

BRUNO LUIZ PEREIRA

**APLICAÇÃO DE SISTEMAS NEURO-FUZZY NO
CONTROLE DE AERONAVES EM OPERAÇÕES
CRÍTICAS DE VOO**



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA**

2021

BRUNO LUIZ PEREIRA

**APLICAÇÃO DE SISTEMAS NEURO-FUZZY NO CONTROLE DE
AERONAVES EM OPERAÇÕES CRÍTICAS DE VOO**

Tese de Doutorado apresentada ao programa de pós-graduação da Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de **DOUTOR EM ENGENHARIA MECÂNICA**.

Área de Concentração: Mecânica dos Sólidos e Vibrações.

Orientador: Prof. Dr. Roberto Mendes Finzi Neto.

Coorientadora: Profa. Dra. Rosana Sueli da Motta Jafelice.

UBERLÂNDIA - MG

2021

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

P436 Pereira, Bruno Luiz, 1993-
2021 Aplicação de Sistemas Neuro-Fuzzy no Controle de
Aeronaves em Operações Críticas de Voo
[recurso eletrônico] / Bruno Luiz Pereira. - 2021.

Orientador: Roberto Mendes Finzi Neto.

Coorientadora: Rosana Sueli da Motta Jafelice.

Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-
graduação em Engenharia Mecânica.

Modo de acesso: Internet.

Disponível em: <http://doi.org/10.14393/ufu.te.2021.555>

Inclui bibliografia.

Inclui ilustrações.

1. Engenharia mecânica. I. Finzi Neto, Roberto Mendes, 1974-, (Orient.). II. Jafelice, Rosana Sueli da Motta, 1964-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Engenharia Mecânica. IV. Título.

CDU: 621

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091


UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Engenharia Mecânica

Av. João Naves de Ávila, nº 2121, Bloco 1M, Sala 212 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

Telefone: (34) 3239-4282 - www.posgrad.mecanica.ufu.br - secposmec@mecanica.ufu.br


ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Mecânica				
Defesa de:	Tese de Doutorado Acadêmico, nº 318, COPEM				
Data:	09/12/2021	Hora de início:	14:00	Hora de encerramento:	18:38
Matrícula do Discente:	11723EMC010				
Nome do Discente:	Bruno Luiz Pereira				
Título do Trabalho:	Aplicação de Sistemas Neuro-Fuzzy no Controle de Aeronaves em Operações Críticas de Voo				
Área de concentração:	Mecânica dos Sólidos e Vibrações				
Linha de pesquisa:	Dinâmica de Sistemas Mecânicos				
Projeto de Pesquisa de vinculação:					

Reuniu-se por meio de videoconferência a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Mecânica, assim composta: Professores Doutores: Rosana Sueli da Motta Jafelice (coorientadora) - FAMAT/UFU; Fran Sérgio Lobato - FEQUI/UFU; Rogério Sales Gonçalves - FEMEC/UFU; Neusa Maria Franco de Oliveira - ITA; Marcos Eduardo Ribeiro do Valle Mesquita - UNICAMP; e Roberto Mendes Finzi Neto - FEMEC/UFU, orientador do candidato. Ressalta-se que os Prof. Roberto Mendes Finzi Neto, Fran Sérgio Lobato, Rogério Sales Gonçalves e o Discente participaram da defesa por meio de videoconferência desde a cidade de Uberlândia/MG; Neusa Maria Franco de Oliveira desde a cidade de São José dos Campos/SP; e Marcos Eduardo Ribeiro do Valle Mesquita desde a cidade de Campinas/SP, em atendimento a Portaria nº 36, de 19 de março de 2020, da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES.

Iniciando os trabalhos, o presidente da mesa, Dr. Roberto Mendes Finzi Neto, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor(a) presidente concedeu a palavra, pela ordem sucessivamente, aos(às) examinadores(as), que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o(a) candidato(a):

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.



Documento assinado eletronicamente por **Roberto Mendes Finzi Neto, Professor(a) do Magistério Superior**, em 09/12/2021, às 18:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rosana Sueli da Motta Jafelice, Professor(a) do Magistério Superior**, em 09/12/2021, às 18:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rogério Sales Gonçalves, Professor(a) do Magistério Superior**, em 09/12/2021, às 18:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marcos Eduardo Ribeiro do Valle Mesquita, Usuário Externo**, em 09/12/2021, às 18:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Fran Sergio Lobato, Professor(a) do Magistério Superior**, em 09/12/2021, às 18:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Neusa Maria Franco de Oliveira, Usuário Externo**, em 09/12/2021, às 18:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3232322** e o código CRC **3AFDD678**.

Dedico este trabalho à minha família, sobretudo aos meus pais, por sempre estarem ao meu lado em todos os momentos.

Agradecimentos

Aos meus pais, Cristiane e Wagner, pelo carinho, amor, compreensão, incentivo, por acreditarem na minha capacidade e serem exemplos de seres humanos, e por estarem presentes nos vários momentos de alegria vividos, e também nos momentos de dificuldade.

Aos demais familiares, sobretudo à minha avó Zélia, pelo amor doado a mim, e por manifestarem, de forma sincera, satisfação e alegria em verem e participarem de minhas conquistas.

À minha namorada Camila, pela paciência, pelo cuidado, pelo carinho, pelo companheirismo, e principalmente pelo amor demonstrado a mim todos os dias.

Aos amigos da UFU, do ITV, da dança, e aos demais amigos, que entram em minha vida de diferentes formas com o simples e despretensioso propósito de compartilharem bons momentos, e sobretudo de me apoiarem nos momentos difíceis.

Aos companheiros do Laboratório de Aeronaves Autônomas e do Laboratório de Planejamento Automático da Manufatura, pelo suporte técnico e principalmente pela amizade.

Aos professores Finzi, Leonardo Sanches e Jean Tavares, e também a todos os bons professores que tive o prazer de conhecer, por terem me proporcionado oportunidades de crescimento pessoal e profissional.

À professora Rosana, por ser exemplo em minha vida de seriedade e compromisso com a nossa profissão de ensinar. Agradeço-lhe também por ter auxiliado a despertar em mim o amor pela pesquisa e o desejo de contribuir com o progresso científico e principalmente com o desenvolvimento social do país através da educação.

À Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de cursar uma pós-graduação de excelência. Ao CNPq, CAPES, Fapemig e FAPDF pelo suporte financeiro, e agradeço também ao Instituto Federal de Brasília pelas horas de trabalho concedidas à minha formação continuada.

PEREIRA, B.L. **APLICAÇÃO DE SISTEMAS NEURO-FUZZY NO CONTROLE DE AERONAVES EM OPERAÇÕES CRÍTICAS DE VOO**. 2021. Tese de Doutorado – Doutorado Acadêmico em Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, 2020.

RESUMO

A perda de controle da aeronave durante o voo é causa de aproximadamente 70% de todas as fatalidades ocorridas em aeronaves com massa de decolagem superior a 5.700 kg. A relevância do tema provoca a comunidade técnica e científica, e leva a uma série de discussões e à geração de normas, procedimentos e dispositivos que buscam mitigar as causas desses acidentes. No intuito de contribuir com o setor aeronáutico no que tange ao desenvolvimento de novas estratégias que procurem minimizar o número de acidentes aéreos fatais, neste trabalho é proposta a utilização de uma nova arquitetura de controle baseada na combinação de sistemas neuro-fuzzy no controle de aeronaves em operações críticas de voo. Para isso, desenvolve-se um novo método de inferência fuzzy, denominado PIA (*Pondered Individual Analysis*), que alia intuitividade e alto desempenho computacional no processo de tradução matemática da base de regras envolvida no processo. A validação da técnica proposta passa pelo desenvolvimento de uma simulação do tipo *software-in-the-loop* entre o MATLAB e o X-Plane 11, em que se verifica a capacidade da arquitetura de controle proposta, em operações críticas de voo, de manter a resposta da aeronave em torno dos sinais de referência, e também pela verificação do desempenho do sistema de controle ao se levar em consideração um modelo dinâmico levantado a partir de dados experimentais, extraídos em ensaio de voo realizado em uma aeronave Cessna 172 em escala reduzida. Os resultados das dinâmicas longitudinal e látero-direcional da aeronave são analisados e comparados aos obtidos com os controladores proporcional integral derivativo, e neuro-fuzzy que utiliza o método de inferência fuzzy de Takagi-Sugeno, e apresentam menor erro médio absoluto com relação ao comportamento desejado para a aeronave, e dessa forma evidenciam que o método PIA demonstra ser uma eficaz ferramenta a ser considerada na resolução de problemas na área de controle.

Palavras-Chave: Análise Individual Ponderada, Controle de Aeronaves, Controle Inteligente, Controle Neuro-Fuzzy, LOC-I, Método de Inferência Fuzzy, PIA, SBRF, Sistema Neuro-Fuzzy, Teoria dos Conjuntos Fuzzy.

PEREIRA, B.L. **APPLICATION OF NEURO-FUZZY SYSTEMS IN AIRCRAFT CONTROL IN CRITICAL FLIGHT OPERATIONS**. 2021. Doctoral Thesis– PhD in Mechanical Engineering, Universidade Federal de Uberlândia, Uberlândia, 2021.

ABSTRACT

Loss of control in-flight is the cause of approximately 70% of all fatalities occurring in aircraft with a take-off mass greater than 5,700 kg. The relevance of this subject provokes the technical and scientific community, and leads to a series of discussions and the generation of norms, procedures and devices that seek to mitigate the causes of these accidents. In order to contribute to the aeronautical sector with regard to the development of new strategies that seek to minimize the number of fatal air accidents, this work proposes the use of a new control architecture based on the combination of neuro-fuzzy systems in the control of aircraft in critical flight operations. For that matter, a new fuzzy inference method, called PIA (Pondered Individual Analysis), is developed, which combines intuitiveness and high computational performance in the process of mathematical translation of the rule base involved in the process. The validation of the proposed technique involves the development of a software-in-the-loop simulation between MATLAB and X-Plane 11, in which the ability of the proposed control architecture, in critical flight operations, to maintain the response of the aircraft around the reference signals is verified, and also it is analyzed the performance of the control system when taking into account a dynamic model raised from experimental data, extracted from a flight test carried out on a small scale Cessna 172 aircraft. The results of the longitudinal and lateral-directional dynamics of the aircraft are analyzed and compared to those obtained with the proportional integral derivative controller and with the neuro-fuzzy controller that uses the Takagi-Sugeno fuzzy inference method, and they present lower mean absolute error in relation to the desired behavior for the aircraft, and thus highlight that the PIA method demonstrates to be an effective tool to be considered in solving problems in the control area.

Keywords: Pondered Individual Analysis, Aircraft Control, Intelligent Control, Neuro-Fuzzy Control, LOC-I, Fuzzy Inference Method, PIA, FRBS, Neuro-Fuzzy System, Fuzzy Sets Theory.

Lista de Figuras

Figura 1.1: Imagens extraídas do simulador de voo X-Plane	2
Figura 1.2: Fluxograma – organização da tese	7
Figura 2.1: Ilustração da arquitetura de um SBRF	12
Figura 2.2: Ilustração do método de inferência de Mamdani	13
Figura 2.3: Ilustração do método de inferência de Takagi-Sugeno	14
Figura 2.4: Ilustração do método Kosko-Mizumoto	16
Figura 2.5: Ilustração do método de inferência de Larsen	16
Figura 2.6: Arquitetura da abordagem HGFRD	18
Figura 2.7: Representação da arquitetura de uma RNA multicamadas	19
Figura 2.8: Esquema de um neurônio i da camada I	20
Figura 2.9: Funções de ativação usualmente utilizadas	20
Figura 2.10: Relação temporal entre os instantes de execução do controle e o número de iterações realizadas para a atualização da rede	22
Figura 2.11: Princípio do controle indireto adaptativo	22
Figura 2.12: Esquema básico da estimação on-line dos parâmetros da rede neural	23
Figura 2.13: Esquema do controle direto adaptativo	24
Figura 2.14: Representação de um sistema neuro-fuzzy	25
Figura 2.15: Representação da evolução diferencial, adaptada de Mór <i>et al.</i> (2015), sendo S o conjunto das possíveis soluções do problema de otimização.	27
Figura 2.16: Visualização dos eixos, ângulos de Euler, velocidades, forças e momentos da aeronave	29
Figura 2.17: Ilustração das principais superfícies de comando	33
Figura 3.1: Percentual de acidentes aéreos em função das categorias de acidentes apresentadas com a nomenclatura padrão	38
Figura 3.2: Fase do voo em que houve a ocorrência de LOC-I	39
Figura 3.3: Bancada de testes AirSTAR	41
Figura 3.4: Túnel de vento e protótipo utilizados em Frink <i>et al.</i> (2017)	42
Figura 3.5: O X-Plane permite simular a operação de aeronaves comerciais e de aeromodelos	45
Figura 3.6: Interface de comunicação entre o Simulink e o X-Plane	45
Figura 3.7: Navegação da aeronave	46
Figura 3.8: Seletor de algumas das configurações de voo do <i>software</i> X-Plane	46
Figura 3.9: Modelagem geométrica e dinâmica realizada no X-Plane 9	47
Figura 3.10: Desempenho do controlador PID em cascata no SITL, sendo o valor de referência em amarelo, e a saída do sistema em preto	47
Figura 3.11: Flight Gear simulando a operação da aeronave aerosonda	50
Figura 3.12: Arquitetura de controle desenvolvida pela empresa Micropilot	51
Figura 3.13: Bancada utilizada para o estudo de sistemas aeronáuticos de asa rotativa	52
Figura 3.14: Performance dos controles	53
Figura 3.15: Funções de pertinência dos SBRFs do tipo 1	53
Figura 3.16: Funções de pertinência dos SBRFs do tipo 2	54

Figura 3.17: Desempenho dos SBRFs dos tipos 1, em azul, e 2, em vermelho, em relação ao sinal de referência, em preto, no controle dos ângulos de arfagem, à esquerda, e rolagem, à direita	54
Figura 3.18: Desempenho dos controladores fuzzy e PD, gráfico azul, em relação ao sinal de referência, em preto, no controle do ângulo de arfagem	54
Figura 3.19: Arquitetura de controle PID fuzzy adotada em Swarup (2020).	55
Figura 3.20: Resposta do sistema de arfagem da aeronave.....	55
Figura 3.21: Superfícies de resposta do SBRF do controlador PID fuzzy	56
Figura 3.22: Desempenho do controlador PID fuzzy	57
Figura 3.23: Arquitetura de controle fuzzy.....	57
Figura 3.24: À esquerda, desempenho do controlador fuzzy não otimizado, em preto, e otimizado, em vermelho; à direita, resultado do hardware-in-the-loop, em preto, e do <i>software-in-the-loop</i> , em vermelho	58
Figura 3.25: Arquitetura de controle ANFIS+PID	59
Figura 3.26: Desempenho das arquiteturas de controle com relação aos ângulos de rolagem, à esquerda, e guinada, à direita.....	60
Figura 4.1: Esquema das etapas que compõem a metodologia da tese	62
Figura 4.2: Esquema do estudo de caso desta tese.....	64
Figura 4.3: Esquema do método de controle	65
Figura 4.4: Esquema do método PIA de inferência fuzzy	66
Figura 4.5: Esquema da sintonização do sistema de controle.....	68
Figura 4.6: Esquema da análise de estabilidade do sistema de controle	69
Figura 4.7: Esquema da comunicação entre MATLAB e X-Plane.....	70
Figura 4.8: Esquema da análise de desempenho do controlador em operações críticas	71
Figura 4.9: Esquema relativo à construção de aeronave em escala reduzida	72
Figura 4.10: Esquema relativo à obtenção de dados de voo	72
Figura 4.11: Esquema que resume a identificação do modelo da aeronave	73
Figura 4.12: Esquema que ilustra a validação da estratégia de controle	73
Figura 5.1: Definição dos pontos centrais das funções de pertinência.....	75
Figura 5.2: Valores dos candidatos	78
Figura 5.3: Ilustração do método PIA, em que αx e αy são os pesos associados, respectivamente, à influência de x e y na variável de saída.....	80
Figura 5.4: Fluxogramas do método PIA	81
Figura 5.5: Variáveis envolvidas no processo de modelagem	82
Figura 5.6: Funções de pertinência das entradas	84
Figura 5.7: Funções de pertinência das saídas	84
Figura 5.8: Superfície de resposta $CD(\alpha, \delta e)$ para os métodos de Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen	86
Figura 5.9: Superfície de resposta $CD(\alpha, \delta e)$ para o método PIA.....	86
Figura 5.10: Superfície de resposta $CL(\alpha, \delta e)$ para os métodos de Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen	87
Figura 5.11: Superfície de resposta $CL(\alpha, \delta e)$ para o método PIA.....	87
Figura 5.12: Processo de otimização do erro absoluto médio (MAE) da resposta do SBRF que utiliza o método PIA em função da iteração da evolução diferencial	89
Figura 5.13: CD e CL em função do tempo com o método PIA.....	89
Figura 6.1: Controle longitudinal	93
Figura 6.2: Controle látero-direcional	93

Figura 6.3: Funções de pertinência das entradas do <i>SBRF2</i>	102
Figura 6.4: Funções de pertinência da saída do <i>SBRF2</i>	102
Figura 6.5: Funções de pertinência da entrada do <i>SBRF1</i>	103
Figura 6.6: Funções de pertinência da saída do <i>SBRF1</i>	103
Figura 6.7: Funções de pertinência das entradas do <i>SBRF3</i>	104
Figura 6.8: Funções de pertinência da saída do <i>SBRF3</i>	104
Figura 6.9: Funções de pertinência das entradas do <i>SBRF5</i>	105
Figura 6.10: Funções de pertinência da saída do <i>SBRF5</i>	105
Figura 6.11: Funções de pertinência da entrada do <i>SBRF4</i>	106
Figura 6.12: Funções de pertinência da saída do <i>SBRF4</i>	107
Figura 6.13: Funções de pertinência das entradas do <i>SBRF6</i>	107
Figura 6.14: Funções de pertinência da saída do <i>SBRF6</i>	108
Figura 6.15: Velocidade da aeronave $U(m/s)$ em função do tempo de simulação $t(s)$	108
Figura 6.16: Ângulo de arfagem da aeronave θ em função do tempo de simulação $t(s)$...	109
Figura 6.17: Altitude da aeronave em relação ao ponto de linearização $h(m)$ em função do tempo de simulação $t(s)$	109
Figura 6.18: Ângulo de rolagem da aeronave ϕ em função do tempo de simulação $t(s)$...	110
Figura 6.19: Ângulo de guinada da aeronave ψ em função do tempo de simulação $t(s)$...	110
Figura 6.20: Ações de controle associadas à dinâmica longitudinal da aeronave	111
Figura 6.21: Ações de controle associadas à dinâmica látero-direcional da aeronave	111
Figura 6.22: Avaliação numérica da derivada da função de Lyapunov V	115
Figura 6.23: Variáveis enviadas pelo X-Plane	118
Figura 6.24: Caixas de seleção da tabela presente na aba <i>Data Output</i>	120
Figura 6.25: Dados de voo no X-Plane 11	121
Figura 6.26: Configurações da comunicação	121
Figura 6.27: Fluxograma – <i>software-in-the-loop</i> entre MATLAB e X-Plane	123
Figura 6.28: Inicialização da comunicação	124
Figura 6.29: Recebimento das mensagens advindas do simulador	124
Figura 6.30: Função que obtém os dados, referentes ao comportamento da aeronave, contidos nos pacotes enviados pelo simulador	125
Figura 6.31: Função que converte um vetor de 4 bytes em um número do tipo <i>float – single precision</i>	125
Figura 6.32: Função responsável pela conversão de um vetor de 4 bytes em uma cadeia binária formada pela concatenação dos números binários gerados a partir de cada um dos bytes	125
Figura 6.33: Função que converte um número binário de 32 bits em um número real do tipo <i>float – single precision</i>	126
Figura 6.34: Função responsável por enviar ao X-Plane um pacote contendo um comando que altera a deflexão do profundor, em inglês, <i>elevator</i>	127
Figura 6.35: Função responsável pela construção dos pacotes de comando enviados ao X-Plane durante a simulação	127
Figura 6.36: Seção do código que encerra a comunicação entre o MATLAB e o X-Plane ..	128
Figura 6.37: Função que envia ao X-Plane 11 um pacote de comandos que informa ao simulador que nenhuma outra mensagem será remetida pelo MATLAB, e que a aeronave, a partir de então, passará a responder apenas a intervenções diretas do usuário	128
Figura 6.38: Ilustração da primeira simulação realizada no simulador de voo X-Plane 11 ..	130

