the oblique flow. In: **Journal of Physics: Conference Series**. Atlanta: IOP Publishing, 2019. Doi:10.1088/1742-6596/1404/1/012095.

RAYMER, D. P. **Aircraft Design: A Conceptual Approach**. New York: Birkhauser, 1992. 760 p. (AIAA Education Series).

SADRAEY, M. H. **Aircraft Performance: An Engineering Approach**. New York: CRC Press, 2017. 572 p. (AIAA Education Series).

Siemens Industry Software. **Simcenter Amesim**. 2021. Disponível em: <<https://www.plm.automation.siemens.com/global/pt/products/simcenter/simcenter-amesim.html>>.

SILVA, A. L. da; PAGLIONE, P.; YONEYAMA, T. Conceptual exible aircraft model for modeling, analysis and control studies. In: **AIAA Atmospheric Flight Mechanics Conference**. Toronto, Canada: [s.n.], 2010.

STEVENS, B. L.; LEWIS, F. L.; JOHNNSON, E. L. **Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems**. New Jersey: John Wiley & Son, 2016. Third. 733 p.

TEWARI, A. **Atmospheric and Space Flight Dynamics**. Washington, D.C.: American Institute of Aeronautics and Astronautics, 2007. 543 p. (Modeling and Simulation in Science, Engineering and Technology).

U.S. Department of Transportation. **Advisory Circular No. 25.241-1**. [S.I.], 2014. ANM-115, 24 p. Acesso em 14 dez. 2021. Disponível em: <https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_25_341-1.pdf>.

APÊNDICE A – MODELOS MATEMÁTICOS

A.1 – MODELOS GRAVITACIONAIS

A.1.1 – Modelo Gravitacional de Newton

O modelo gravitacional de Newton expressa a força gravitacional entre dois corpos, tratando ambos como massas pontuais. Nesse modelo, a força gerada em cada um dos corpos possui uma única componente não nula, que aponta na direção da outra massa. De acordo com o modelo, a expressão do vetor campo gravitacional é expressa pela Equação A.1.

$$\vec{g} = -GM_T \frac{\vec{r}}{r^3} \quad (\text{A.1})$$

onde $G = 6,6729 \times 10^{-11}$ é a constante gravitacional, M_T é a massa da Terra e \vec{r} é a posição da aeronave com respeito ao centro o planeta.

O vetor força gravitacional, por sua vez, é expresso de acordo com a Equação A.2.

$$\vec{F}_g|_{\mathcal{I}} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (\text{A.2})$$

onde m é a massa da aeronave.

A.1.2 – Modelo Gravitacional de Alta Ordem

O Planeta Terra não é uma esfera perfeita, possuindo um diâmetro maior na região da Linha do Equador. Essa não esfericidade pode causar impactos significativos quando considerados longos espaços de tempo, especialmente para veículos em grandes altitudes. Uma abordagem é considerar o planeta como um corpo axissimétrico, como apresentado em (TEWARI, 2007). Desse modo, o potencial gravitacional é dado pela expressão A.3.

$$\Phi(r, \phi) = \frac{GM_T}{r} \left\{ 1 - \sum_{n=2}^{\infty} \left(\frac{R}{r} \right)^n J_n P_n(\cos \phi) \right\} \quad (\text{A.3})$$

onde ϕ é a co-latitude, J_n são as constantes de Jeffrey (únicas para cada planeta), P_n é o polinômio de Legendre relacionado ao harmônico esférico de ordem n , o qual é dado pela

expressão A.4.

$$P_n(\cos \phi) = \frac{(2n-1) \cos \phi P_{n-1}(\cos \phi) - (n-1) P_{n-2}(\cos \phi)}{n} \quad (\text{A.4})$$

Nesse trabalho é considerado um harmônico esférico de ordem 4, o qual está relacionado às constantes de Jeffrey $J_2 = 0,00108263$, $J_3 = -0,00000254$ e $J_4 = -0,00000161$.

Desse modo, o vetor campo gravitacional passa a ser dado pela expressão A.5.

$$\vec{g} = -\frac{\partial \Phi}{\partial r} \vec{i}_r - \frac{\partial \Phi}{\partial \phi} \vec{i}_\phi \quad (\text{A.5})$$

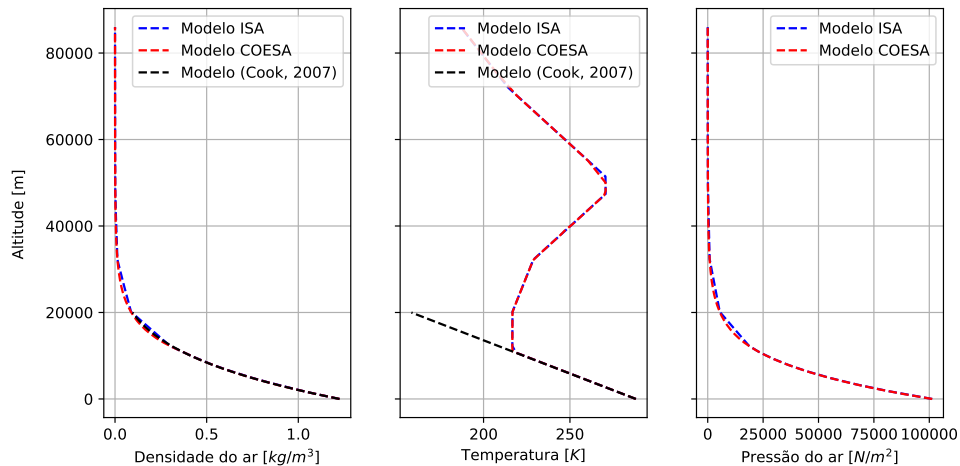
O vetor força gravitacional, por sua vez, é dado por A.6.

$$\vec{F}_g|_{\mathcal{I}} = m\vec{g} \quad (\text{A.6})$$

A.2 – MODELOS ATMOSFÉRICOS

O comportamento de uma aeronave depende fortemente do ambiente na qual ela se encontra. O modelo atmosférico é responsável por relacionar certas propriedades ambientais (temperatura, pressão e densidade do ar) a uma certa altitude geométrica. Três modelos estão disponíveis na biblioteca, sendo eles: *International Standard Atmosphere - ISA*, *U.S Standard Atmosphere 1976 - COESA* e um modelo simplificado apresentado por Cook (2007). Os dois primeiros modelos são válidos para altitudes de até 86km enquanto o último não é recomendado para altitudes superiores a 30 mil pés. O comportamento de cada modelo é apresentado na Figura A.1.

Figura A.1 – Modelos atmosféricos presentes na biblioteca PyAAT.



A.2.1 – Atmosfera ISA

Na implementação do modelo *International Standard Atmosphere*, os valores de temperatura e pressão são interpolados para uma determinada temperatura a partir dos valores apresentados em (AIRBUS, 2002). Como apresentado na Tabela A.1.

Tabela A.1 – Derivadas de estabilidade da aeronave conceitual.

Altitude geométrica Ht [m]	temperatura T [K]	Pressão P [Pa]
0,0	288,15	101300,0
1524,0	278,25	84300,0
3048,0	268,35	69700,0
6096,0	258,45	57200,0
4572,0	248,55	46600,0
7620,0	238,65	37600,0
9144,0	228,75	30100,0
10668,0	218,85	23800,0
12192,0	216,65	18800,0
20063,0	216,64	5474,9
32162,0	228,65	868,02
47350,0	270,65	110,91
51413,0	270,65	66,939
71802,0	214,65	3,9564
86000,0	186,87	0,3734

Fonte: Adaptado de (AIRBUS, 2002)

A densidade padrão em uma determinada altitude é calculada de acordo com a Equação A.7, onde assume-se o ar como um gás ideal.

$$\rho = \frac{P}{RT} \quad (\text{A.7})$$

onde R é a constante universal dos gases ideais $R = 287.053 \text{ J/kg/K}$.