Figura 6.39: Ângulo de arfagem da aeronave $\theta(o)$ em função do tempo $t(s)$ na primeira simulação.....	130
Figura 6.40: Ângulo de rolagem da aeronave $\phi(o)$ em função do tempo $t(s)$ na primeira simulação.....	130
Figura 6.41: Ângulo de arfagem da aeronave $\theta(o)$ em função do tempo $t(s)$ na segunda simulação.....	131
Figura 6.42: Ângulo de rolagem da aeronave $\phi(o)$ em função do tempo $t(s)$ na segunda simulação.....	132
Figura 6.43: Ângulo de arfagem da aeronave $\theta(o)$ em função do tempo $t(s)$ na terceira simulação.....	133
Figura 6.44: Ângulo de rolagem da aeronave $\phi(o)$ em função do tempo $t(s)$ na terceira simulação.....	133
Figura 6.45: Ilustração da quarta simulação realizada no simulador de voo X-Plane 11	134
Figura 6.46: Ângulo de arfagem da aeronave $\theta(o)$ em função do tempo $t(s)$ na quarta simulação.....	134
Figura 6.47: Ângulo de rolagem da aeronave $\phi(o)$ em função do tempo $t(s)$ na quarta simulação.....	135
Figura 6.48: Ângulo de arfagem da aeronave $\theta(o)$ em função do tempo $t(s)$ na terceira simulação.....	136
Figura 6.49: Ângulo de rolagem da aeronave $\phi(o)$ em função do tempo $t(s)$ na terceira simulação.....	136
Figura 6.50: Fuselagem, trem de pouco e hélice do aeromodelo Cessna 172 Class 500 ...	137
Figura 6.51: Controlador de voo Pixhawk e sua localização na aeronave.....	138
Figura 6.52: <i>Software</i> Arduplane e aeronave Cessna 172 em escala reduzida	139
Figura 6.53: Dados extraídos em voo	140
Figura 6.54: Amostra de identificação do modelo dinâmico	140
Figura 6.55: Amostra de validação do modelo dinâmico	141
Figura 6.56: Ajuste do modelo dinâmico às amostras selecionadas	141
Figura 6.57: Resultados obtidos a partir do controlador neuro-fuzzy que lança mão do método PIA de inferência fuzzy.....	143
Figura 6.58: Análise comparativa das técnicas de controle	143

Lista de Tabelas

Tabela 1.1: Fatores Causadores da Perda de Controle da Aeronave Durante o Voo	5
Tabela 3.1: Trabalhos que Exploram a Perda de Controle da Aeronave Durante o Voo	36
Tabela 3.2: Categorias de Acidentes e as Descrições de Suas Terminologias Padrão	37
Tabela 3.3: Principais Categorias de Acidentes Aéreos Quanto à Letalidade	38
Tabela 3.4: Fases de Voo	39
Tabela 3.5: Fatores Recorrentes em Acidentes Devido à LOC-I.....	40
Tabela 3.6: Trabalhos que Exploram a Utilização de Simuladores de Voo	44
Tabela 3.7: Inteligência Computacional Aplicada ao Controle de Sistemas	48
Tabela 3.8: Inteligência Computacional Aplicada à Modelagem e Controle de Aeronaves ...	49
Tabela 3.9: Características dos Controladores no Domínio do Tempo	56
Tabela 3.10: Características dos Controladores no Domínio do Tempo.....	59
Tabela 5.1: Exemplo 1 - Base de Regras.....	76
Tabela 5.2: Exemplo 1 - Relação Entre as Variáveis.	77
Tabela 5.3: Exemplo 2 - Base de Regras.....	79
Tabela 5.4: Exemplo 2 - Relação Entre as Variáveis.	79
Tabela 5.5: Base de Regras	85
Tabela 5.6: Parâmetros da Evolução Diferencial	88
Tabela 5.7: Estatísticas dos Métodos de Inferência Fuzzy para <i>CD</i>	90
Tabela 5.8: Estatísticas dos Métodos de Inferência Fuzzy para <i>CL</i>	90
Tabela 6.1: Entradas e Saídas dos Sistemas Baseados em Regras Fuzzy	94
Tabela 6.2: <i>SBRF1</i> - Base de Regras.....	94
Tabela 6.3: <i>SBRF2</i> - Base de Regras.....	95
Tabela 6.4: <i>SBRF3</i> - Base de Regras.....	96
Tabela 6.5: <i>SBRF4</i> - Base de Regras.....	96
Tabela 6.6: <i>SBRF5</i> - Base de Regras.....	97
Tabela 6.7: <i>SBRF6</i> - Base de Regras.....	97
Tabela 6.8: Valores dos Parâmetros para a Aeronave Cessna 172	99
Tabela 6.9: Parâmetros da Evolução Diferencial	101
Tabela 6.10: Erro MAE da Resposta do Sistema Controlado	112
Tabela 6.11: Intervalo de Avaliação da Função de Lyapunov <i>V</i>	114

Lista de Abreviaturas

AG – Algoritmo Genético

ANAC – Agência Nacional de Aviação Civil

ANFIS – *Adaptive-Network-based Fuzzy Inference System*

APR – *Approach*

ASCII – *American Standard Code for Information Interchange*

CFIT – *Controlled Flight into Terrain*

CRZ – *Cruise*

DST – *Descent*

ECL – *En Route Climb*

FAA – *Federal Aviation Administration*

FLC – Fuzzy Logic Controller

GOA – *Go-around*

GPS – *Global Positioning System*

HGFRD – *Hybrid Adaptive Granular Fuzzy Approach to Rule Discovery*

IATA – *International Air Transport Association*

IP – *Internet Protocol*

ICL – *Initial Climb*

LAA – Laboratório de Aeronaves Autônomas

LND – *Landing*

LOC-I – *Loss of Control In-Flight*

LQR – *Linear Quadratic Regulator*

MAE – *Mean Absolute Error*

MAPL – *Manufacturing Automated Planning Lab*

NASA – *National Aeronautics and Space Administration*

PD – Proporcional Derivativo

PI – Proporcional Integral

PIA – *Pondered Individual Analysis*

PID – Proporcional Integral Derivativo

RAM – *Random Access Memory*

RNA – Rede Neural Artificial.

RTGM – *Real Time Global Modeling*

SBRF – Sistema Baseado em Regras Fuzzy

SD – *Secure Digital*

SISO – *Single Input Single Output*

SITL – *Software-In-The-Loop*

SSD – *Solid-State Drive*

TOF – *Take-off*

TS – Takagi-Sugeno

UDP – *User Datagram Protocol*

VANT – Veículo Aéreo Não Tripulado

ZN – Ziegler-Nichols

Sumário

CAPÍTULO I: INTRODUÇÃO	1
1.1. Objetivos	4
1.2. Justificativa	5
CAPÍTULO II: FUNDAMENTAÇÃO TEÓRICA	8
2.1. Controle e Sistemas Não-Lineares	8
2.2. Teoria dos Conjuntos Fuzzy	10
2.3. Métodos de Inferência Fuzzy	15
2.4. Redes Neurais	19
2.4.1. Conceitos Gerais	19
2.4.2. Treinamento da Rede Neural	21
2.5. ANFIS	25
2.6. Evolução Diferencial	26
2.7. Modelagem Dinâmica de uma Aeronave de Asa Fixa	28
CAPÍTULO III: REVISÃO DE LITERATURA	35
3.1. Perda de Controle da Aeronave Durante o Voo	35
3.2. Simuladores de Voo	43
3.3. Inteligência Computacional Aplicada a Sistemas Aeronáuticos	48
CAPÍTULO IV: METODOLOGIA	61
CAPÍTULO V: PIA - UM NOVO MÉTODO DE INFERÊNCIA FUZZY	74
5.1. Método PIA: Descrição, Conceitos e Definições	75
5.2. Estudo de Caso: Modelagem Aerodinâmica	82
CAPÍTULO VI: CONTROLE NEURO-FUZZY PARA AERONAVES	92
6.1. Arquitetura de Controle Neuro-Fuzzy	92
6.2. Sintonização e Estabilidade do Sistema de Controle	98
6.3. <i>Software-in-the-Loop</i> – Operações Críticas de Voo	115
6.4. Controle de Aeronave em Escala Reduzida	137
CAPÍTULO VII: CONCLUSÕES	145
7.1. Principais Conclusões	145
7.2. Perspectivas e Trabalhos Futuros	147
7.3. Principais Publicações	148
REFERÊNCIAS BIBLIOGRÁFICAS	151

APÊNDICE	158
A) Filtro de Kalman.....	158
A.1 Expansão do Método para Problemas com Mais de Uma Variável de Estado e Equações do Filtro de Kalman.....	159
A.2 Determinação dos Parâmetros do Filtro de kalman	160
A.2.1 Método Analítico	161
A.2.2 Método Numérico	162

CAPÍTULO I

INTRODUÇÃO

As técnicas da inteligência computacional baseiam-se na observação de como os seres vivos, e em particular os seres humanos, comportam-se enfrentando e resolvendo diferentes problemas. Estas técnicas são geralmente versáteis e ajustadas para controlar diferentes sistemas dinâmicos, mesmo sob qualquer variação de suas propriedades. De fato, estas variações podem acontecer, como por exemplo, devido ao envelhecimento ou à falha de componentes mecânicos. Nesse sentido, os parâmetros de controle precisam ser acessados e adaptados em tempo real ou em operação off-line.

Com o aumento da capacidade de processamento de sistemas embarcados, ocorrido nos últimos anos, torna-se viável lançar mão de técnicas de otimização e da inteligência computacional na resolução de problemas de controle em tempo real. Os controladores clássicos tornam-se ineficazes à medida em que há o aumento da complexidade dos processos, como por exemplo em cadeias de controle multivariável com acoplamento entre as variáveis, presença de fortes não linearidades e variações rápidas dos parâmetros dos processos (SILVA, 2006).

Em um sistema aeronáutico, as variações das propriedades dinâmicas podem ocorrer devido a vários fatores: alteração das condições de voo, alterações climáticas, congelamento ou colapso de partes da estrutura, perda de potência do sistema propulsor, ocorrência do *stall* (perda abrupta de sustentação de uma superfície aerodinâmica devido ao aumento de seu ângulo de ataque), dentre outros fatores.

De acordo com o relatório desenvolvido pela IATA (*International Air Transport Association*), (IATA, 2020), entre 2015 e 2019, aproximadamente 70% de todas as fatalidades ocorridas em aeronaves com massa de decolagem superior a 5.700 kg foram causadas devido à perda de controle da aeronave durante o voo (em inglês, *Loss of Control In-Flight* – LOC-I). A relevância do tema provoca a comunidade técnica e científica, e leva a uma série de discussões e à geração de normas a respeito de novas abordagens necessárias à mitigação das causas desses acidentes (OLIVEIRA, 2018).

A perda de controle da aeronave durante o voo pode resultar de uma série de fatores, incluindo falhas no motor, condições meteorológicas severas, congelamento, falhas operacionais e o *stall* (IATA, 2015). A partir de 2006, através de uma estação móvel de testes de dinâmica de voo e controle de aeronaves em escala reduzida, a NASA (*National Aeronautics and Space Administration*) desenvolve pesquisas visando a melhor compreensão do LOC-I e o desenvolvimento de estratégias para a retomada de controle da aeronave submetida a condições críticas de voo (COX, CUNNINGHAM e JORDAN, 2012; FRINK, *et al.*, 2017).

As verificações que envolvem novas modelagens e o desenvolvimento de sistemas de controle mais eficientes para aeronaves comerciais, sobretudo devido aos altos custos associados ao projeto e ao desenvolvimento de protótipos, geralmente não são inicialmente realizadas em aeronaves reais ou nem mesmo em aeromodelos, uma vez que quaisquer problemas poderiam ocasionar perdas materiais consideráveis. Dessa forma, a simulação é um recurso correntemente utilizado na validação prévia de técnicas e de novas tecnologias do setor aeronáutico. A Fig. 1.1 ilustra uma simulação realizada no simulador de voo X-Plane.



Figura 1.1: Imagens extraídas do simulador de voo X-Plane (autoria própria).

Em Craighead *et al.* (2007) são apresentados os principais simuladores de voo utilizados para o desenvolvimento de produtos e para a simulação de pilotagem, destacando-se o simulador de voo X-Plane, certificado pela FAA (*Federal Aviation Administration*) para o treinamento de pilotos. Este simulador é utilizado, de forma bem-sucedida, no teste de controladores aplicados a aeronaves reais (BITTAR, OLIVEIRA e FIGUEREIDO, 2014). Há trabalhos (BITTAR e OLIVEIRA, 2013; BITTAR *et al.*, 2014; BITTAR, OLIVEIRA e FIGUEREIDO, 2014; ZABIDIN, PAIRAN e SHAMSUDIN, 2020) que propõem a utilização do Simulink e do X-Plane na realização de simulações do tipo *software-in-the-loop* (SITL), em que o X-Plane presta-se como o ambiente de modelagem e simulação de voo, e no Simulink é desenvolvida a interface de comunicação entre os programas e a implementação da arquitetura de controle da planta.

Há trabalhos na literatura que utilizam técnicas da inteligência computacional para o controle de aeronaves submetidas a falha de componentes e atuadores (KWONG *et al.*, 1995; CHEN, SHI e LIM, 2016), para o controle de pouso (LIU, NAADIMUTHU e LEE, 2008), para o controle de atitude (CHEN e CHENG, 1998; KURNAZ, CETIN e KAVNAK, 2010; THUMS, TORRES e PALHARES, 2012; JHA, GAUR e YADAV, 2016; XU, ZHANG e PAN, 2016; ULUS e ESKI, 2021) e para a modelagem aerodinâmica (BRANDON e MORELLI, 2012), porém em nenhuma pesquisa conhecida pelos autores ocorre a utilização de simuladores de voo certificados por agências reguladoras, ou o uso de procedimentos experimentais, na validação das técnicas de controle desenvolvidas baseadas na inteligência computacional.

Neste projeto de pesquisa, opta-se pela utilização do sistema neuro-fuzzy dado à sua característica de aliar a capacidade de adaptação aos problemas da rede neural (HAYKIN, 2008), com a robustez e intuitividade da teoria dos conjuntos fuzzy (ZADEH, 1965), assim como em outros trabalhos que utilizam técnica similar (JANG, 1993), conhecida como ANFIS (*Adaptive-Neuro-Based Fuzzy Inference System*), no setor aeronáutico (KWONG *et al.*, 1995; LIU, NAADIMUTHU e LEE, 2008; KURNAZ, CETIN e KAYNAK, 2010; BRANDON e MORELLI, 2012; ULUS e ESKI, 2021). Este trabalho, porém, inova ao propor e aplicar um novo método de inferência fuzzy denominado PIA (*Pondered Individual Analysis*), que garante intuitividade e dispensa o uso de métodos de integração numérica, ao controle de uma aeronave submetida a condições críticas de voo, simuladas por meio do simulador de voo X-Plane.

1.1. Objetivos

Os objetivos principais da pesquisa são a análise, projeto e desenvolvimento de um controlador inteligente adaptativo a ser aplicado no controle de aeronaves em operações críticas de voo, como condições climáticas severas, rajadas de vento e mau funcionamento ou falha do sistema propulsor, dando ênfase no desenvolvimento do algoritmo e na contribuição da pesquisa na área da inteligência computacional.

Como estudo de caso do sistema de controle, utiliza-se a aeronave Cessna 172, devido à grande documentação disponível na literatura, simplicidade e importância do modelo, além do simulador de voo X-Plane como ambiente de simulação desta aeronave.

Os objetivos específicos do projeto são:

- Concepção de um novo método de inferência fuzzy denominado PIA (*Pondered Individual Analysis*);
- Projeto e implementação de um sistema de controle neuro-fuzzy para uma aeronave Cessna 172, com foco no controle longitudinal e látero-direcional do avião em operações críticas de voo;
- Realização de simulações no software MATLAB para a validação numérica e análise de estabilidade da arquitetura de controle desenvolvida, a partir de uma modelagem dinâmica linear do comportamento da aeronave em voo;
- Validação numérica da arquitetura de controle a partir de uma simulação do tipo *software-in-the loop* através do MATLAB e do simulador de voo X-Plane;
- Concepção de um protótipo em escala reduzida da aeronave Cessna 172;
- Realização de ensaios com o protótipo no intuito de coletar dados e analisar comparativamente as técnicas clássicas de controle com as desenvolvidas neste projeto.

Os principais fatores de inovação são citados a seguir:

- Desenvolvimento do novo método de inferência fuzzy PIA;
- Utilização de uma arquitetura de controle neuro-fuzzy associada ao método PIA de inferência fuzzy;
- Aplicação do sistema neuro-fuzzy em operações críticas de voo no simulador X-Plane;
- Validação numérica de um controlador neuro-fuzzy;
- Análise comparativa entre as técnicas de controle clássicas e as baseadas na inteligência computacional a partir de dados numéricos e experimentais.

1.2. Justificativa

Conforme discutido anteriormente, a perda de controle da aeronave durante o voo (LOC-I) é responsável por aproximadamente 70% de todas as fatalidades ocorridas em aeronaves com massa de decolagem superior a 5.700 kg (780 das 1116 mortes).

Além das falhas relacionadas à tripulação e à operação da aeronave em condições de voo crítico (com ocorrência em cerca de 55% dos acidentes causados devido ao LOC-I), na Tab. 1.1 são apresentadas as principais condições de voo presentes nos acidentes relacionados à perda de controle da aeronave, sendo que em muitos dos casos há a ocorrência simultânea de mais de um fator de risco (IATA, 2020).

Tabela 1.1: Fatores Causadores da Perda de Controle da Aeronave Durante o Voo (IATA, 2020).

Fatores Causadores de LOC-I	Ocorrência (%)
Operações de Voo (falha humana)	55
Condições meteorológicas críticas	45
Voo manual / Falta de aderência aos Procedimentos Operacionais Padrão (POPs)	50
Operação fora das limitações da aeronave	40
Performance da tripulação	50
Velocidade vertical/lateral fora do limite operacional da aeronave	35
Mau funcionamento da aeronave	35
Mau funcionamento / falha do sistema propulsor	20

A partir dos dados apresentados na Tab. 1.1, verifica-se que o estudo de técnicas aplicadas no controle de aeronaves submetidas a condições climáticas severas, rajadas de vento, e mau funcionamento ou falha do sistema propulsor pode assumir grande importância não somente no cenário acadêmico (COX, CUNNINGHAM e JORDAN, 2012), mas também na contribuição técnica relativa à segurança de aeronaves.

A aplicação de técnicas da inteligência computacional na resolução de problemas relacionados ao controle e modelagem de aeronaves é uma temática já desenvolvida em trabalhos prévios, conforme será apresentado na revisão de literatura, mas em nenhum trabalho conhecido pelos autores há o uso de simuladores de voo catalogados por agências reguladoras, nem se realiza uma análise comparativa entre as técnicas de controle clássicas e as baseadas na inteligência computacional a partir de dados numéricos e experimentais. Além disso, não há conhecimento de trabalhos que associam a inteligência computacional ao controle de aeronaves de asa fixa submetidas a problemas de LOC-I, revelando assim potenciais fatores de inovação na área.

Neste momento, cabe destacar que o uso de técnicas de inteligência computacional no controle de sistemas aeronáuticos, atualmente, em função do elevado rigor técnico necessário à validação das mesmas antes de serem exploradas comercialmente no setor aeronáutico, limita-se majoritariamente a aplicações terrestres não-críticas. Este trabalho, dessa forma, apresenta um foco estritamente acadêmico, embora busque também encorajar a comunidade técnica e científica especializada quanto ao maior uso da inteligência computacional na área.