A.2.2 – Atmosfera COESA

O modelo COESA é implementado de forma semelhante ao modelo ISA, porém seguindo os valores tabelados em (National Aeronautics and Space Administration, 1976), com amostras em espaçadas de 500 metros no intervalo entre -1000,0m e 85500m.

A.2.3 – Modelo atmosférico simplificado

O modelo atmosférico apresentado por (COOK, 2007) é válido até a altitude geométrica de 36000 pés. De acordo com o modelo, a temperatura é descrita de acordo com a Equação A.8.

$$T = 288.16 + I_r H t \quad (\text{A.8})$$

onde $I_r = -0,0065$ é a derivada de temperatura.

A densidade do ar, por sua vez, é dada pela Equação A.9.

$$\rho = 1.225 \left(\frac{T}{288.16} \right)^{\left[-\frac{g}{I_r R} + 1 \right]} \quad (\text{A.9})$$

O modelo apresentado por Cook (2007) não descreve o comportamento da pressão, e, nesse caso, é considerado o modelo ISA ao nível do mar.

A.3 – MODELO DE RAJADA FAR 25.341-1

O comportamento da aeronave na presença de ventos é importante no estudo da estabilidade, controle e na certificação de aeronaves. Nesse sentido, o modelo adotado na biblioteca é o extraído da norma de certificação FAR 25.341-1 (U.S. Department of Transportation, 2014), a qual considera uma rajada com componente de velocidade unicamente vertical, dada pela Equação A.10.

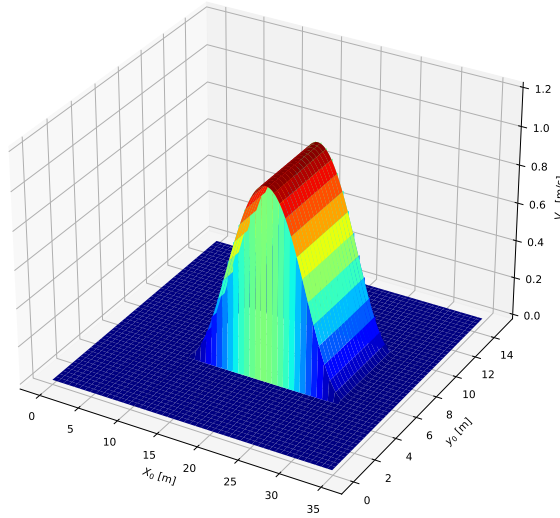
$$V_w = \begin{cases} -\frac{V_0}{2} \left[1 - \cos \left(\pi \frac{x_0 - x_{0w}}{H_w} \right) \right], & \text{se } x_{0w} \leq x_0 \leq x_{fw} \text{ e } y_{ow} \leq y_o \leq y_{fw} \\ 0, & \text{caso contrário} \end{cases} \quad (\text{A.10})$$

onde V_w é a velocidade vertical do vento, V_0 é a velocidade máxima do vento, x_{0w} , x_{fw} , y_{ow} e y_{fw} limitam a região afetada e H_w representa o comprimento da rajada. A figura A.2 ilustra o comportamento de uma rajada com parâmetros $H_w = 15\text{m}$, $V_0 = 1\text{m/s}$, $x_{0w} = 10\text{ m}$, $x_{fw} = 30\text{m}$, $y_{ow} = 5$ e $y_{fw} = 10$.

A.4 – SENSORES E ATUADORES

O equacionamento apresentado na Sessão 2.2 diz respeito unicamente ao movimento de um corpo rígido em 6GDL. Contudo, outras dinâmicas podem acarretar impacto significativo no comportamento da aeronave e devem ser observadas, como é o caso das da presença de sensores e atuadores.

Figura A.2 – Modelos de rajada de acordo com a norma FAR25341-1.



Fonte: Autor

Os atuadores são modelados como um simples atraso de primeira ordem, como expresso pela função de transferência da Equação A.11.

$$\frac{\delta_{at}}{\mu_{at}} = \frac{1/\tau_{at}}{s + 1/\tau_{at}} \quad (\text{A.11})$$

onde δ_{at} é a valor da deflexão do atuador, μ_{at} é o valor comandado, e τ_{at} é a constante de tempo do atuador.

No caso dos sensores, geralmente o maior impacto está associado ao filtro utilizado para o processamento do ruídos. Por esse motivo, sensores são modelados como filtros passa-baixa, como apresentado na Equação A.12.

$$\frac{x_f}{x_r} = \frac{1/\tau_x}{s + 1/\tau_x} \quad (\text{A.12})$$

onde x_f é a valor da medido, x_r é o valor real do estado, e τ_x é a constante de tempo do filtro.

A.5 – MATRIZES DE ROTAÇÃO

O aprontamento relativo (ou atitude) entre dois sistemas de coordenadas pode ser representado de forma intuitiva através da parametrização dos ângulos de Euler. Essa parametrização consiste na realização de uma sequência máxima de três rotações sucessivas em torno de eixos não consecutivos (TEWARI, 2007). Note que a sequencia de

rotações escolhida é um parâmetro importante, dado que rotações não são comutativas, ou seja, a ordem das rotações afeta o apontamento final.

Considere um sistemas de coordenadas $\mathcal{A} : \{A, \hat{a}_x, \hat{a}_y, \hat{a}_z\}$, que sofre uma sequência de rotações ψ , θ e ϕ na sequência 3-2-1, temos:

- 1ª rotação $C_3(\psi)$: ψ em torno de \hat{a}_z , gerando um sistema $\mathcal{A}' : \{A, \hat{a}_{x'}, \hat{a}_{y'}, \hat{a}_z\}$;
- 2ª rotação $C_2(\theta)$: θ em torno de $\hat{a}_{y'}$, criando um sistema $\mathcal{A}'' : \{A, \hat{a}_{x''}, \hat{a}_{y''}, \hat{a}_{z''}\}$;
- 3ª rotação $C_1(\phi)$: ϕ em torno de $\hat{a}_{x''}$, criando o sistema final $\mathcal{A}''' : \{A, \hat{a}_{x'''}, \hat{a}_{y'''}, \hat{a}_{z'''}\}$;

Cada uma dessas rotações elementares é representada matematicamente pela multiplicação do sistema original por uma matriz de rotação elementar de dimensão 3X3. Essa matriz realiza operações trigonométricas que convertem cada uma das componentes do sistema original para o sistema final, sendo que cada uma das rotações é definida como apresentado nas Equações A.13, A.14 e A.15.

$$C_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (\text{A.13})$$

$$C_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (\text{A.14})$$

$$C_3(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.15})$$

Pode-se aplicar uma sequência de rotações simplesmente multiplicando o sistema intermediário pela rotação seguinte, ou de forma equivalente, multiplicar o sistema original pela multiplicação das matrizes de rotação utilizadas. Para o caso da sequência de rotação 3-2-1 fica como apresentado na equação A.16.

$$A''' = C_1(\phi)C_2(\theta)C_3(\psi)A \quad (\text{A.16})$$

que é equivalente ao expresseo na equação A.17.

$$A''' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} A \quad (\text{A.17})$$

Ou ainda, multiplicando as matrizes, tem-se a Equação A.18.

$$A''' = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ \sin \phi \cos \psi \sin \theta - \cos \theta \sin \psi & \sin \phi \sin \psi \sin \theta + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi & \cos \phi \sin \psi \sin \theta - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} A \quad (\text{A.18})$$