Ressalta-se que os trabalhos que aplicam as técnicas da inteligência computacional no controle e modelagem de sistemas aeronáuticos propõem inovações associadas à aplicação de técnicas já conhecidas. Neste trabalho, há também a proposição de uma técnica de inteligência computacional inovadora: o método PIA (novo método de inferência fuzzy), o qual apresenta alto desempenho, quando comparado a outros métodos na minimização do erro médio quadrático do processo, e pode ser utilizado em uma diversidade de aplicações.

Esta tese está dividida em fundamentação teórica, revisão bibliográfica, metodologia, PIA: um novo método de inferência fuzzy, controle neuro-fuzzy para aeronaves, conclusões, referências bibliográficas e por fim os anexos. Na Fig. 1.2 são indicados e relacionados os capítulos e as principais seções do texto com os seus respectivos temas.

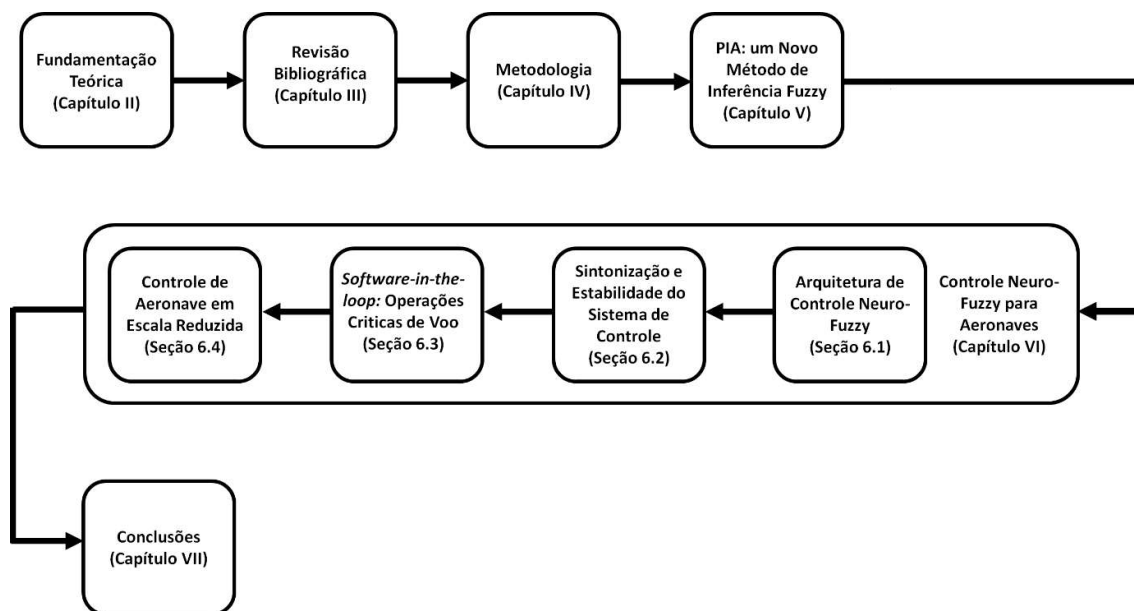


Figura 1.2: Fluxograma – organização da tese (autoria própria).

No intuito de fornecer suporte ao entendimento das principais temáticas envolvidas nesta tese, no próximo capítulo é apresentada a fundamentação teórica que norteia os principais desenvolvimentos presentes neste texto.

CAPÍTULO II

FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho consiste em fornecer ao leitor o conhecimento necessário para o entendimento das principais aplicações e dos conceitos desenvolvidos nesta tese. Este capítulo é dividido em controle e sistemas não-lineares, teoria dos conjuntos fuzzy, métodos de inferência fuzzy, redes neurais, ANFIS, evolução diferencial e modelagem dinâmica de uma aeronave de asa fixa.

2.1. Controle e Sistemas Não-Lineares

O controle não-linear é definido como sendo o conjunto de procedimentos destinados à estabilização ou ao rastreamento das variáveis de saída de um sistema não-linear em relação a uma determinada referência.

Sistemas não-lineares apresentam comportamentos que os diferem dos sistemas lineares, e dentre estes comportamentos, os principais são indicados a seguir (SILVA, 2006):

- Dependência da amplitude de excitação;
- Tempo de escape finito;
- Pontos de equilíbrio múltiplos;
- Não unicidade da solução;
- Elevada sensibilidade aos parâmetros e às condições iniciais (caos);

- Existência de ciclos limite de oscilação;
- Respostas harmônicas e sub-harmônicas a uma entrada senoidal.

De modo a facilitar a análise e o controle de sistemas dessa natureza, correntemente é aplicado sobre os mesmos o processo de linearização (SILVA, 2006). Dada a Eq. (2.1), admitindo que a função $f[x(t), u(t)]$ seja diferenciável, sendo $x(t)$ o estado do sistema e $u(t)$ a ação de controle aplicada ao mesmo, ambos no tempo t :

$$\dot{x}(t) = f[x(t), u(t)], \quad (2.1)$$

esta pode ser expandida em série de Taylor em torno de um ponto $P = (x_p, u_p)$. Ao se realizar a expansão em torno de um ponto de equilíbrio (x_e, u_e) tal que $x_e = 0$ e $u_e = 0$, tem-se que:

$$\dot{x} = Ax + Bu + f_{so}(x, u), \quad (2.2)$$

sendo $A = \left(\frac{\partial f}{\partial x} \right)_{\substack{x=0 \\ u=0}}$ e $B = \left(\frac{\partial f}{\partial u} \right)_{\substack{x=0 \\ u=0}}$.

Desprezando os termos de ordem superior da expansão em série de Taylor $f_{so}(x, u)$, realiza-se a linearização do sistema em torno do ponto de equilíbrio (x_e, u_e) .

Após o processo de linearização, pode ser necessário apresentar o sistema dinâmico na forma discreta, permitindo que o mesmo seja utilizado no projeto de controladores digitais. Na Eq. (2.3) é apresentado o sistema na forma discreta, considerando o segurador de ordem zero, ou seja, mantendo valores constantes de ação de controle u_k durante o intervalo de tempo T_s , sendo T_s o tempo de amostragem adotado para o sistema e k a iteração corrente,

$$x_{k+1} = Gx_k + Hu_k. \quad (2.3)$$

As matrizes G e H são determinadas através das Eqs. (2.4) e (2.5):

$$G = e^{AT_s}, \quad (2.4)$$

$$H = A^{-1}(e^{AT_s} - I)B, \quad (2.5)$$

em que I é a matriz identidade.

A partir do sistema linearizado, é possível aplicar as teorias de controle clássico. Contudo, ao se realizar o processo de linearização, algumas características do sistema passam a ser desprezadas, sobretudo quando se afasta do ponto de equilíbrio em que foi realizada a linearização.

Outro aspecto de importante consideração é a estabilidade de sistemas não-lineares (KHALIL, 2002; SILVA, 2006; PRECUP, TOMESCU e PREITL, 2009). Dado um sistema não-

linear escrito da forma apresentada na Eq. (2.6), em que $x, \dot{x} \in R^{n \times 1}$, é o estado do sistema, $u, \dot{u} \in R^{m \times 1}$, é a ação de controle aplicada ao mesmo, $f(x), \dot{f}(x) \in R^{n \times 1}$, e $b(x), \dot{b}(x) \in R^{n \times m}$, são funções que descrevem a dinâmica do sistema e $x(t_0)$ é o estado inicial no tempo t_0 :

$$\dot{x}(t) = f(x) + b(x)u(x), x(t_0) = x_0, \quad (2.6)$$

é possível definir uma função $V: R^n \rightarrow R, V(x) = x^T P x$, denominada de função Lyapunov, sendo $P \in R^{n \times n}$ uma matriz positiva definida, a partir da qual pode-se iniciar a análise da estabilidade do sistema.

A análise da estabilidade passa pela verificação da derivada temporal da função de Lyapunov \dot{V} , Eq. (2.7). Se $\dot{V}(x) < 0 \forall x$, tem-se que o sistema em malha fechada, composto pelo controlador e pelo processo não-linear, será localmente assintoticamente estável no sentido de Lyapunov na origem x_0 .

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} = \\ \dot{V}(x) &= (f(x) + b(x)u(x))^T P x + x^T P (f(x) + b(x)u(x)) = \\ \dot{V}(x) &= F(x) + B(x), \end{aligned} \quad (2.7)$$

em que:

$$F(x) = f(x)^T P x + x^T P f(x), \quad B(x) = u^T(x) b^T(x) P x + x^T P b(x) u(x). \quad (2.8)$$

Existem técnicas aplicadas em controle, como as redes neurais, a teoria dos conjuntos fuzzy e os sistemas neuro-fuzzy, que levam em consideração as não-linearidades do sistema, e dessa forma são importantes objetos de estudo na área de controle de sistemas não-lineares.

2.2. Teoria dos Conjuntos Fuzzy

Como referência para estudar a teoria dos conjuntos fuzzy, são utilizados (JAFELICE, 2003; GOMIDE, GUDWIN e TANSCHKEIT, 2015), e alguns dos principais conceitos concernentes à área são mencionados em sequência.

Conjunto Fuzzy: Um conjunto fuzzy A definido em um conjunto universo U é o gráfico da função $\mu_A: X \rightarrow [0, 1]$, chamada de função de pertinência de A . Em outras palavras, o conjunto A é dado por:

$$A = \{(x, \mu_A(x)) : x \in X, \mu_A(x) \in [0, 1]\}. \quad (2.9)$$

O subconjunto clássico de U definido por $\text{supp}(A) = \{x \in U : \mu_A(x) > 0\}$ é chamado de suporte de A .

Considerando A como sendo um subconjunto fuzzy de U e $\alpha \in [0, 1]$, o α -nível de A é o subconjunto clássico de U definido por $[A]^\alpha = \{x \in U : \mu_A(x) \geq \alpha\}$, se $0 < \alpha \leq 1$. O nível zero de um subconjunto de A é definido como o menor conjunto fechado (clássico) de U que contém o conjunto suporte de A .

Um subconjunto A é chamado número fuzzy quando o conjunto universo em que μ_A é definida é o conjunto de número reais \mathbb{R} e A satisfaz as seguintes condições:

- Todos os α -níveis de A são não-vazios, com $0 \leq \alpha \leq 1$;
- Todos os α -níveis de A são intervalos fechados de \mathbb{R} ;
- $\text{supp}(A) = \{x \in \mathbb{R} : \mu_A(x) > 0\}$ é limitado.

Os números fuzzy mais comuns são: triangular, trapezoidal e em forma de sino. Outros conceitos importantes dentro da teoria dos conjuntos fuzzy são a s-norma e t-norma.

Uma co-norma triangular (s-norma) é uma operação binária $s : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfazendo as seguintes condições:

- Comutatividade: $xsy = ysx$;
- Associatividade: $xs(ysz) = (xsy)sz$;
- Monotonicidade: Se $x \leq y$ e $w \leq z$ então $xsw \leq ysz$;
- Condições de fronteira: $xs0 = x$, $xs1 = 1$.

Uma norma triangular (t-norma) é uma operação binária $t : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfazendo as seguintes condições:

- Comutatividade: $xty = ytx$;
- Associatividade: $xt(ytz) = (xty)tz$;
- Monotonicidade: Se $x \leq y$ e $w \leq z$ então $xw \leq yz$;
- Condições de fronteira: $0tx = 0$, $1tx = x$.

Claramente, o operador \max é uma s-norma e o operador \min é uma t-norma.

Pode-se relacionar dois ou mais conjuntos fuzzy, e assim modelar, controlar ou classificar sistemas com incertezas, a partir de um Sistema Baseado em Regras Fuzzy (SBRF).

Sistema Baseado em Regras Fuzzy: Contém quatro componentes, a saber, um processador de entrada que realiza a fuzzificação dos dados de entrada, uma coleção de regras nebulosas chamada base de regras, um método de inferência fuzzy e um processador de saída que fornece um número real como saída. Uma vez estabelecida uma base de regras, isto é, como relacionamos os conjuntos fuzzy pela forma “Se...então...”, um SBRF pode ser visto como um mapeamento entre a entrada e a saída da forma $y = f(x)$, $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$. A Fig. 2.1 ilustra a arquitetura de um SBRF.

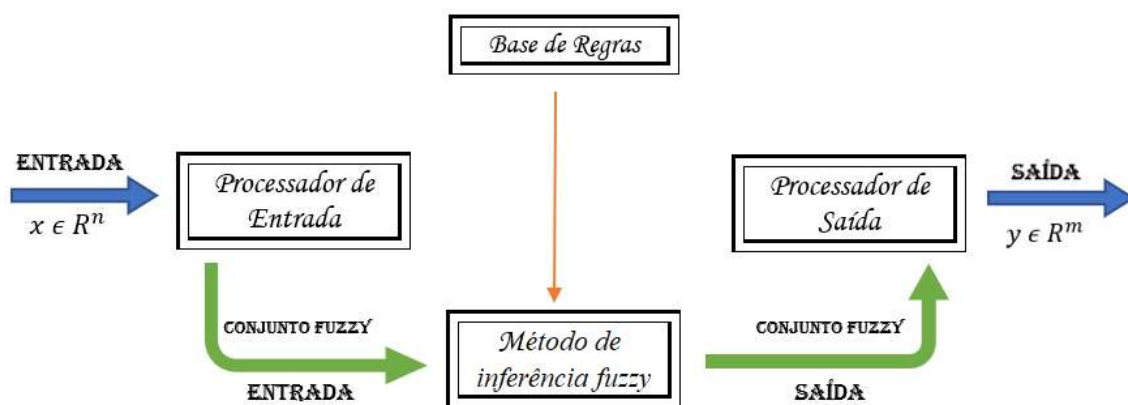


Figura 2.1: Ilustração da arquitetura de um SBRF (autoria própria).

Esta classe de sistema é amplamente utilizada em problemas de modelagem, controle e classificação. Os componentes do SBRF são descritos a seguir:

- **Processador de Entrada (Fuzzificação):** Neste componente as entradas do sistema são traduzidas em conjuntos fuzzy em seus respectivos domínios. É neste momento em que são construídas as funções de pertinências para a descrição das entradas.
- **Base de Regras:** Este componente, juntamente com a máquina de inferência, pode ser considerado o núcleo dos sistemas baseados em regras fuzzy. Ele é composto por uma coleção de proposições fuzzy na forma ‘Se...então...’.
- **Método de Inferência Fuzzy:** É neste componente que cada proposição fuzzy é traduzida matematicamente por meio de métodos de Inferência Fuzzy. Apresentam-se os métodos de Mamdani e de Takagi-Sugeno de Inferência Fuzzy:
 - **Método de Mamdani:** Este método agrega cada uma das regras – ‘Se (antecedente) então (consequente)’ – através do operador lógico *OU*, que é modelado pelo operador

máximo e, em cada regra, o operador lógico E é modelado pelo operador mínimo. Como exemplo, citam-se as regras a seguir:

Regra 1: Se $(X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } B_1)$ então $(Z \text{ é } C_1)$;

Regra 2: Se $(X_1 \text{ é } A_2 \text{ e } X_2 \text{ é } B_2)$ então $(Z \text{ é } C_2)$.

A Fig. 2.2 ilustra como uma saída real z de um sistema de inferência do tipo Mamdani é gerada a partir das entradas x_1 e x_2 reais e a regra de composição $\max - \min$. A saída $z \in \mathbb{R}$ é obtida pela defuzzificação do conjunto fuzzy $C = C'_1 \cup C'_2$ da Fig. 2.2.

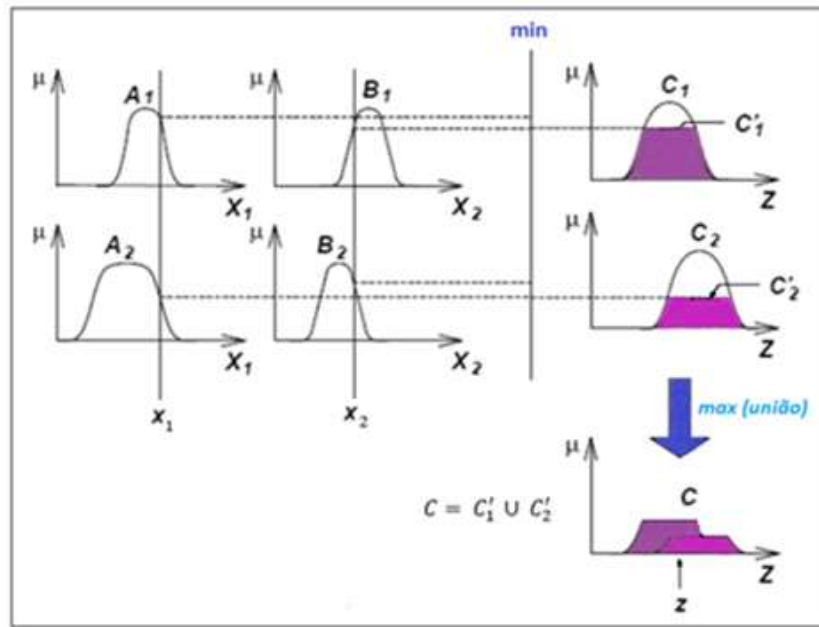


Figura 2.2: Ilustração do método de inferência de Mamdani (autoria própria).

- **Método de Takagi-Sugeno:** A diferença básica deste método com relação ao método de Mamdani está na modelagem dos consequentes do SBRF, os quais são funções do tipo $f_i = f_i(x_1, x_2, \dots, x_n)$ associadas à saída z do SBRF, sendo x_1, x_2, \dots, x_n as entradas (VALLE, 2015; BARROS, BASSANEZI e LODWICK, 2016). Duas regras fuzzy, como exemplo, são mostradas a seguir:

Regra 1: Se $(X_1 \text{ é } A_1 \text{ e } X_2 \text{ é } B_1)$ então $(Z \text{ é } f_1 = f_1(x_1, x_2))$;

Regra 2: Se $(x_1 \text{ é } A_2 \text{ e } x_2 \text{ é } B_2)$ então $(Z \text{ é } f_2 = f_2(x_1, x_2))$.

Na Fig. 2.3, a saída real z de um sistema de inferência fuzzy é gerada a partir das entradas reais x_1 e x_2 .

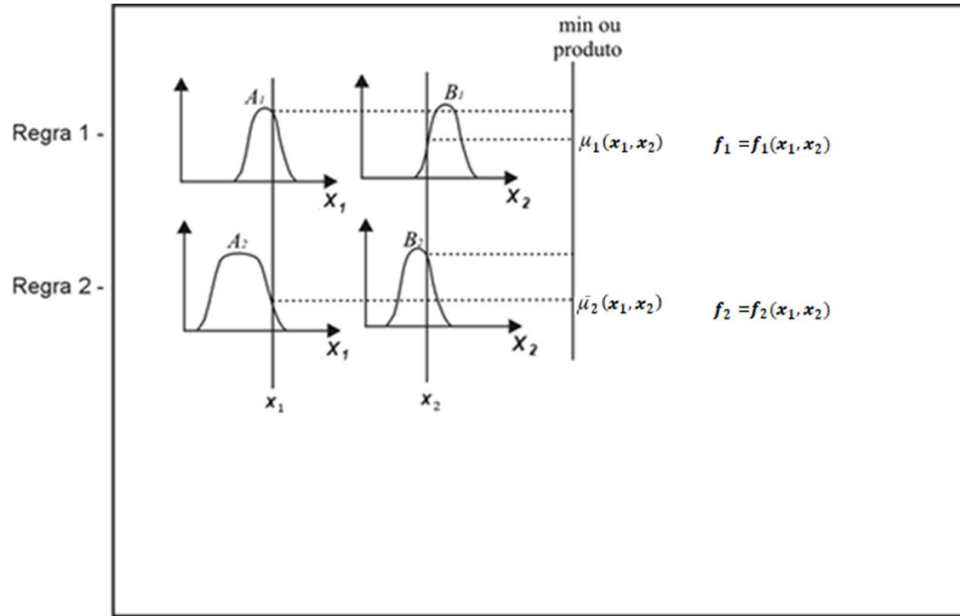


Figura 2.3: Ilustração do método de inferência de Takagi-Sugeno (JAFELICE, 2003).

- **Processador de Saída (Defuzzificação):** Na teoria dos conjuntos fuzzy, pode-se dizer que a defuzzificação é o processo de se representar a combinação de conjuntos fuzzy (ou não), dispostos na saída do processo de inferência, através de um número real. Vários defuzzificadores são propostos na literatura para o método de Mamdani, enquanto que há apenas dois para método de Takagi-Sugeno (MENDEL, 2017). A seguir, apresentam-se os métodos mais comuns de defuzzificação.
 - **Média Ponderada:** É o método de defuzzificação mais usado quando se utiliza o método de inferência fuzzy de Takagi-Sugeno. A saída z para o SBRF é calculada a partir da Eq. (2.10), em que k é o número de funções consequentes associadas à saída do SBRF, e o peso $\mu_i(x_1, x_2, \dots, x_n)$, $i = 1, \dots, k$, é o grau de compatibilidade das entradas x_1, x_2, \dots, x_n , obtido através de uma t-norma, com relação à função consequente $f_i = f_i(x_1, x_2, \dots, x_n)$ (MIZUMOTO, 1995; BARROS, BASSANEZI e LODWICK, 2016).

$$z = \frac{\sum_{i=1}^k \mu_i(x_1, x_2, \dots, x_n) f_i}{\sum_{i=1}^k \mu_i(x_1, x_2, \dots, x_n)}. \quad (2.10)$$

- **Centro de Gravidade:** Este método de defuzzificação é semelhante à média ponderada para distribuição de dados, com a diferença que o peso $\mu_C(Z)$ indica o grau de compatibilidade do valor Z com o conceito modelado pelo conjunto fuzzy C , onde R é a região de integração e z a saída do SBRF. O centro de gravidade é a forma de defuzzificação mais utilizada quando a inferência fuzzy é dada a partir do método de Mamdani, dado por (MIZUMOTO, 1995):

$$z = \frac{\int_R Z \mu_C(Z) dZ}{\int_R \mu_C(Z) dZ}. \quad (2.11)$$

Em função da importância dos métodos de inferência fuzzy no comportamento matemático da estrutura de um SBRF, a próxima seção trata especificamente deste tema.

2.3. Métodos de Inferência Fuzzy

O método de inferência fuzzy é o componente de um SBRF responsável pela tradução matemática de sua base de regras, exercendo grande influência não apenas na resposta do sistema, mas também em seu custo computacional. As técnicas de inferência fuzzy mais utilizadas na literatura são os métodos de Mamdani e de Takagi-Sugeno (BLEJ e AZIZI, 2016).

Há trabalhos que apontam os métodos de Mamdani e Takagi-Sugeno como os principais métodos de inferência fuzzy e propõem comparações cujos resultados demonstram as vantagens do uso de Takagi-Sugeno sobre Mamdani sobretudo com relação a seu melhor desempenho computacional (SIDDIQUE e ADELI, 2013; EYOH e UMOH, 2013; FAHMY, ZAHER e KANDIL, 2015; SHLEEG e ELLABIB, 2013). A literatura contempla trabalhos que propõem outros métodos de inferência fuzzy, ou a possibilidade de adicionar inovações às técnicas existentes, melhorando assim o desempenho computacional e a aplicabilidade de um SBRF (SON, VIET e HAI, 2017).

Há métodos, a exemplo dos de Kosko-Mizumoto e Larsen, que diferem do método Mamdani apenas na definição do conjunto fuzzy de saída. No método de Kosko-Mizumoto (KOSKO, 1994; MIZUMOTO, 1995), Fig. 2.4, o conjunto fuzzy de saída é dado por $C = C'_1 + C'_2$, enquanto no método de Larsen (LARSEN, 1980), a diferença está na obtenção das regiões C'_1 e C'_2 , conforme mostrado na Fig. 2.5.

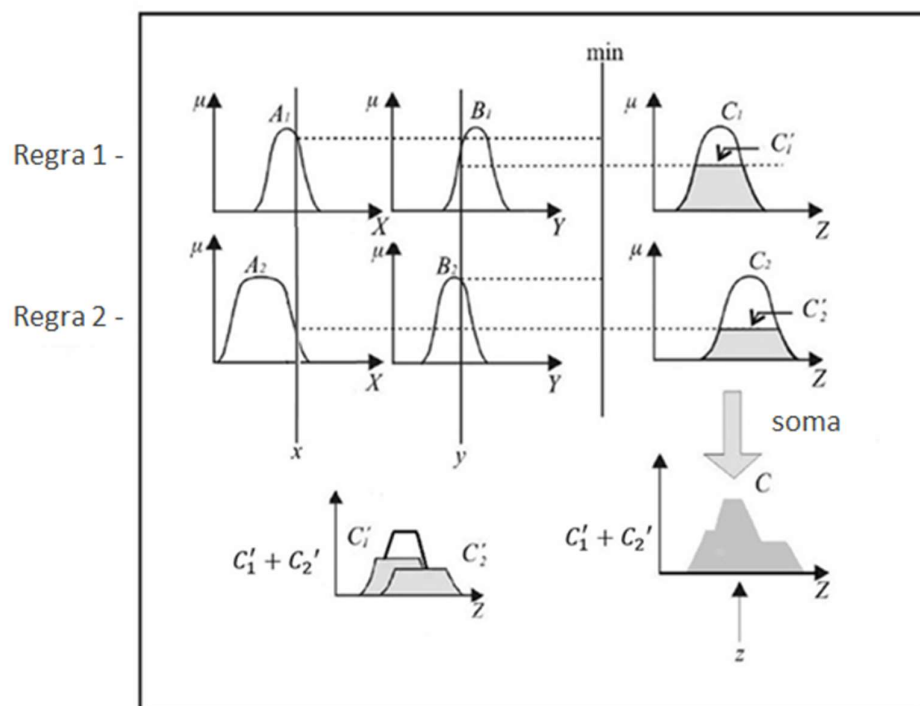


Figura 2.4: Ilustração do método Kosko-Mizumoto (autoria própria).

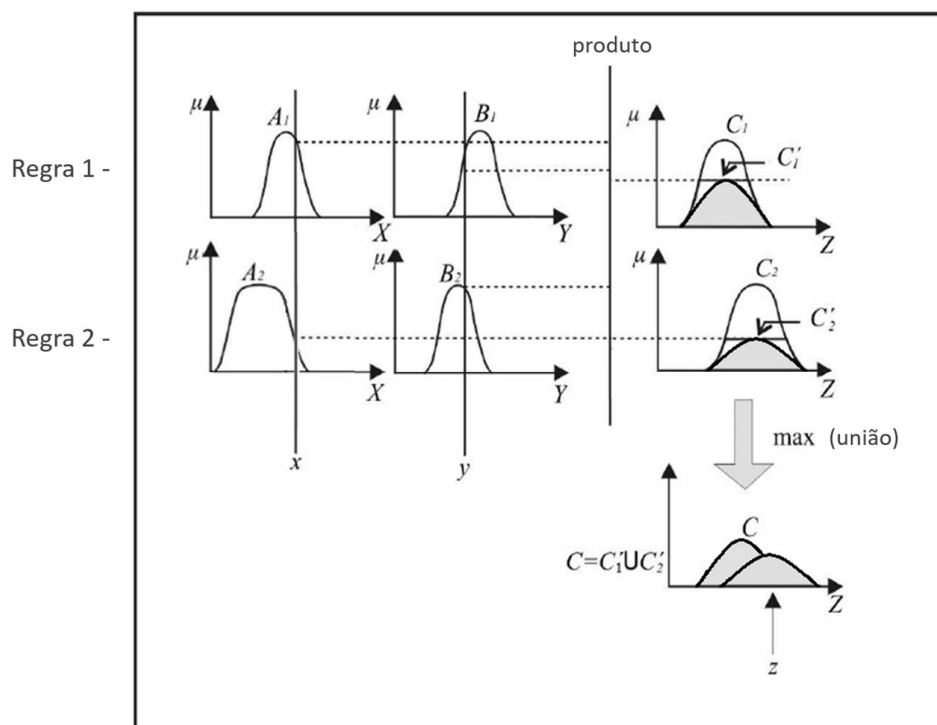


Figura 2.5: Ilustração do método de inferência de Larsen (autoria própria).

Selvachandran *et al.* (2019) propõem um novo sistema de inferência fuzzy aplicado a números complexos, baseado no método de Mamdani, para problemas de tomada de decisão com vários atributos, especialmente quando se trabalha com dados com tendência periódica, como chuvas registradas em uma região ou ondas sonoras produzidas por um instrumento musical.

Os autores em (JEE, TAY e LIM, 2015) desenvolvem uma nova abordagem baseada no método de Takagi-Sugeno para reduzir o número de regras que precisam ser reunidas para a construção de um SBRF. Esta técnica é composta por dois estágios, a saber, no estágio 1, um algoritmo genético é usado para obter um pequeno conjunto de regras a serem coletadas e no estágio 2, as regras restantes são deduzidas aproximadamente por um esquema de raciocínio de similaridade que preserva a monotonicidade das relações matemáticas. Em (ZHANG E SHEN, 2019), os autores desenvolvem uma técnica de interpolação de regras fuzzy para SBRFs que utilizam o método de inferência de Takagi-Sugeno e apresentam base de regras esparsa, no intuito de inferir conclusões para as instâncias sem correspondência.

Outro artigo introduz a noção de partição fuzzy monótona (KERK, TAY, e LIM, 2019), que é útil para a construção de um SBRF monótono baseado no método de Takagi-Sugeno de ordem zero. Os autores desenvolvem um método intervalar para modelar as incertezas de um sistema, que considera o intervalo mínimo de aceitabilidade de uma regra fuzzy, resultando assim em um SBRF monótono e intervalar.

No intuito de aplicar o SBRF a problemas específicos e inovadores, Jamshidi *et al.* (2013) usam o método de Mamdani para a avaliação de risco de dutos, em (SHLEEG e ELLABIB, 2013) há uma comparação do desempenho dos métodos de Mamdani e Takagi-Sugeno na análise do risco de câncer de mama, e em (AHMAD *et al.*, 2019) o método Takagi-Sugeno é utilizado para o diagnóstico inteligente da hepatite.

Dentro do contexto de trabalhos que propõem inovações às técnicas existentes, Bemani e Akbarzadeh (2019) apresentam uma abordagem fuzzy adaptativa híbrida denominada HGFRD (*Hybrid Adaptive Granular Fuzzy Approach to Rule Discovery*), que efetivamente utiliza as vantagens dos métodos de inferência fuzzy de Mamdani e Takagi-Sugeno em um único processo de aprendizagem, dividido em 2 estágios.

O primeiro estágio de aprendizagem do HGFRD se inicia utilizando o método de inferência fuzzy de Mamdani, pois fornece melhor generalização e uma solução inicial mais razoável para o estágio de otimização secundário. Por outro lado, o segundo estágio lança mão do método de Takagi-Sugeno para o procedimento de ajuste fino dos parâmetros do

SBRF, oferecendo assim menor carga de processo, melhor capacidade de aproximação do ótimo global e a possibilidade de utilização de métodos de otimização mais eficazes, levando a uma melhor precisão do modelo.

Antes do início do segundo estágio, a estrutura e os parâmetros obtidos do método de Mamdani devem ser transformados em uma estrutura e em parâmetros do método de Takagi-Sugeno que sejam matematicamente equivalentes. Enquanto as partes antecedentes das regras fuzzy permanecem intactas durante essa transformação, a parte consequente de cada regra é reestruturada para o formato compatível com o método de Takagi-Sugeno equivalente, de modo que o valor de cada coeficiente seja inicializado pelo valor de centro correspondente à função de pertinência de saída do método de Mamdani.

A Fig. 2.6 ilustra a arquitetura da abordagem HGFRD proposta em (BEMANI e AKBARZADEH, 2019):

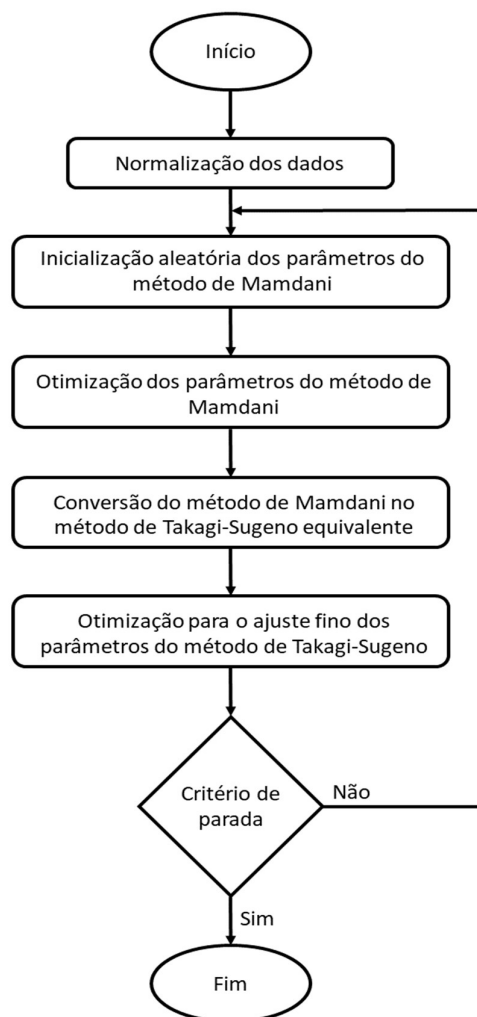


Figura 2.6: Arquitetura da abordagem HGFRD (BEMANI e AKBARZADEH, 2019).

O HGFRD é aplicado a dez *benchmarks* padrão e os resultados são comparados com outras vinte e duas estratégias recentes. Resultados numéricos confirmam que o HGFRD atinge maior capacidade de ajuste a dados experimentais, e com baixo custo computacional; embora esta tese proponha o método PIA de inferência fuzzy, que demonstra apresentar resultados mais satisfatórios que os obtidos com o HGFRD na resolução do estudo de caso exposto neste trabalho.

Em certas aplicações, é desejável que os parâmetros relacionados a um SBRF sejam obtidos a partir de algum método de otimização. Jang (1993) propôs um sistema neuro-fuzzy, denominado ANFIS, a ser explanado na seção 2.5, que mantém a estrutura do sistema de inferência fuzzy, mas que alia também a capacidade de adaptação de uma rede neural, tema explorado na próxima seção.

2.4. Redes Neurais

2.4.1. Conceitos Gerais

Uma Rede Neural Artificial (RNA), Fig. 2.7, é uma estrutura matemática constituída por um número finito de unidades individualizadas, também designadas por neurônios, Fig. 2.8, organizados em camadas (SILVA, 2006). A rede neural é utilizada, dentre outras aplicações, para o reconhecimento de padrões, para a obtenção de modelos para sistemas estáticos e dinâmicos e também como estrutura matemática de controladores, apresentando a característica de ser adequada a uma série de problemas devido à sua capacidade de ajuste e adaptação aos problemas.

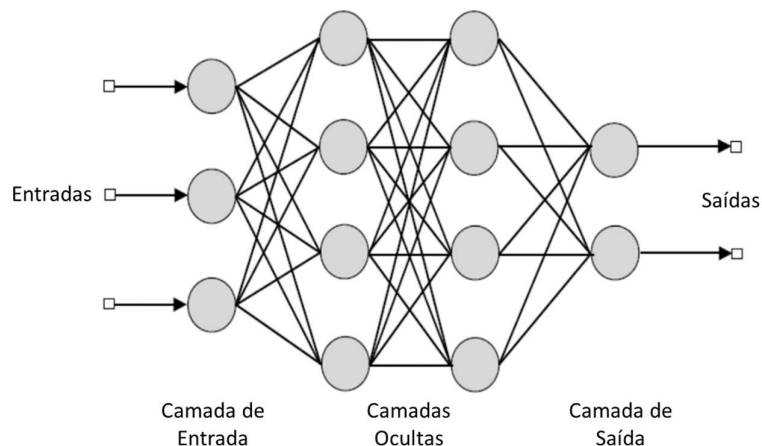


Figura 2.7: Representação da arquitetura de uma RNA multicamadas (SILVA, 2006).

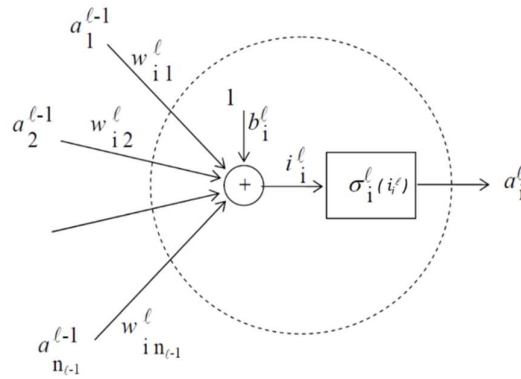


Figura 2.8: Esquema de um neurônio i da camada l (SILVA, 2006).

A saída a_i^l de cada neurônio i da camada l é apresentada na Eq. (2.12), sendo σ_i^l a função de ativação de cada neurônio i da camada l da rede neural, cuja saída é limitada geralmente ao intervalo $[-1, 1]$ e é a responsável por prover à estrutura matemática a característica desejada de não-linearidade. Os parâmetros w_{ij}^l e b_i^l são respectivamente o peso sináptico e o valor do *bias* associados ao neurônio i , ambos determinados durante o processo de treinamento da rede neural, sendo:

$$i_i^l = \sum_{j=1}^{n_{l-1}} w_{ij}^l a_j^{l-1} + b_i^l,$$

$$a_i^l = \sigma_i^l(i_i^l). \quad (2.12)$$

A Fig. 2.9 é a representação das funções de ativação σ_i^l usualmente utilizadas de cada neurônio i da camada l : degrau, linear, sigmoide ou tangente hiperbólica.

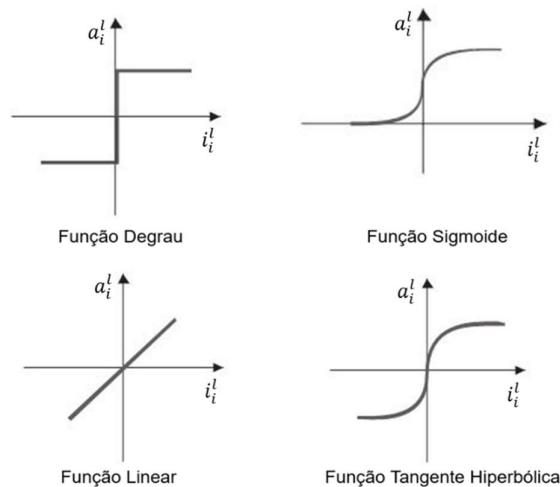


Figura 2.9: Funções de ativação usualmente utilizadas (autoria própria).

2.4.2. Treinamento da Rede Neural

Após a fixação do número de camadas e do número de neurônios em cada camada, procedimento realizado geralmente de forma intuitiva, é necessário determinar os valores dos parâmetros w_{ij}^l e b_i^l que conduzem ao melhor desempenho da rede, processo denominado treinamento da rede neural.

O treinamento da rede neural é tradicionalmente executado através de métodos de otimização inspirados nos métodos da máxima descida e de métrica variável (HAYKIN, 2008) – embora haja na literatura trabalhos em que métodos heurísticos, como a evolução diferencial e os algoritmos genéticos, passam a ser utilizados para tal função (PEREIRA *et al.*, 2016) – e baseia-se na minimização da função custo Cf , Eq. (2.13):

$$Cf = \sum_{k=1}^{Nd} [y(k) - \hat{y}(k)]^2, \quad (2.13)$$

em que Nd é o número de elementos da sequência $y(k)$; $y(k), k = 1, \dots, Nd$, é o conjunto de dados utilizados para o treinamento da rede; e $\hat{y}(k), k = 1, \dots, Nd$, são os valores obtidos na saída da rede neural. Ao se realizar com sucesso o processo de minimização da função custo, espera-se que a rede neural, para os valores de entrada que geram $y(k)$, apresente valores de saída $\hat{y}(k)$ tais que $\hat{y}(k) \simeq y(k), k = 1, \dots, Nd$.

O treinamento da RNA pode ser realizado de duas formas: em grupo, chamado alternativamente de treinamento off-line; e em linha, também chamado de on-line ou em tempo real.