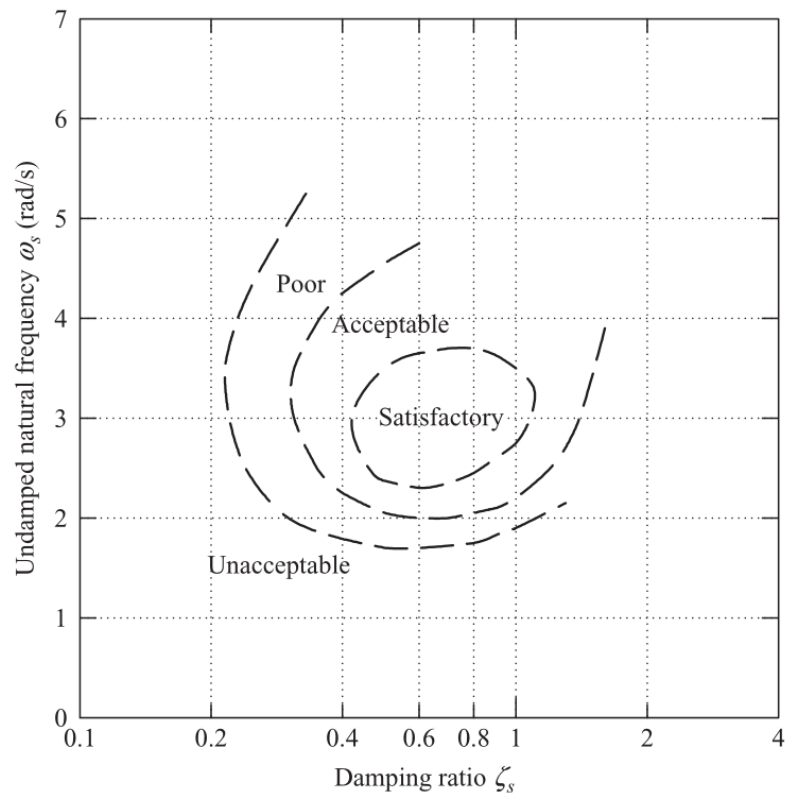
O mesmo processo pode ser seguido para qualquer sequência de ângulos.

APÊNDICE B – CRITÉRIOS DE QUALIDADE DE VOO

B.1 – QUALIDADES DE VOO DO MODO LONGITUDINAL DE PERÍODO CURTO

O modo de período curto é crítico para a capacidade de manobra da aeronave e, por isso, possui requisitos bem delimitados. O critério considerado na biblioteca é o método gráfico, popularmente conhecido como "critério da impressão digital". A estratégia consiste em mapear a frequência natural não amortecida ω_{pc} e a razão de amortecimento ζ_{pc} em um mapa de classificações, o qual é apresentado na Figura B.1.

Figura B.1 – Critério da impressão digital para o modo de período curto.



Fonte: (COOK, 2007).

B.2 – QUALIDADES DE VOO DO MODO LONGITUDINAL DE PERÍODO LONGO

As restrições para o modo de período longo são mais relaxadas, dependendo exclusivamente da razão de amortecimento ζ_{ph} , como apresentado na Tabela B.1.

Tabela B.1 – Qualidades de voo do modo de período longo.

Nível	ζ_{ph}
Nível 1	0.04
Nível 2	0.0
Nível 3	Caso instável, $T_{ph} > 55s$

Fonte: Adaptado de (COOK, 2007).

B.3 – QUALIDADES DE VOO DO MODO DE ROLAMENTO PURO

Como o rolamento puro descreve um modo de período curto para a dinâmica látero-direcional, é importante que possua respostas rápidas aos comando do piloto. A tabela B.2s especifica as constantes de tempo T_r relacionadas às qualidades de voo para cada tipo de aeronave e fase de voo.

Tabela B.2 – Qualidades de voo do modo de rolamento puro.

Classe da aeronave	Fase de voo	Valores máximos de T_r		
		Nível 1	Nível 2	Nível 3
I, IV	A, C	1.0	1.4	-
II, III	A, C	1.4	3.0	-
I, II, III, IV	B	1.4	3.0	-

Fonte: Adaptado de (COOK, 2007).

B.4 – QUALIDADES DE VOO DO MODO ESPIRAL

O modo espiral pode possuir valores flexíveis, podendo ser instável até certo nível. Nesse sentido, caso o modo seja estável, automaticamente é classificado com qualidade de voo de Nível 1, e caso seja instável, deve-se observar a tabela B.3.