O treinamento em grupo é utilizado para processos invariantes no tempo ou para obter valores iniciais dos parâmetros da rede. O valor de Nd deve ser da ordem das centenas, o que permite ter uma grande variedade de amplitudes e informações do processo. O número de iterações utilizado para obter a convergência da rede é variável e é função da respectiva rapidez de convergência do processo de otimização e do valor aceitável para Cf . Após o treino em grupo, as redes devem ser testadas utilizando outras sequências de entrada/saída para verificar a sua boa capacidade de generalização.

Já o treinamento em linha é necessário quando se precisa identificar um processo variante no tempo. Ressalta-se que para garantir a rápida adaptação da rede ao sistema, o número de iterações realizadas durante o treinamento por cada passo de execução, por exemplo, de um sistema controle, deverá ser no mínimo na ordem de dezenas, conforme é ilustrado na Fig. 2.10. Em sistemas não-lineares, porém, o treinamento em linha pode causar

o esquecimento das características globais da rede, ficando esta mais adaptada apenas para pontos de funcionamento com características locais.

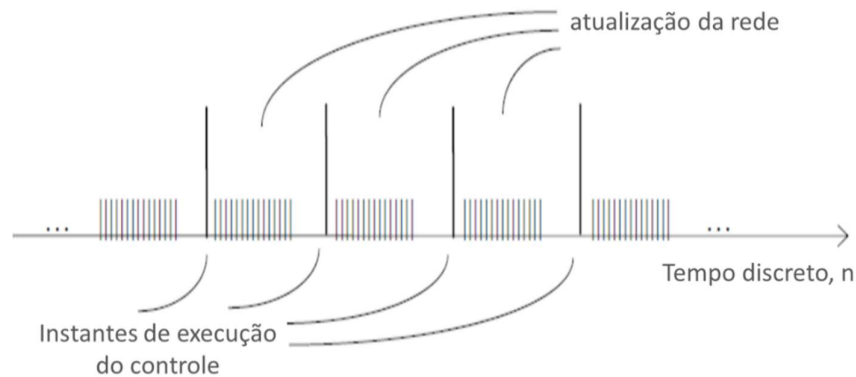


Figura 2.10: Relação temporal entre os instantes de execução de um sistema de controle e o número de iterações realizadas para a atualização da rede (SILVA, 2006).

2.4.3. Aplicação de Redes Neurais em Sistemas de Controle Adaptativos

De acordo com Haykin (2008), há basicamente duas formas de se trabalhar com redes neurais aplicadas a sistemas de controle adaptativos: controle direto adaptativo e indireto adaptativo.

Nos métodos de controle indireto adaptativo, a obtenção dos pesos e *bias* da rede neural não está ligada diretamente ao controle, mas sim à estimação da dinâmica da planta, Fig. 2.11. Nesta figura, a rede neural artificial encontra-se implementada no bloco “Estimação do modelo da planta” (cujo detalhamento é apresentado na Fig. 2.12), e a partir deste modelo em RNA é realizada a sintonia do controlador.

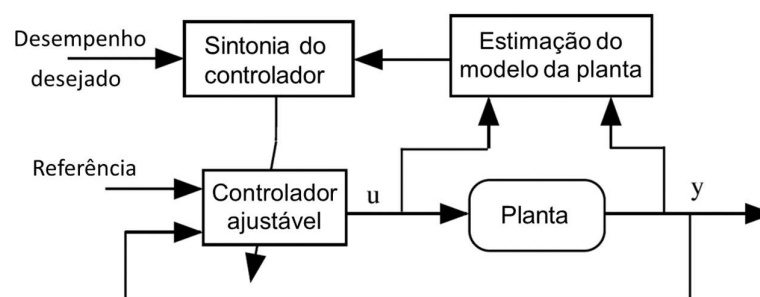


Figura 2.11: Princípio do controle indireto adaptativo (LANDAU *et al.*, 2011).

Neste método de controle é utilizado um procedimento subdividido em 2 passos, a saber:

- Obtém-se um modelo da planta P através de uma amostra contendo dados de entrada e saída da mesma. Para exemplificar, considere um sistema de uma entrada $u(k)$ e uma saída $y(k)$ para o tempo discreto k , Figs. 2.11 e 2.12. O modelo de identificação da rede neural artificial com realimentação da entrada, \hat{P} , pode ser dado da forma:

$$\hat{y}(k+1) = f(y(k), \dots, y(k-q+1), u(k), \dots, u(k-q+1)), \quad (2.14)$$

em que q é a ordem do sistema desconhecido e \hat{y} é a estimativa da saída y . A estimativa $\hat{y}(k+1)$ é então subtraída de $y(k+1)$ (saída real do sistema, obtida por meio de sensores) para produzir o sinal de erro:

$$e(k+1) = y(k+1) - \hat{y}(k+1). \quad (2.15)$$

O erro $e(k+1)$ é usado para ajustar os pesos sinápticos da rede neural, de modo a minimizar o erro na estimação da saída y durante o processo de treinamento da estrutura. Ressalta-se que é a saída real do sistema (e não a saída do modelo de identificação) que é realimentada para a entrada do modelo. A Fig. 2.12 ilustra o procedimento de obtenção do modelo de identificação \hat{P} , em que Z^{-1} representa um atraso discreto no tempo, e considerando-se que o sistema apresenta ordem $q = 1$.

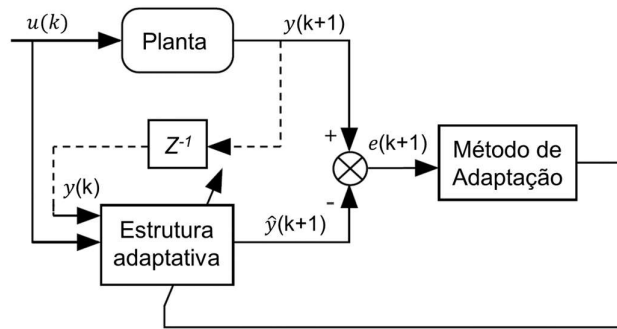


Figura 2.12: Esquema básico da estimação on-line dos parâmetros da rede neural (LANDAU *et al.*, 2011).

- O modelo identificado da planta é utilizado para se obter as estimativas da saída do sistema em função da entrada imposta no mesmo. Dessa forma, a partir deste, é possível projetar controladores adaptativos, conforme verificado na Fig. 2.11, que obtêm o conjunto de ações de controle capaz de garantir que o sistema controlado atinja o desempenho desejado.

Já nos métodos de controle direto adaptativo, a dinâmica da planta é desconhecida, e dessa forma o controlador e a planta formam um sistema realimentado de laço fechado cujas

entradas são a referência e uma versão atrasada da saída da planta; e a saída do sistema de controle é a correspondente ação de controle a ser aplicada no mesmo, Fig. 2.13.

Em (HAYKIN, 2008), os valores dos pesos e dos *bias* são diretamente ajustados para melhorar o desempenho do sistema, embora não existam métodos precisos para ajustar estes parâmetros, visto que a planta desconhecida encontra-se entre o controlador e o erro de saída.

O esquema do controle direto adaptativo, apresentado na Fig. 2.13, baseia-se na observação de que a diferença entre a saída da planta e a saída do modelo de referência, chamada subsequentemente de erro planta-modelo, é uma medida comparativa entre o desempenho real do controlador e o desejado. Esta informação, juntamente com outras, é utilizada pelo método de adaptação para ajustar diretamente em tempo real os parâmetros do controlador adaptativo, de modo a forçar, de forma assintótica, o erro planta-modelo para zero (LANDAU *et al.*, 2011).

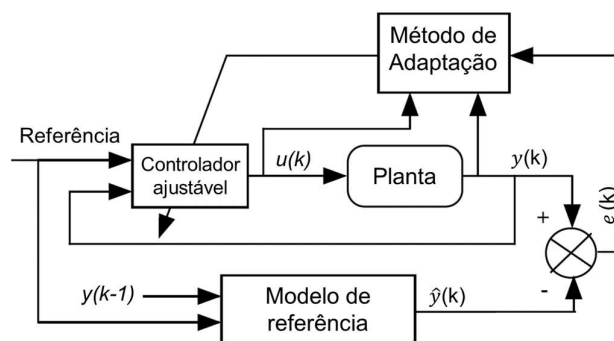


Figura 2.13: Esquema do controle direto adaptativo (LANDAU *et al.*, 2011).

O controle direto adaptativo, porém, apresenta algumas limitações. Embora o desempenho do sistema possa, em muitos casos, ser especificado em termos de um modelo de referência, as condições para a existência de um controlador viável que permita, em malha fechada, seguir esse modelo são restritivas (LANDAU *et al.*, 2011).

Na próxima seção são apresentadas as principais características do ANFIS, sistema híbrido que alia as características de um sistema de inferência fuzzy, com a adaptabilidade de uma rede neural.

2.5. ANFIS

O ANFIS (*Adaptive-Network-based Fuzzy Inference System*), surgiu da fusão da RNA com o SBRF, Fig. 2.14. Dessa forma, o ANFIS herda a estrutura de um SBRF, ou seja, contém o processo de fuzzificação, uma base de regras do tipo *Se ... Então*, uma máquina de inferência fuzzy e a defuzzificação; porém com a mesma representação da estrutura e capacidade de aprendizado e adaptação de uma RNA através do ajuste dos parâmetros relacionados ao sistema fuzzy.

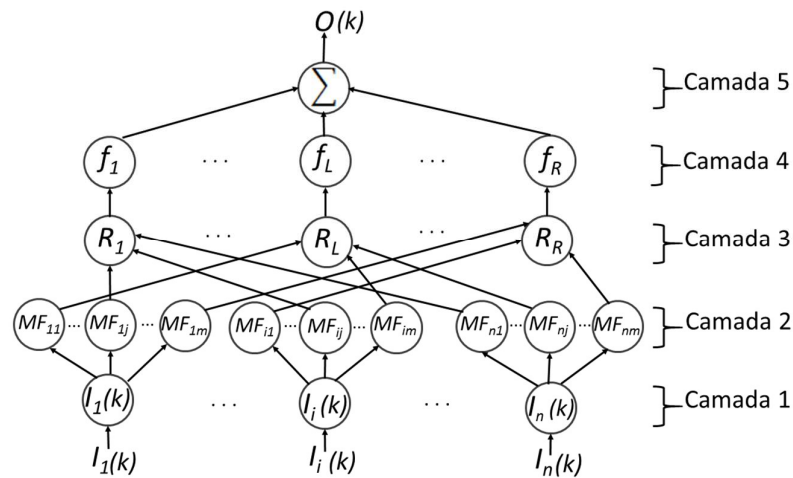


Figura 2.14: Representação de um sistema neuro-fuzzy (autoria própria).

A seguir é indicada a composição e a função de cada uma das camadas do ANFIS, considerando a utilização do método de inferência fuzzy de Takagi-Sugeno (TAKAGI e SUGENO, 1985; JANG, 1993; CHEN, LIN e LIN, 2009):

- **Camada 1:** Cada nó da primeira camada recebe uma das variáveis de entrada $I_i(k)$ do sistema neuro-fuzzy, em que a saída do nó i da camada 1 no tempo k , $u_i^{(1)}(k)$, é dada por:

$$u_i^{(1)}(k) = I_i(k). \quad (2.16)$$

- **Camada 2:** Realização da fuzzificação dos dados de entrada, ou seja, estes são traduzidos em conjuntos fuzzy em seus respectivos domínios. É neste momento em que são construídas as funções de pertinência MF para a descrição das entradas. Adotando funções de pertinência do tipo gaussiana, a saída do nó ij da camada 2 no tempo k , $u_{ij}^{(2)}(k)$, é dada por:

$$u_{ij}^{(2)}(k) = e^{-\frac{(I_i(k) - \mu_{ij})^2}{\sigma_{ij}^2}}, \quad (2.17)$$

em que $m_{ij}(k)$ e $\sigma_{ij}^2(k)$ são respectivamente a média e a variância da função de pertinência gaussiana MF_{ij} .

- **Camada 3:** Coleção de proposições do tipo Se ... Então, formando assim a base de regras do ANFIS. Para toda regra R_L , os operadores lógicos E e OU são modelados respectivamente através dos operadores matemáticos produto e máximo, de modo que a saída do nó L desta camada, $u_L^{(3)}(k)$, seja função das saídas da camada 2 selecionadas a partir da regra R_L .
- **Camada 4:** Cada nó desta camada, denominado nó consequente, realiza o papel de um consequente, que é definido como sendo uma função $f_L: R^n \rightarrow R$ tal que $f_L = f_L(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k)$, em que $w_{1L}, \dots, w_{jL}, \dots, w_{oL}$ são pesos a serem determinados na fase de treinamento do ANFIS. Dessa forma, a saída do nó L da camada 4 $u_L^{(4)}(k)$ é calculada por:

$$u_L^{(4)}(k) = u_L^{(3)}(k) f_L(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k). \quad (2.18)$$

- **Camada 5:** Determinação da saída do sistema neuro-fuzzy, dada pela Eq. (2.19):

$$O(k) = \frac{\sum_{L=1}^R u_L^{(4)}(k)}{\sum_{L=1}^R u_L^{(3)}(k)}. \quad (2.19)$$

Considerando que o ANFIS seja invariante no tempo e que o método de inferência utilizado seja o de Takagi-Sugeno de primeira ordem, os consequentes passam a ser polinômios de primeira ordem, e assim tem-se que:

$$O = \frac{\sum_{L=1}^R u_L^{(4)}}{\sum_{L=1}^R u_L^{(3)}} = \frac{\sum_{L=1}^R u_L^{(3)}(w_{0L} + w_{1L}I_1 + \dots + w_{iL}I_i + \dots + w_{nL}I_n)}{\sum_{L=1}^R u_L^{(3)}}. \quad (2.20)$$

Quando se trabalha com problemas de modelagem ou controle neuro-fuzzy, é necessário que seja utilizado um método de otimização que minimize a função custo relacionada ao processo e consequentemente busque uma solução de interesse, que passa por exemplo pela determinação dos pesos da estrutura matemática. Dessa forma, na próxima seção, será apresentada a evolução diferencial, o método de otimização utilizado nesta tese.

2.6. Evolução Diferencial

A evolução diferencial, algoritmo desenvolvido por Storn e Price (1997), é um método de minimização de funções, podendo ser até não lineares e não diferenciáveis, que se mostra capaz de se chegar ao valor ótimo global até mais facilmente e rapidamente que outros

métodos clássicos e heurísticos. O método requer poucas variáveis de controle, é robusto, de fácil implementação e apresenta boa adaptabilidade à computação paralela (STORN e PRICE, 1997).

O método da evolução diferencial busca encontrar o valor mínimo global de uma função $f_{obj}(X)$, sendo X um vetor cujos elementos representam cada uma das variáveis independentes da função. As etapas e operadores constituintes do algoritmo estão descritos a seguir (HAN, LIN e CHANG, 2013) e representados na Fig. 2.15.

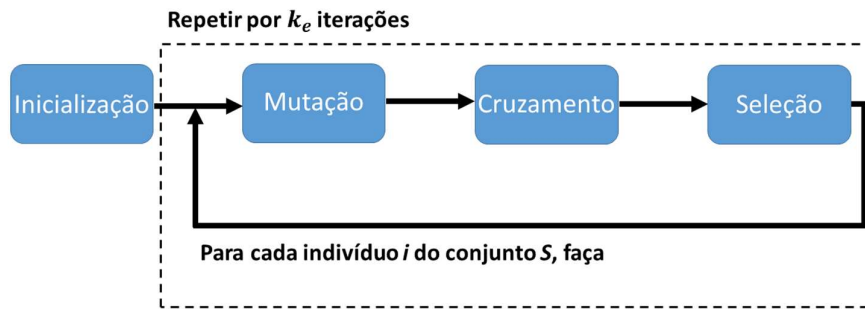


Figura 2.15: Representação da evolução diferencial, adaptada de Mór *et al.* (2015), sendo S o conjunto das possíveis soluções do problema de otimização.

Durante a **inicialização** do algoritmo, os valores dos parâmetros utilizados durante a execução do método são definidos, e são também determinados os limites de busca para cada variável. Os vetores (ou indivíduos) $X_{i,0}$, cada um representando uma possível solução para o problema de otimização e cujos elementos são os valores normalizados das variáveis do processo, são inicializados seguindo uma distribuição uniforme definida dentro do intervalo $[0,1]$, e em seguida é calculada a aptidão de todos os vetores com relação à função objetivo f_{obj} .

Na fase de **mutação**, utiliza-se a alteração dos vetores $X_{i,k}$ a partir da Eq. (2.21), em que $V_{i,k}$ é o vetor resultante obtido a partir do vetor $X_{i,k}$ na iteração k , F é o fator de perturbação do vetor $X_{i,k}$, geralmente apresentando valor contido no intervalo $[0,2]$, e $X_{j,k}$ e $X_{l,k}$ são outros vetores escolhidos aleatoriamente pertencentes ao conjunto (ou população) S das soluções possíveis para o problema de otimização também na iteração k . Esse procedimento é realizado n vezes, em que n é o número de elementos do conjunto S :

$$V_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}). \quad (2.21)$$

Na fase de **cruzamento** (ou recombinação), busca-se mesclar as informações presentes nos vetores $V_{i,k}$ e $X_{i,k}$, garantindo maior diversidade na população. Para cada elemento e do vetor $V_{i,k}$, utiliza-se um gerador de valores uniformemente distribuídos dentro do intervalo $[0,1]$, $rand(0,1)$, e compara-se com o valor da taxa de cruzamento (CR), também definido dentro do intervalo $[0,1]$. A tomada de decisão desse operador é especificada na Eq. (2.22), em que $C_{i,k}$ é o vetor candidato a participar do conjunto S na iteração $k + 1$ e $randnum(1, n_e)$, sendo n_e o número de elementos dos vetores, é um gerador de números inteiros uniformemente distribuídos dentro do intervalo $[1, n_e]$. Esse procedimento é repetido para todos os vetores do conjunto S :

$$C_{e,i,k} = \begin{cases} V_{e,i,k}, & \text{SE } rand(0,1) \leq CR \text{ OU } e = randnum(1, n_e); \\ X_{e,i,k}, & \text{SE } rand(0,1) > CR \text{ E } e \neq randnum(1, n_e). \end{cases} \quad (2.22)$$

A **seleção** é o último operador padrão da evolução diferencial e que se baseia na análise dos vetores $C_{i,k}$ gerados no cruzamento. A Eq. (2.23) descreve o procedimento de seleção, a ser realizado também para todos os vetores do conjunto S . Após a seleção, avança-se para a iteração $k + 1$, e é repetido todo o procedimento a partir do operador mutação até que na iteração k_e alguma condição de parada pré-determinada seja atendida:

$$X_{i,k+1} = \begin{cases} X_{i,k}, & \text{se } f_{obj}(X_{i,k}) \leq f_{obj}(C_{i,k}); \\ C_{i,k}, & \text{outro caso.} \end{cases} \quad (2.23)$$

2.7. Modelagem Dinâmica de uma Aeronave de Asa Fixa

As equações do movimento para qualquer configuração tradicional de aeronave, como helicópteros e aviões, podem ser derivadas a partir da segunda lei de Newton aplicada a corpos rígidos com 6 graus de liberdade, possuindo assim 3 graus de translação e 3 graus de rotação (MACHINI, 2016).

Utiliza-se para a modelagem dois referenciais, sendo eles: um referencial inercial fixo à terra denotado por I_{CS} e outro baricêntrico fixo à aeronave B_{CS} capaz de transladar e rotacionar junto com a mesma. Uma vez que as forças e momentos, sejam elas aerodinâmicas ou inerciais, atuam no corpo, as equações do movimento são derivadas em relação ao referencial B_{CS} . Por consequência, pode-se definir qualquer vetor escrito no referencial B_{CS} no referencial I_{CS} utilizando os ângulos de Euler. Estes representam a atitude da aeronave, que é definida por: rolagem (ϕ), arfagem (θ) e guinada (ψ).