Tabela B.3 – Qualidades de voo do modo espiral.

Fase de voo	Valores máximos de T_s		
	Nível 1	Nível 2	Nível 3
A, C	17.3	11.5	7.2
B	28.9	11.5	7.2

Fonte: Adaptado de (COOK, 2007).

B.5 – QUALIDADES DE VOO DO MODO *DUTCH-ROLL*

O modo *Dutch-roll* é um modo de período curto, o o que torna crítico na execução de manobras látero-direcionais. Por essa razão, as restrições são especificados em maior detalhe, como apresentado na Tabela B.4.

Tabela B.4 – Qualidades de voo do modo *Dutch-roll*.

Classe da aeronave	Fase de voo	Valores mínimos							
		Nível 1			Nível 2			Nível 3	
		ζ_{dr}	$\zeta_{dr}\omega_{dr}$	ω_{dr}	ζ_{dr}	$\zeta_{dr}\omega_{dr}$	ω_{dr}	ζ_{dr}	ω_{dr}
I, IV	A	0.19	0.35	1.0	0.02	0.05	0.5	0	0.4
II, III	A	0.19	0.35	0.5	0.02	0.05	0.5	0	0.4
I, II, III, IV	B	0.08	0.15	0.5	0.02	0.05	0.5	0	0.4
I, IV	C	0.08	0.15	1.0	0.02	0.05	0.5	0	0.4
II, III	C	0.08	0.1	0.5	0.02	0.05	0.5	0	0.4

Fonte: Adaptado de (COOK, 2007).

APÊNDICE C – DADOS DA AERONAVE DE ASA FIXA DE GRANDE PORTE

Os dados apresentados são adaptações dos resultados obtidos por Silva, Paglione e Yoneyama (2010). A condição de referência é um voo reto nivelado, a uma altitude de 10000m e velocidade de 224.6 m/s. As características geométricas, inércia e propulsivas apresentadas na Tabela C.1 são inspiradas nas aeronaves EMBRAER ERJ-190 e Boeing 737-200.

Tabela C.1 – Parâmetros globais da aeronave.

Parâmetros globais da aeronave	Medida	Unidade
Comprimento (l)	33	m
Comprimento da asa (b)	28,42	m
Corda média aerodinâmica da asa (\bar{c})	3,67	m
Área em planta da asa (S)	95	m ²
Massa (m)	45000	kg
Momento de inércia em x (I_{xx})	554000	kg m ²
Momento de inércia em y (I_{yy})	2530000	kg m ²
Momento de inércia em z (I_{zz})	3010000	kg m ²
Produto de inércia em z (I_{xzy})	106000	kg m ²
Produto de inércia em y (I_{yz}) e z (I_{xy})	0	kg m ²
Posição do motor esquerdo no sistema \mathcal{B}	$x: 0, y: -5.0, z: 1.42$	m
Posição do motor direito no sistema \mathcal{B}	$x: 0, y: 5.0, z: 1.42$	m
Apontamento de ambos os motores no sistema \mathcal{B}	$\psi_p: 0, \theta_p: 3$	°
Tração máxima de cada motor (T_{max})	35000	N
Coeficiente de densidade de ambos os motores (n_ρ)	0.775	adm
Coeficiente de velocidade de ambos os motores (n_V)	0	adm

Fonte: Adaptado de (SILVA; PAGLIONE; YONEYAMA, 2010)

Os coeficientes aerodinâmicos foram obtidos com auxílio dos *softwares Advanced Aircraft Analysis* (AAA) e Tornado. As derivadas de estabilidade e derivadas de controle são apresentadas na Tabela C.2.

Tabela C.2 – Derivadas de estabilidade e controle da aeronave de asa fixa de grande porte.