A posição da aeronave em relação ao eixo inercial I_{cs} , bem como os ângulos de Euler, são dispostos no vetor $\vec{\eta}$, Eq. (2.24).

$$\vec{\eta} = \begin{bmatrix} X_E \\ Y_E \\ Z_E \text{ ou } h \\ \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{posição } x \text{ em relação a } I_{cs} \\ \text{posição } y \text{ em relação a } I_{cs} \\ \text{posição } z \text{ em relação a } I_{cs} \\ \text{ângulo de rolagem} \\ \text{ângulo de arfagem} \\ \text{ângulo de guinada} \end{bmatrix}. \quad (2.24)$$

As componentes das velocidades lineares e angulares em relação ao referencial B_{cs} são arranjadas no vetor \vec{v} e expressas a partir da Eq. (2.25).

$$\vec{v} = \begin{bmatrix} U \\ V \\ W \\ P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \text{velocidade longitudinal} \\ \text{velocidade lateral} \\ \text{velocidade vertical} \\ \text{velocidade angular de rolagem} \\ \text{velocidade angular de arfagem} \\ \text{velocidade angular de guinada} \end{bmatrix}. \quad (2.25)$$

Por fim, na Eq. (2.26), são visualizadas as componentes das forças e momentos que agem sobre a aeronave.

$$\begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \text{força longitudinal} \\ \text{força transversal} \\ \text{força vertical} \\ \text{momento de rolagem} \\ \text{momento de arfagem} \\ \text{momento de guinada} \end{bmatrix}. \quad (2.26)$$

A partir da Fig. 2.16 é possível visualizar os eixos, os ângulos de Euler, as velocidades, forças e momentos definidos anteriormente.

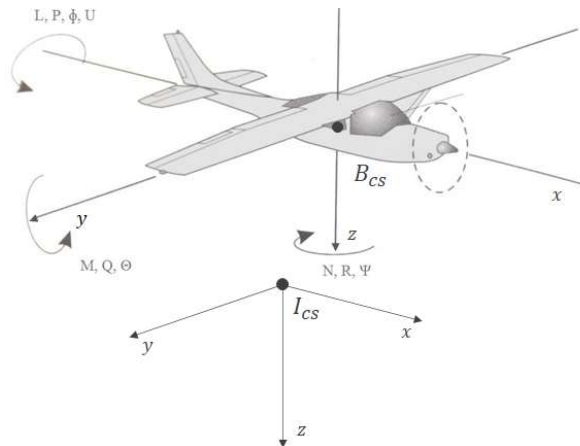


Figura 2.16: Visualização dos eixos, ângulos de Euler, velocidades, forças e momentos da aeronave (autoria própria).

Após a definição das variáveis envolvidas na modelagem dinâmica da aeronave, é necessária a análise cinemática do sistema. A equação cinemática para a translação e rotação de um corpo rígido fixo pode ser expressa da forma:

$$\begin{bmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{bmatrix} = R_{z,\psi} R_{y,\theta} R_{x,\phi} \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \quad (2.27)$$

onde $R_{z,\psi}$, $R_{y,\theta}$ e $R_{x,\phi}$ são as matrizes de rotação do referencial B_{CS} para I_{CS} nas direções z , y e x respectivamente (ROSKAM, 2001).

Desta forma, a velocidade angular da aeronave escrita no referencial B_{CS} pode se relacionar com a taxa de variação dos ângulos de Euler utilizando das matrizes de rotação que resultam na seguinte relação cinemática:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + R_{x,\phi}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_{x,\phi}^T R_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \quad (2.28)$$

Aplicando a segunda lei de Newton para um corpo rígido com 6 graus de liberdade, sendo “ \times ” o operador indicativo de produto vetorial, tem-se que:

$$m(\vec{\dot{v}}_1 + \vec{v}_2 \times \vec{v}_1) = \vec{\tau}_1, \quad (2.29)$$

$$I_{CG} \vec{\dot{v}}_2 + \vec{v}_2 \times (I_{CG} \vec{v}_2) = \vec{\tau}_2, \quad (2.30)$$

em que m é a massa da aeronave, I_{CG} é seu tensor de inércia, $\vec{v}_1 = [U, V, W]^T$, $\vec{v}_2 = [P, Q, R]^T$, $\vec{\tau}_1 = [X, Y, Z]^T$ e $\vec{\tau}_2 = [L, M, N]^T$.

A partir do desenvolvimento das Eqs. (2.29) e (2.30) apresentado em Fossen (2011), sendo g a aceleração da gravidade, tem-se:

$$\begin{bmatrix} m(\dot{U} + QW - RV + g \sin(\theta)) \\ m(\dot{V} + UR - WP - g \cos(\theta) \sin(\phi)) \\ m(\dot{W} + VP - QU - g \cos(\theta) \cos(\phi)) \\ I_x \dot{P} - I_{xz}(\dot{R} + PQ) + (I_z - I_y)QR \\ I_y \dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR \\ I_z \dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix}, \quad (2.31)$$

sendo I_x , I_y e I_z os momentos de inércia da aeronave em torno respectivamente dos eixos x , y e z , e I_{xz} é o produto de inércia em relação aos eixos x e z .

As Eqs. (2.28) e (2.31) modelam o comportamento dinâmico da aeronave e formam um sistema possível e determinado com nove variáveis e nove equações. Essas equações

podem ser integradas ao longo do tempo a fim de se obter a variação temporal das variáveis de estado. O processo de integração, porém, se contabilizar todas as não-linearidades do sistema, torna-se um processo matematicamente complexo e muitas vezes inviável sobretudo quando se busca uma solução analítica. Assim, utilizam-se ferramentas de linearização em torno de pontos de operação da aeronave, como a teoria das pequenas perturbações, que simplificam consideravelmente o processo de integração numérica.

A teoria das perturbações define que é possível escrever um estado s como a soma do seu valor nominal (constante) s_0 , que representa uma condição de equilíbrio do sistema, e uma perturbação no entorno desta condição de equilíbrio δs . A partir disto, definem-se as variáveis apresentadas anteriormente na sua forma perturbada:

$$\vec{\tau} = \vec{\tau}_0 + \vec{\delta\tau} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta L \\ \delta M \\ \delta N \end{bmatrix}, \quad (2.32)$$

$$\vec{v} = \vec{v}_0 + \vec{\delta v} = \begin{bmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \\ Q_0 \\ R_0 \end{bmatrix} + \begin{bmatrix} \delta U \\ \delta V \\ \delta W \\ \delta P \\ \delta Q \\ \delta R \end{bmatrix} = \begin{bmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \\ Q_0 \\ R_0 \end{bmatrix} + \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}, \quad (2.33)$$

$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \phi_0 \\ \psi_0 \end{bmatrix} + \begin{bmatrix} \delta\theta \\ \delta\phi \\ \delta\psi \end{bmatrix} = \begin{bmatrix} \theta_0 \\ \phi_0 \\ \psi_0 \end{bmatrix} + \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix}. \quad (2.34)$$

Segundo Fossen (2011), as equações lineares perturbadas que representam o movimento da aeronave são dadas por:

$$\begin{bmatrix} m[\dot{u} + Q_0 w + W_0 q - R_0 v - V_0 r + g \cos(\theta_0) \theta] \\ m[\dot{v} + U_0 r + R_0 u - W_0 p - P_0 w - g \cos(\theta_0) \cos(\phi_0) \phi + g \sin(\theta_0) \sin(\phi_0) \theta] \\ m[\dot{w} + V_0 p + P_0 v - U_0 q - Q_0 u + g \cos(\theta_0) \sin(\phi_0) \phi + g \sin(\theta_0) \cos(\phi_0) \theta] \\ I_x \dot{p} - I_{xz} \dot{r} + (I_z - I_y)(Q_0 r + R_0 q) - I_{xz}(P_0 q + Q_0 p) \\ I_y \dot{q} + (I_x - I_z)(P_0 r + R_0 p) - 2I_{xz}(R_0 r + P_0 p) \\ I_z \dot{r} - I_{xz} \dot{p} + (I_y - I_x)(P_0 q + Q_0 p) + I_{xz}(Q_0 r + R_0 q) \end{bmatrix} = \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta L \\ \delta M \\ \delta N \end{bmatrix}. \quad (2.35)$$

Na Eq. (2.35), as variáveis que possuem sub-índice “0” são denominadas variáveis de equilíbrio ou de trimagem, as quais são obtidas, Eq. (2.36), assumindo na Eq. (2.31) que a aeronave esteja em equilíbrio dinâmico:

$$\begin{bmatrix} m(Q_0 W_0 - R_0 V_0 + g \sin(\theta_0)) \\ m(U_0 R_0 - P_0 W_0 - g \cos(\theta_0) \sin(\phi_0)) \\ m(P_0 V_0 - Q_0 U_0 - g \cos(\theta_0) \cos(\phi_0)) \\ (I_z - I_y)Q_0 R_0 - P_0 Q_0 I_{xz} \\ (P_0^2 - R_0^2)I_{xz} + (I_x - I_z)P_0 R_0 \\ (I_y - I_x)P_0 Q_0 + Q_0 R_0 I_{xz} \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{bmatrix}. \quad (2.36)$$

No intuito de novamente simplificar a modelagem matemática de uma aeronave, é comum desacoplar sua dinâmica de corpo rígido em 2 subsistemas: dinâmica longitudinal, composta pelos estados u, w, q e θ ; e dinâmica látero-direcional, composta pelos estados v, p, r e ϕ . Assim, negligencia-se o efeito de qualquer estado de um subsistema sobre o outro.

A principal hipótese que sustenta o desacoplamento é a de que a fuselagem é esbelta, ou seja, o comprimento é muito maior que a largura e a altura da aeronave. Supõe-se também que a velocidade longitudinal seja muito maior que as velocidades vertical e transversal.

As Eqs. (2.37), (2.38) e (2.39) descrevem a dinâmica longitudinal de uma aeronave (FOSSEN, 2011):

$$\begin{aligned} m[\dot{u} + Q_0 w + W_0 q + g \cos(\theta_0) \theta] &= \delta X, \\ m[\dot{w} - U_0 q - Q_0 u + g \sin(\theta_0) \cos(\phi_0) \theta] &= \delta Z, \\ I_y \dot{q} &= \delta M, \end{aligned} \quad (2.37)$$

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_y \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 & mQ_0 & mW_0 \\ -mQ_0 & 0 & -mU_0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} mg \cos(\theta_0) \\ mg \sin(\theta_0) \cos(\phi_0) \\ 0 \end{bmatrix} \theta = \begin{bmatrix} \delta X \\ \delta Z \\ \delta M \end{bmatrix}, \quad (2.38)$$

$$\dot{\theta} = q. \quad (2.39)$$

As Eqs. (2.40), (2.41) e (2.42) descrevem a dinâmica látero-direcional de uma aeronave, (FOSSEN, 2011):

$$\begin{aligned} m[\dot{v} + U_0 r - W_0 p - g \cos(\theta_0) \cos(\phi_0) \phi] &= \delta Y, \\ I_x \dot{p} - I_{xz} \dot{r} + (I_z - I_y)Q_0 r - I_{xz}Q_0 p &= \delta L, \end{aligned} \quad (2.40)$$

$$I_z \dot{r} - I_{xz} \dot{p} + (I_y - I_x)Q_0 p + I_{xz}Q_0 r = \delta N,$$

$$\begin{bmatrix} m & 0 & 0 \\ 0 & I_x & -I_{xz} \\ 0 & -I_{xz} & I_z \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -mW_0 & mU_0 \\ 0 & -I_{xz}Q_0 & (I_z - I_y)Q_0 \\ 0 & (I_y - I_x)Q_0 & I_{xz}Q_0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} -mg \cos(\theta_0) \cos(\phi_0) \\ 0 \\ 0 \end{bmatrix} \phi = \begin{bmatrix} \delta Y \\ \delta L \\ \delta N \end{bmatrix}, \quad (2.41)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta_0) \\ 0 & 1/\cos(\theta_0) \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix}. \quad (2.42)$$

As forças e momentos aerodinâmicos e propulsivos, Eq. (2.26), são gerados pela própria estrutura da aeronave, pelos flaps δ_F , pelos propulsores δ_T e pelas demais superfícies de comando descritas na Eq. (2.43) e ilustradas na Fig. 2.17.

$$\begin{bmatrix} \delta_T \\ \delta_E \\ \delta_F \\ \delta_A \\ \delta_R \end{bmatrix} = \begin{bmatrix} \text{Força de propulsão} \\ \text{Deflexão do profundor} \\ \text{Deflexão dos flaps} \\ \text{Deflexão dos ailerons} \\ \text{Deflexão do leme} \end{bmatrix}. \quad (2.43)$$

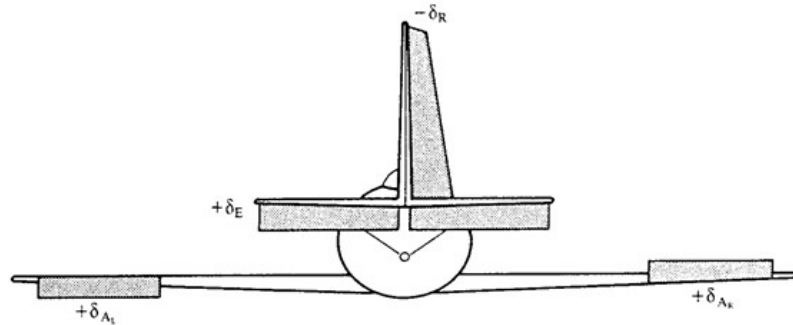


Figura 2.17: Ilustração das principais superfícies de comando (FOSSEN, 2011).

McLean (1990) expressa as forças e momentos longitudinais e látero-direcionais em função dos propulsores e das superfícies de comando de acordo respectivamente com as Eqs. (2.44) e (2.45):

$$\begin{bmatrix} \delta X \\ \delta Z \\ \delta M \end{bmatrix} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{w}} & X_{\dot{q}} \\ Z_{\dot{u}} & Z_{\dot{w}} & Z_{\dot{q}} \\ M_{\dot{u}} & M_{\dot{w}} & M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_w & X_q \\ Z_u & Z_w & Z_q \\ M_u & M_w & M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} X_{\delta_T} & X_{\delta_E} & X_{\delta_F} \\ Z_{\delta_T} & Z_{\delta_E} & Z_{\delta_F} \\ M_{\delta_T} & M_{\delta_E} & M_{\delta_F} \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_E \\ \delta_F \end{bmatrix}, \quad (2.44)$$

$$\begin{bmatrix} \delta Y \\ \delta L \\ \delta N \end{bmatrix} = \begin{bmatrix} Y_{\dot{v}} & Y_{\dot{p}} & Y_{\dot{r}} \\ L_{\dot{v}} & L_{\dot{p}} & L_{\dot{r}} \\ N_{\dot{v}} & N_{\dot{p}} & N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} Y_v & Y_p & Y_r \\ L_v & L_p & L_r \\ N_v & N_p & N_r \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_A} & Y_{\delta_R} \\ L_{\delta_A} & L_{\delta_R} \\ N_{\delta_A} & N_{\delta_R} \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix}. \quad (2.45)$$

Os parâmetros $X_{\dot{u}}$, $X_{\dot{w}}$, ..., N_{δ_R} são relativos respectivamente às derivadas parciais $\frac{\partial X}{\partial \dot{u}}$, $\frac{\partial X}{\partial \dot{w}}$, ..., $\frac{\partial N}{\partial \delta_R}$ e são denominados de coeficientes aerodinâmicos, os quais são função da aeronave em estudo e da sua condição de operação. Roskam (2001) apresenta em seu livro uma tabela contendo os coeficientes aerodinâmicos das principais aeronaves comerciais em diversas condições de operação. Ressalta-se que se as dinâmicas dos atuadores forem

importantes para uma maior exatidão do modelo resultante da aeronave, coeficientes aerodinâmicos como X_{δ_T} e Y_{δ_A} devem ser incluídos no processo de modelagem.

Por fim, ao se levar em conta as relações matemáticas apresentadas nas Eqs. (2.38) e (2.39), e os efeitos dos atuadores sobre o sistema aeronáutico, Eq. (2.44), é possível obter um sistema de equações que descreve o comportamento dinâmico longitudinal de uma aeronave submetida a pequenas perturbações, Eqs. (2.46) e (2.47).

$$\begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & -X_{\dot{q}} \\ -Z_{\dot{u}} & m - Z_{\dot{w}} & -Z_{\dot{q}} \\ -M_{\dot{u}} & -M_{\dot{w}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -X_u & mQ_0 - X_w & mW_0 - X_q \\ -mQ_0 - Z_u & -Z_w & -mU_0 - Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} mg \cos(\theta_0) \\ mg \sin(\theta_0) \cos(\phi_0) \\ 0 \end{bmatrix} \theta = \begin{bmatrix} X_{\delta_T} & X_{\delta_E} & X_{\delta_F} \\ Z_{\delta_T} & Z_{\delta_E} & Z_{\delta_F} \\ M_{\delta_T} & M_{\delta_E} & M_{\delta_F} \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_E \\ \delta_F \end{bmatrix}. \quad (2.46)$$

$$\dot{\theta} = q. \quad (2.47)$$

Analogamente, ao se levar em conta as relações matemáticas apresentadas nas Eqs. (2.41) e (2.42), e os efeitos dos atuadores sobre o sistema aeronáutico, Eq. (2.45), é possível obter um sistema de equações que descreve o comportamento dinâmico látero-direcional de uma aeronave submetida a pequenas perturbações, Eqs. (2.48) e (2.49).

$$\begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{p}} & -Y_{\dot{r}} \\ -L_{\dot{v}} & I_x - L_{\dot{p}} & -I_{xz} - L_{\dot{r}} \\ -N_{\dot{v}} & -I_{xz} - N_{\dot{p}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & -mW_0 - Y_p & mU_0 - Y_r \\ -L_v & -I_{xz}Q_0 - L_p & (I_z - I_y)Q_0 - L_r \\ -N_v & (I_y - I_x)Q_0 - N_p & I_{xz}Q_0 - N_r \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} -mg \cos(\theta_0) \cos(\phi_0) \\ 0 \\ 0 \end{bmatrix} \phi = \begin{bmatrix} Y_{\delta_A} & Y_{\delta_R} \\ L_{\delta_A} & L_{\delta_R} \\ N_{\delta_A} & N_{\delta_R} \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix}. \quad (2.48)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta_0) \\ 0 & 1/\cos(\theta_0) \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix}. \quad (2.49)$$

Após a apresentação da modelagem dinâmica linear de uma aeronave de asa fixa, no próximo capítulo será exibido o estado da arte no que tange à modelagem e sobretudo ao controle de aeronaves, simuladores, e operações críticas de voo.

CAPÍTULO III

REVISÃO DE LITERATURA

A revisão de literatura deste trabalho consiste em apresentar o estado da arte no que se refere ao estudo de temáticas relacionadas ao controle de aeronaves, sobretudo quanto à utilização de técnicas de inteligência computacional para tal. Este capítulo é dividido em perda de controle da aeronave durante o voo, simuladores de voo e inteligência computacional aplicada a sistemas aeronáuticos.

3.1. Perda de Controle da Aeronave Durante o Voo

A perda de controle da aeronave durante o voo, também conhecida como LOC-I (*Loss of Control In-flight*) agrega um conjunto de acidentes em que a tripulação não consegue manter o controle da aeronave em voo, resultando em um desvio de rota irreversível (IATA, 2015). É um tema de grande relevância no setor aeronáutico devido ao fato de ser causa de boa parte dos acidentes aéreos fatais, e dessa forma é tema de estudo das comunidades técnicas e científicas do setor aeronáutico.

A Tab. 3.1 exhibe trabalhos que exploram a temática, e em seguida são discutidos os principais desenvolvimentos, informações e as conclusões relevantes para o desenvolvimento desta tese.

Tabela 3.1: Trabalhos que Exploram a Perda de Controle da Aeronave Durante o Voo.

Trabalho	Assunto
IATA (2020)	Estatísticas – Acidentes aéreos de 2015 a 2019
Oliveira (2018)	Requisitos para o treinamento <i>full-stall</i>
Frink <i>et al.</i> (2017)	Modelagem de LOC-I com ênfase no fenômeno de <i>stall</i>
IATA (2015)	Estatísticas – Acidentes aéreos de 2010 a 2014
Cox, Cunningham e Jordan (2012)	Controle de aeronaves em escala reduzida – LOC-I
Murch, Cox e Cunningham (2009)	Controle de Aeronaves: considerações de software
Jordan e Bailey (2008)	Bancada e metodologia da NASA para o controle de Veículos Aéreos Não Tripulados (VANTs) submetidos a LOC-I
Jordan <i>et al.</i> (2006)	Apresentação da bancada da NASA de estudo de VANTs submetidos a LOC-I
Wu, Zhang e Zhou (2000)	Filtro de Kalman: Efetividade do sistema de controle de uma aeronave

IATA (2020) desenvolve um relatório a respeito dos acidentes aéreos ocorridos de 2015 a 2019 em aeronaves comerciais com massa de decolagem máxima superior a 5700 kg. Neste relatório é indicado que, embora apenas 8% dos acidentes aéreos têm como causa a LOC-I, Fig. 3.1, aproximadamente 70% das fatalidades (780 mortes) foram causadas por acidentes classificados dentro desta categoria. As categorias de acidentes são apresentadas na Tab. 3.2.

Tabela 3.2: Categorias de Acidentes e as Descrições de Suas Terminologias Padrão.

Categoria de Acidente	Descrição/Tradução
<i>Runway/Taxiway Excursion</i>	Saída inadequada da aeronave da pista de pouso
<i>In-flight Damage</i>	Danos durante o voo
<i>Hard Landing</i>	Aterrissagem forçada
<i>Gear-up Landing/ Gear Collapse</i>	Pouso sem o trem de pouso totalmente estendido
<i>Ground Damage</i>	Incidentes em que o pessoal da companhia aérea causa danos a uma aeronave no solo
<i>Loss of Control In-flight</i>	Perda de controle da aeronave durante o voo
<i>Tailstrike</i>	Colisão da extremidade traseira de uma aeronave na pista
<i>Undershoot</i>	Pouso antes ou na frente do alvo pretendido
<i>Other</i>	Outras causas de acidentes
<i>Collision</i>	Colisão
<i>Controlled Flight into Terrain</i>	Deficiência nos sistemas de medição da aeronave

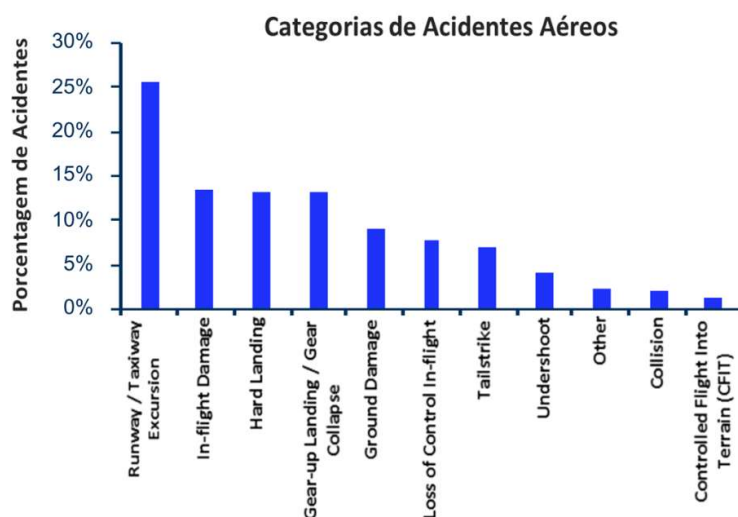


Figura 3.1: Percentual de acidentes aéreos em função das categorias de acidentes apresentadas com a nomenclatura padrão (IATA, 2020).

A Tab. 3.3 sintetiza as 3 principais categorias de acidentes aéreos no que diz respeito à letalidade, sendo *Controlled Flight into Terrain* (CFIT) os acidentes em que a aeronave preserva toda sua controlabilidade, mas que por motivos diversos os sistemas de medição e a tripulação não são capazes de detectar uma divergência de rota, e *Runway/Taxiway Excursion*, categoria mais recorrente de acidente aéreo, é a falha de sistema ou colisão da aeronave em solo.

Tabela 3.3: Principais Categorias de Acidentes Aéreos Quanto à Letalidade (IATA, 2020).

Categoria de Acidente	Número de Acidentes	Acidentes Fatais	Fatalidades	Fatalidades (%)
<i>Loss of Control In-Flight (LOC-I)</i>	22	19	780	70%
<i>Controlled Flight into Terrain (CFIT)</i>	4	3	124	11%
<i>Runway/Taxiway Excursion</i>	74	4	55	5%
<i>Total</i>	292	37	1116	100%

IATA (2020) faz uma série de levantamentos de dados na tentativa de determinar a origem e os principais motivos que levam à LOC-I. Na Fig. 3.2 é apresentado um gráfico que indica a fase do voo em que as tripulações que sofreram o acidente relatam ter iniciado a perda de controle da aeronave. As especificações das siglas referentes às fases de voo apresentadas na Fig. 3.2 estão descritas na Tab. 3.4.

Tabela 3.4: Fases de Voo.

Fase	Especificação
TOF	Decolagem (<i>Take-off</i>)
ICL	Subida inicial (<i>Initial Climb</i>)
ECL	Em rota de subida (<i>En Route Climb</i>)
CRZ	Cruzeiro (<i>Cruise</i>)
DST	Descida (<i>Descent</i>)
APR	Aproximação (<i>Approach</i>)
GOA	Arremetida (<i>Go-around</i>)
LND	Aterrisagem (<i>Landing</i>)

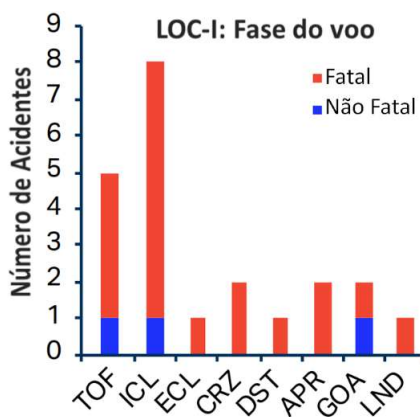


Figura 3.2: Fase do voo em que houve a ocorrência de LOC-I (IATA, 2020).

Ao se considerar a Fig. 3.2 e também o tempo de duração de cada uma das fases, verifica-se que é em regime transiente que mais ocorre este fenômeno, como por exemplo na decolagem e na subida inicial.

Os fatores mais recorrentes quando há a perda de controle da aeronave durante o voo são elencados na Tab. 3.5. Vale ressaltar que mais de um fator costuma ocorrer simultaneamente.

Tabela 3.5: Fatores Recorrentes em Acidentes Devido à LOC-I (IATA, 2020).

Classificação	Fator	Ocorrência (%)
Deficiências	Operações de voo (falhas humanas)	55
	Gerenciamento de segurança	50
	Sistemas de treinamento	40
	Procedimentos operacionais padrão (POPs) e verificação	40
	Supervisão regulatória	40
Erros da tripulação	Operação manual / Controles de voo	50
	Aderência dos procedimentos aos POPs	50
	Comunicação entre pilotos	35
	Falha de <i>checklist</i>	25
	Falhas gerais de comunicação	20
Ameaças ambientais	Condições meteorológicas severas	45
	Pouca visibilidade/ condições meteorológicas de voo por instrumentos	20
	Rajadas de vento	15
	Condições de congelamento	15
	Tempestade de raios	5
Estados indesejados da aeronave	Operação fora das limitações da aeronave	40
	Velocidade vertical/lateral fora do limite operacional da aeronave	35
	Abrupto controle da aeronave	30
	Controle de voo / Automação	20
	Penetração desnecessária em condições meteorológicas adversas	15

Ameaças durante o voo	Mau funcionamento da aeronave	35
	Mau funcionamento / falha do sistema propulsor	20
	Fadiga	15
	Pressão operacional	15
Fatores não-mensuráveis	Performance da tripulação	50
	Monitoramento / Verificação	50
	Liderança	40
	Liderança que deve ser demonstrada pelo capitão	40
	Gerenciamento de eventualidades	30

Ao se analisar cada um desses fatores seguidos por suas taxas de ocorrência, merecem atenção: condições meteorológicas severas (45%), operação fora das limitações da aeronave (40%), velocidade vertical/lateral fora do limite operacional (35%), mau funcionamento da aeronave (35%) e mau funcionamento/falha do sistema propulsor (20%). Esses fatores são alvos de estudo pelas comunidades técnicas e científicas, e também são objetos de estudo desta tese no que diz respeito ao desenvolvimento de sistemas de controle que possam minimizar os efeitos dos mesmos sobre a controlabilidade de um sistema aeronáutico.

Cox, Cunningham e Jordan (2012), trabalho realizado pela NASA, lança mão de aeronaves em escala reduzida para a validação de modelos dinâmicos, controle e tecnologias voltadas a condições de voo de alto risco e LOC-I. A agência espacial desenvolveu uma plataforma de testes, denominada AirSTAR, Fig. 3.3, que de 2009 a 2011 viabilizou a realização de 58 voos de teste.



Figura 3.3: Bancada de testes AirSTAR (COX, CUNNINGHAM e JORDAN, 2012).

Os objetivos principais de Cox, Cunningham e Jordan (2012) são a identificação e modelagem aerodinâmica da aeronave, e a realização de testes de algoritmos adaptativos no intuito de verificar a robustez dos mesmos em condições críticas de voo. Os autores indicam como limitações da pesquisa a necessidade de alcance visual da aeronave, visto que não se dispunha de pilotagem automática, e também a necessidade de construção de um protótipo maior e com cauda em T, configuração crítica em voos com condições meteorológicas adversas.

Frink *et al.* (2017) trabalham a temática da LOC-I, porém com ênfase em um fenômeno denominado *stall*, a perda abrupta da força de sustentação de partes ou de toda a aeronave durante o voo. É um tema de bastante relevância devido ao potencial de causar acidentes graves, visto que o piloto perde completamente o controle do sistema, e além de ser um fenômeno de difícil reprodução, devido à alta complexidade matemática e da dificuldade de se reproduzir experimentos com razoável grau de segurança.

O trabalho de Frink *et al.* (2017) compara o resultado de modelos representativos, criados a partir de dados aerodinâmicos computacionais e de túnel de vento; com modelos específicos, produzidos a partir de dados de *stall* em voo. O estudo conclui a semelhança matemática entre os modelos representativos e específicos, o que sugere que análises numéricas e ensaios com túnel de vento podem servir efetivamente como ferramentas de estudo para o fenômeno de *stall*.

Através do túnel de vento e do protótipo ilustrados na Fig. 3.4, o trabalho levanta a curva do coeficiente de sustentação da aeronave em função do ângulo de incidência do vento com relação à superfície da aeronave, denominado ângulo de ataque α , sendo este variado de 0° até 90° . Além disso simulam-se diferentes intensidades e ângulos de rajadas de vento laterais, e também o efeito de falhas na estrutura aeronáutica no coeficiente de sustentação da mesma. Todas as informações subsidiam uma maior compreensão e a previsão da ocorrência do *stall* no protótipo ensaiado.



Figura 3.4: Túnel de vento e protótipo utilizados em Frink *et al.* (2017).

Oliveira (2018) aponta em seu trabalho o protocolo internacional (FAA-H-8083- 3B, 2016) que dispõe sobre os procedimentos de recuperação de *stall*, resumidamente e em ordem cronológica expostos a seguir:

- Desabilitar o *auto-pilot* e o *auto-throttle*;
- Aplicar tanta ação de profundor quanto o necessário para abaixar o nariz da aeronave;
- Ajustar o ângulo de rolagem;
- Ajustar *throttle* da aeronave assim que o ângulo de ataque estiver dentro dos limites operacionais;
- Recolher speed brakes;
- Após retomado o controle da aeronave, retornar ao percurso desejado.

Diante do protocolo e da tentativa mundial de se reduzir o número de acidentes devido à LOC-I, Oliveira (2018) propõe uma investigação, com 20 pilotos de teste da empresa EMBRAER, a respeito da periculosidade na realização de testes de voo que envolvam a indução do fenômeno de *stall*, e a efetividade do treinamento de recuperação de *stall* em simuladores de voo. A grande maioria dos pilotos considera que o risco que envolve um ensaio de voo é inaceitável, envolvendo a potencial perda da tripulação e do protótipo; além disso, embora uma discreta maioria dos pilotos considera que ensaios virtuais de *stall* seriam uma opção válida de treinamento, boa parte dos entrevistados não acredita na eficácia da proposição, alegando principalmente divergências entre a simulação e a vida real.

Apesar das desconfiças com relação à eficácia de simuladores de voo, estes são cada vez mais utilizados no projeto de aeronaves e no treinamento de pilotos, dessa forma, este assunto é abordado a seguir.

3.2. Simuladores de Voo

A Tab. 3.6 exibe trabalhos que exploram a temática, e em seguida são discutidos os principais desenvolvimentos, informações e as conclusões relevantes para o desenvolvimento desta tese.

Tabela 3.6: Trabalhos que Exploram a Utilização de Simuladores de Voo.

Trabalho	Assunto
Zabidin, Pairan e Shamsudin (2020)	Estudo do desempenho de uma arquitetura de controle de VANT baseada em PIDs em cascata, a partir de uma simulação do tipo <i>software-in-the-loop</i> realizada entre o LabView e o X-Plane.
Bittar <i>et al.</i> (2014)	X-Plane, <i>software-in-the-loop</i> e navegação de VANT.
Bittar, Oliveira e Figueredo (2014)	X-Plane, <i>hardware-in-the-loop</i> e controle de VANT em operações críticas de voo.
Bittar e Oliveira (2013)	X-Plane e controle de VANT.
Garcia e Barnes (2010)	X-Plane e simulação da operação de múltiplos helicópteros.
Ribeiro e Oliveira (2010)	X-Plane e <i>software-in-the-loop</i> .
Craighead <i>et al.</i> (2007)	Apresentação dos simuladores comerciais e <i>open source</i>
Adiprawita, Ahmad e Sembiring (2008)	Teste de voo automático e identificação dinâmica de sistemas para helicópteros não tripulados a partir do método da resposta em frequência.

Em Bittar *et al.* (2014), pesquisa realizada no Instituto Tecnológico de Aeronáutica (ITA), utiliza-se o software X-Plane para o estudo do controle e da navegação de VANT. O trabalho indica que o X-Plane é um simulador de voo certificado pela agência de aviação americana – FAA (*Federal Aviation Administration*) para o treinamento de pilotos e que controladores testados neste software foram bem-sucedidos em aeronaves reais.

O X-Plane utiliza como ferramenta matemática a teoria de elementos de pá (Bittar *et al.*, 2014), que se baseia na divisão da aeronave em pequenos elementos, nos quais são calculados os esforços envolvidos, e por último a soma dos efeitos de cada um destes elementos no sistema aeronáutico. Neste simulador aeronáutico é possível trabalhar tanto com aeronaves comerciais, quanto com VANTs, Fig. 3.5, que podem ser modelados geometricamente e exportados ao X-Plane a partir do software Plane-Maker.



Figura 3.5: O X-Plane permite simular a operação de aeronaves comerciais e de aeromodelos (adaptada de BITTAR *et al.*, 2014).

Bittar *et al.* (2014) desenvolvem uma interface de comunicação entre o Simulink e o X-Plane, em que $P1$, $P2$, $P3$ e $P4$ são as portas de comunicação, Fig. 3.6. Esta interface permite a utilização do simulador de voo como a planta do sistema a ser controlado, e o MATLAB como plataforma de desenvolvimento do controlador aeronáutico, assim, viabilizando a realização do *software-in-the-loop*.

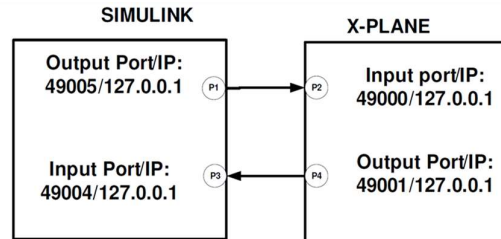


Figura 3.6: Interface de comunicação entre o Simulink e o X-Plane (BITTAR *et al.*, 2014).

A partir das Eqs. (3.1) e (3.2), é possível controlar o ângulo de guinada da aeronave e consequentemente gerenciar sua navegação, Fig. 3.7, sendo D_{graus} a distância da aeronave em graus com relação ao ponto de destino, γ o ângulo formado entre a aeronave e o ponto de destino, medido em sentido horário a partir de um vetor com sentido norte, Lat_{wp} e Lon_{wp} são as coordenadas geográficas do ponto de destino, e Lat_{VANT} e Lon_{VANT} são as coordenadas geográficas da aeronave (Bittar *et al.*, 2014).

$$D_{\text{graus}} = \sqrt{(Lon_{wp} - Lon_{VANT})^2 + (Lat_{wp} - Lat_{VANT})^2} \quad (3.1)$$

$$\gamma = \tan^{-1} \left[\frac{Lon_{wp} - Lon_{VANT}}{Lat_{wp} - Lat_{VANT}} \right] \quad (3.2)$$

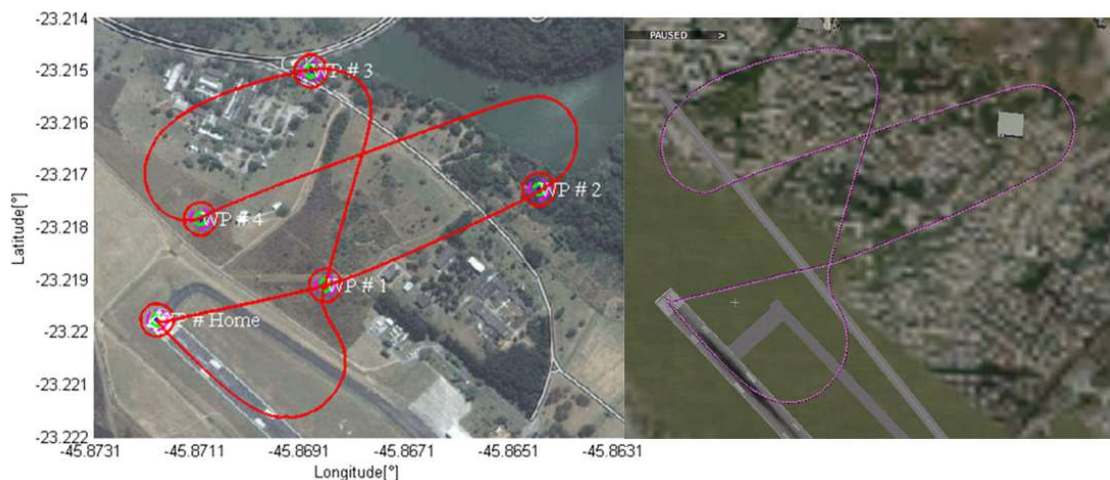


Figura 3.7: Navegação da aeronave (BITTAR *et al.*, 2014).

Em Bittar e Oliveira (2013), utiliza-se a interface de comunicação apresentada na Fig. 3.6, mas alterada de modo a viabilizar a execução de um *hardware-in-the-loop*, no qual em um computador é executado o X-Plane, e em outro opera o sistema de controle da aeronave simulada. Este procedimento separa fisicamente a planta e o sistema de controle, e assim aproxima mais a simulação de uma situação real.

Bittar, Oliveira e Figueredo (2014) compilam todos os desenvolvimentos anteriormente expressos e apresentam como fator de inovação a utilização de controlador Proporcional Integral Derivativo (PID) em cascata no controle de uma aeronave submetida a condições críticas de voo. As condições de voo são simuladas no software X-Plane, Fig. 3.8, em que é possível selecionar o tempo (ensolarado, chuvoso, tempestuoso), a visibilidade do voo, a velocidade do vento, a velocidade e ângulo de cisalhamento do vento, o nível de turbulência, a falha de componentes da aeronave, dentre outras configurações.

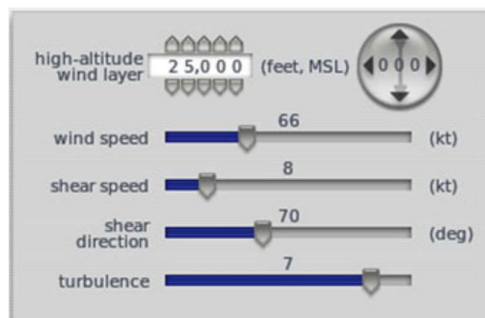


Figura 3.8: Seletor de algumas das configurações de voo do software X-Plane (BITTAR, OLIVEIRA e FIGUEREDO, 2014).