Coef.	0	α	β	p	q	r	δ_e	δ_a	δ_r
C_L	0.382	6.29	0.0	0.0	14.60	0.0	0.3891	0.0	0.0
C_D	0.0252	0.201	0.0	0.0	0.281	0.0	0.0126	0.0	0.0
C_Y	0.0	0.0	-0.785	0.0794	0.0	-0.572	0.0	-0.0094	0.303
C_l	0.0	0.0	-0.121	-0.522	0.0	0.254	0.0	-0.1784	-0.0464
C_m	0.0622	-3.63	0.0	0.0	-45.5	0.0	-1.598	0.0	0.0
C_n	0.0	0.0	0.174	-0.0587	0.0	-0.277	0.0	0.008	0.1594

Fonte: Adaptado de (SILVA; PAGLIONE; YONEYAMA, 2010)

APÊNDICE D – DADOS DA AERONAVE NÃO TRIPULADA COM CAPACIDADE DE POUSO E DECOLAGEM VERTICAL

A geometria da aeronave considerada é proposta por Bliamis et al. (2021), enquanto as de inércia e características propulsivas são inspiradas na aeronave conceitual apresentada por Chung, Ma e Shiau (2019). Esses valores são sintetizados na Tabela D.1.

Tabela D.1 – Parâmetros globais da aeronave.

Parâmetros globais da aeronave	Medida	Unidade
Comprimento da asa (b)	1.2	m
Corde média aerodinâmica da asa (\bar{c})	0.55	m
Área em planta da asa (S)	0.66	m ²
Massa (m)	47.5	kg
Momento de inércia em x (I_{xx})	590.4	kg m ²
Momento de inércia em y (I_{yy})	2698.6	kg m ²
Momento de inércia em z (I_{zz})	3210.4	kg m ²
Produto de inércia em z (I_{xzy})	113.1	kg m ²
Produto de inércia em y (I_{yz}) e z (I_{xy})	0	kg m ²
Posição dos motores coaxiais no sistema \mathcal{B}	$x: -0.25, y: 0, z: 0$	m
Posição do motor direito no sistema \mathcal{B}	$x: 0.6, y: 0.32, z: 0$	m
Posição do motor esquerdo no sistema \mathcal{B}	$x: 0.6, y: -0.32, z: 0$	m
Apontamento dos motores coaxiais no sistema \mathcal{B}	$\psi_p: 0, \theta_p: 90$	°
Coeficiente de tração dos motores coaxiais (C_{tb})	0.43×10^{-5}	N/(rad/s) ²
Coeficiente de torque dos motores frontais (C_{db})	7.2×10^{-7}	N·m/(rad/s) ²
Coeficiente de tração dos motores frontais (C_{te} e C_{td})	1.43×10^{-4}	N/(rad/s) ²
Coeficiente de torque dos motores frontais (C_{de} e C_{dd})	2.40×10^{-6}	N·m/(rad/s) ²

Fonte: Adaptado de (BLIAMIS et al., 2021) e (CHUNG; MA; SHIAU, 2019)

Os coeficientes aerodinâmicos foram obtidos com auxílio de simulações CFD considerando velocidade do ar de 18.66 m/s. As derivadas de estabilidade e derivadas de controle são apresentadas na Tabela D.2.

Tabela D.2 – Derivadas de estabilidade e controle da aeronave não tripulada conceitual.

Coef.	0	α	β	δ_{ed}	δ_{ee}
C_L	0.0	4.49	0.0	0.0	0.0
C_D	0.02	0.05	0.0	0.0	0.0
C_Y	0.0	0.0	-0.169	0.0	0.0
C_l	0.0	0.0	-0.088	-0.12	0.12
C_m	0.0	-0.57	0.0	-0.57	0.57
C_n	0.0	0.0	0.002	0.0	0.0

Fonte: Adaptado de (BLIAMIS et al., 2021)

APÊNDICE E – LINK PAR CÓDIGOS E EXEMPLOS

A versão mais recente da biblioteca, documentação, exemplos, bem como a completude dos códigos apresentados nesse trabalho pode ser acessada através do link: <https://github.com/KenedyMatiasso/PyAAT>.