Finalizando a exposição de trabalhos que exploram a utilização de simuladores de voo, Zabidin, Pairan e Shamsudin (2020) propõem a modelagem e controle de um VANT de asa rotativa a partir da utilização dos softwares LabVIEW e X-Plane 9. Neste software é realizada toda a modelagem geométrica e matemática do VANT, Fig. 3.9, enquanto que naquele é implementado o controlador proporcional integral derivativo (PID) em cascata a ser validado a partir da realização do *software-in-the-loop* (SITL). A partir dos gráficos apresentados na Fig. 3.10, extraídos do X-Plane 9 durante o SITL, evidencia-se a capacidade do controlador PID em cascata de controlar satisfatoriamente os ângulos de arfagem θ e rolagem ϕ do VANT.

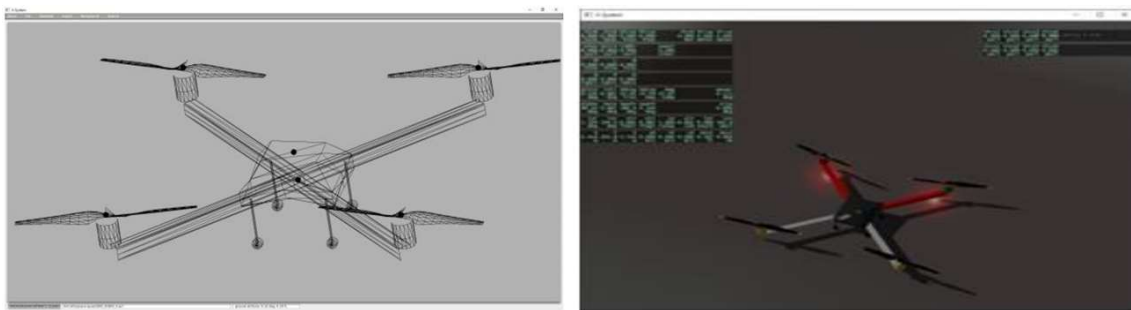


Figura 3.9: Modelagem geométrica e dinâmica realizada no X-Plane 9 (ZABIDIN, PAIRAN e SHAMSUDIN, 2020).

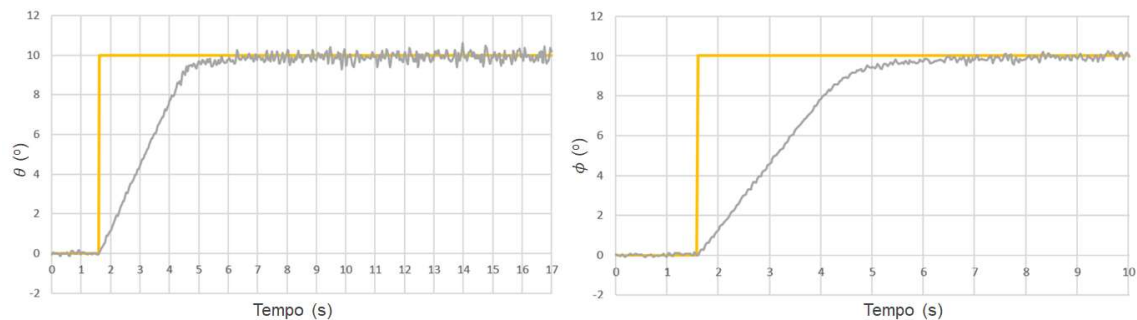


Figura 3.10: Desempenho do controlador PID em cascata no SITL, sendo o valor de referência em amarelo, e a saída do sistema em preto (ZABIDIN, PAIRAN e SHAMSUDIN, 2020).

Após a explanação de pesquisas que exploram a temática de simuladores de voo, em sequência são apresentados trabalhos que exploram a temática da inteligência computacional, sobretudo aplicada a sistemas aeronáuticos.

3.3. Inteligência Computacional Aplicada a Sistemas Aeronáuticos

A inteligência computacional é um conjunto de técnicas que se baseia na observação de como os seres vivos, e em especial os seres humanos, comportam-se solucionando diferentes problemas. É um conjunto de técnicas que, devido à alta capacidade de adaptação e versatilidade, além da viabilidade tecnológica proporcionada pela nova geração de computadores e plataformas de desenvolvimento, são aplicadas em diferentes áreas do conhecimento e em uma diversidade de aplicações.

Dentre as áreas do conhecimento em que a inteligência computacional ganha espaço, merece destaque o controle de sistemas. O conjunto de técnicas da inteligência computacional, utilizado para garantir que um sistema opere dentro das especificações pré-determinadas, é denominado controle inteligente. A Tab. 3.7 exhibe trabalhos que exploram a temática em diversas aplicações.

Tabela 3.7: Inteligência Computacional Aplicada ao Controle de Sistemas.

Trabalho	Assunto
Li, Wang, Wu, Lam e Gao (2018)	Controlador <i>sliding-mode</i> : Aplicação no controle de um sistema com tempo de atraso e incertezas, modelado a partir de um sistema fuzzy do tipo 2.
Zhao, Wang, Zhang, Liu e Yang (2018)	Rede neural artificial: Controle de vibração de corda com zona morta do atuador.
Xu e Sun (2018)	ANFIS: Controle de sistemas realimentados na presença de distúrbios.
Zhou, Wu e Shi (2017)	ANFIS: Controle de sistemas com tempo de atraso e saturação do atuador.
Zhou, Li, Wu, Wang e Ahn (2017)	ANFIS: Controle de sistemas com dinâmica não-modelada e saturação do atuador.
Wang, Sun e Liu (2017)	Rede neural artificial: Controle de sistemas com tempo de atraso e incerteza no modelo dinâmico.
Cui, Yang, Li e Sharma (2017)	Rede neural artificial: Controle de submarino.
Chen, Zhang e Liu (2016)	Controlador fuzzy: Piloto automático veicular.

Há também na literatura trabalhos que lançam mão da inteligência computacional especificamente para a modelagem e controle de aeronaves. A Tab. 3.8 exhibe trabalhos que exploram a temática, e em seguida são discutidos os principais desenvolvimentos, informações e as conclusões relevantes para o desenvolvimento desta tese.

Tabela 3.8: Inteligência Computacional Aplicada à Modelagem e Controle de Aeronaves.

Trabalho	Assunto
Ulus e Eski (2021)	Análise comparativa do desempenho de diferentes arquiteturas de controle fuzzy híbridas aplicadas, em simulação, ao controle de atitude de um VANT
Cui e Zhang (2021)	Simulação do tipo <i>hardware-in-the-loop</i> na análise do desempenho de um controlador fuzzy aplicado a um veículo aéreo não tripulado comandado a partir de um motor <i>turbojet</i>
Li (2020)	Controlador PID-fuzzy aplicado ao controle de um veículo aéreo não tripulado
Raboaca, Dumitrescu e Manta (2020)	Controle de trajetória de aeronaves através de equipamento de radar a partir da utilização de um algoritmo baseado em lógica fuzzy
Radhakrishnan e Swarup (2020)	Comparação entre a performance de um controlador PID-fuzzy com demais técnicas de controle aplicadas ao ângulo de arfagem (θ) de uma aeronave
Scott e González (2020)	Desenvolvimento de um controlador fuzzy aplicado a aeronave sem a utilização de um modelo dinâmico do sistema
Wang <i>et al.</i> (2020)	Controle preditivo da relação ar/combustível nos motores de veículos aéreos não tripulados a partir de um sistema neuro-fuzzy
Hu, Xu e Hu (2018)	Desenvolvimento de um controlador fuzzy adaptativo para um veículo aéreo hipersônico
Chen, Shi e Lim (2016)	Rede neural artificial: Controle de helicóptero submetido a falha nos atuadores e incerteza do processo.

Jha, Gaur e Yadav (2016)	Controle de aeronaves: Diferentes técnicas de inteligência computacional.
Xu, Zhang e Pan (2016)	Rede neural artificial: Controle de aeronave hipersônica.
Brandon e Morelli (2012)	ANFIS: Modelagem aerodinâmica de aeronave militar.
Thums, Torres e Palhares (2012)	PID em cascata otimizado via algoritmo genético: Controle longitudinal de aeronave.
Kurnaz, Cetin e Kaynak (2010)	ANFIS: Aplicação em simulador de voo.
Liu, Naadimuthu e Lee (2008)	ANFIS: Controle de pouso.
Chen e Cheng (1998)	H-infinito e algoritmo genético: Controle longitudinal de aeronave.
Kwong <i>et al.</i> (1995)	ANFIS: Controle de aeronave submetida a falha de componentes.

Kurnaz, Cetin e Kaynak (2010) utilizam o ANFIS, otimizado através do método da máxima descida, para o controle látero-direcional (controle do ângulo de rolagem) e longitudinal (controle de altitude e velocidade) de um VANT denominado aerosonda. Este trabalho lança mão do simulador de voo *open source* Flight Gear, Fig. 3.11, e do software MATLAB respectivamente para a modelagem e controle da aeronave.

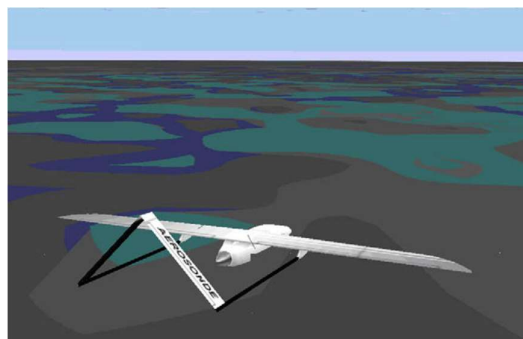


Figura 3.11: Flight Gear simulando a operação da aeronave aerosonda (KURNAZ, CETIN e KAYNAK, 2010).

Thums, Torres e Palhares (2012) propõem um controlador longitudinal de VANTs com a mesma arquitetura de controle do sistema comercial Micropilot, Fig. 3.12. O fator de inovação está na metodologia de sintonização do sistema de controle: inicialmente, aplica-se um algoritmo genético na obtenção dos parâmetros da malha interna, delimitada pelas linhas pontilhadas; em seguida, o mesmo método de otimização é usado na sintonização dos parâmetros do PID externo e do multiplicador K_F . Esta metodologia mostra-se capaz de satisfatoriamente controlar a altitude h , o ângulo de guinada θ e a velocidade V_T de um veículo aéreo não tripulado a partir da atuação do profundor u_{elev} e do motor u_{thr} .

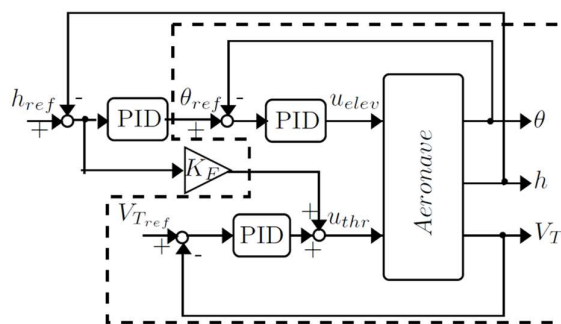


Figura 3.12: Arquitetura de controle desenvolvida pela empresa Micropilot (THUMS, TORRES e PALHARES, 2012).

Brandon e Morelli (2012) desenvolvem uma técnica de modelagem ANFIS da dinâmica de um supersônico. Uma aeronave desta categoria é posta em operação, enquanto são extraídos dados experimentais em voo, os quais são submetidos a um processo de ajuste a partir de um método de otimização. Este processo de busca da solução passa pela otimização da base de regras, da quantidade e dos parâmetros das funções de pertinência. Os resultados do trabalho indicam a capacidade do ANFIS no ajuste aos dados experimentais da aeronave, tendo obtido taxas de ajuste aos dados superiores a 95%.

Jha, Gaur e Yadav (2016) propõem a aplicação de diferentes técnicas para o controle do ângulo de arfagem de uma aeronave, a saber, controle PID convencional, controle auto-organizador Proporcional Integral (PI) mais Proporcional Derivativo (PD), e controle fuzzy PD mais controle fuzzy PI. Os resultados indicam que o primeiro apresenta melhor solução de compromisso entre tempo de resposta, oscilação e tempo de acomodação; o segundo apresenta o melhor tempo de resposta, porém alto nível de oscilação; e o terceiro apresenta oscilação próxima de zero, porém elevado tempo de resposta.

Xu, Zhang e Pan (2016) aplicam ao controle longitudinal de um veículo aéreo hipersônico uma rede neural artificial, denominada RNA com funções de ativação de base radial, que se diferencia das redes neurais clássicas com múltiplas camadas pela presença de apenas uma camada oculta de neurônios, e a utilização de gaussianas como as funções de ativação (HAYKIN, 2008). Este tipo de RNA possui a vantagem de apresentar uma estrutura matemática mais simples, o que facilita o processo de otimização do controlador, e viabiliza a análise de estabilidade analiticamente através de manipulações matemáticas da função de Lyapunov adotada para o problema. Xu, Zhang e Pan (2016) provam a estabilidade do sistema, e apresentam em simulação que o veículo aéreo hipersônico controlado através da RNA consegue seguir os valores de referência pré-estipulados.

Chen, Shi e Lim (2016) realizam o controle da bancada apresentada na Fig. 3.13, sendo o sistema sujeito a incertezas, a distúrbios variantes no tempo, e a falhas nos atuadores. O trabalho lança mão da RNA com funções de ativação de base radial, e dessa forma também é provada analiticamente a estabilidade do sistema controlado. A bancada controlada através da RNA consegue satisfatoriamente seguir os valores de referência dos ângulos de arfagem e guinada, indicando novamente a capacidade da inteligência computacional em ser utilizada na resolução de problemas de controle.



Figura 3.13: Bancada utilizada para o estudo de sistemas aeronáuticos de asa rotativa (CHEN, SHI e LIM, 2016).

Hu, Xu e Hu (2018) propõem a teoria dos conjuntos fuzzy no controle de um veículo aéreo hipersônico durante a fase de cruzeiro do voo. Utiliza-se para tal o método de inferência fuzzy de Takagi-Sugeno no desenvolvimento da arquitetura de controle, que associado a uma manipulação matemática que parte da definição da função de Lyapunov, viabiliza a verificação da robustez e estabilidade do sistema.

No intuito de comparar o desempenho do controlador que lança mão da lógica fuzzy, desenvolve-se neste trabalho um sistema de controle ótimo, mas que despreza as incertezas e as não-linearidades intrínsecas à dinâmica do processo. Na Fig. 3.14 são apresentados os gráficos dos erros de altitude e de velocidade da aeronave ao se utilizar, em simulação, os controladores fuzzy (em preto) e ótimo (em azul). Verifica-se que o controlador fuzzy obteve os valores de erro mais próximos a zero.

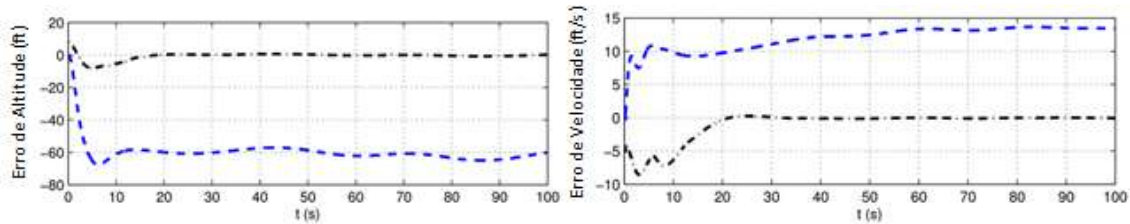


Figura 3.14: Performance dos controles (HU, XU e HU, 2018).

A equipe da NASA, no centro de pesquisa de Langley, desenvolvem a aplicação da teoria dos conjuntos fuzzy no controle de aeronaves. Scott e González (2020) lançam mão de um controlador genérico que opera a partir do método de inferência fuzzy de Takagi-Sugeno, que consegue de forma satisfatória controlar uma aeronave enquanto são extraídos dados de voo da mesma. Esses dados são processados pelo método RTGM (*Real Time Global Modeling*), que obtém um modelo específico para aeronave, o qual é utilizado na concepção de um controlador PD para o sistema.

A bancada utilizada para os testes de desempenho do sistema de controle consiste em um túnel de vento de 12 pés, em que um protótipo de aeronave é fixado em um dispositivo que permite apenas a arfagem e a rolagem da mesma.

Neste trabalho da NASA, compara-se inicialmente o desempenho dos SBRFs do tipo 1 e do tipo 2, cujas funções de pertinência, MF , são apresentadas respectivamente nas Figs. 3.15 e 3.16, no controle dos ângulos de arfagem e rolagem da aeronave, Fig. 3.17.

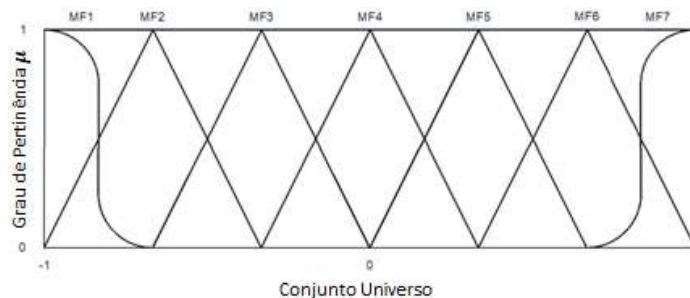


Figura 3.15: Funções de pertinência dos SBRFs do tipo 1 (SCOTT e GONZÁLEZ, 2020).

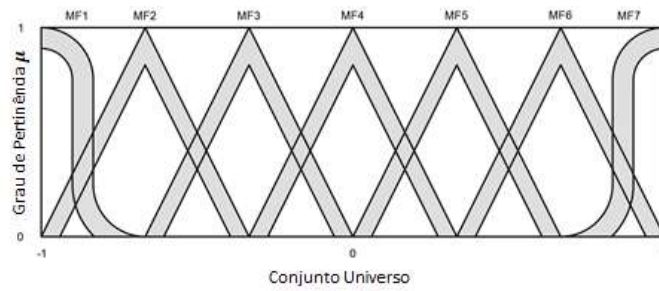


Figura 3.16: Funções de pertinência dos SBRFs do tipo 2 (SCOTT e GONZÁLEZ, 2020).

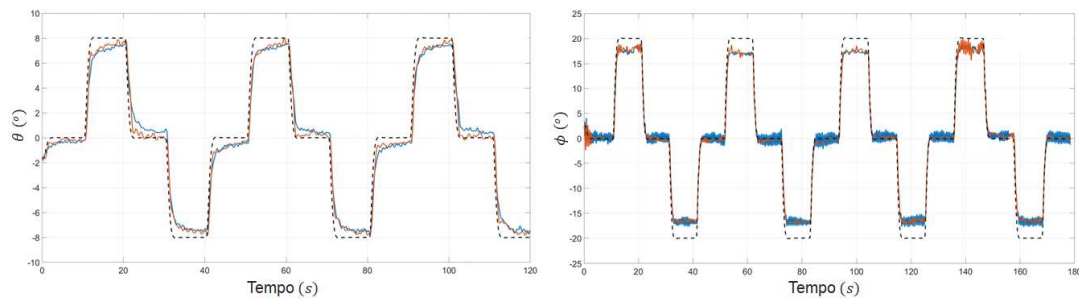


Figura 3.17: Desempenho dos SBRFs dos tipos 1, em azul, e 2, em vermelho, em relação ao sinal de referência, em preto, no controle dos ângulos de arfagem, à esquerda, e rolagem, à direita (SCOTT e GONZÁLEZ, 2020).

Os resultados do trabalho indicam a proximidade de desempenho dos SBRFs dos tipos 1 e 2, porém havendo ligeira superioridade deste último em se aproximar do sinal de referência. Na Fig. 3.18 é apresentado o desempenho dos controladores fuzzy, na faixa em azul, e PD, na porção final do gráfico. Verifica-se que o controlador fuzzy, embora notoriamente apresente desempenho inferior ao controlador PD, de fato obtém bons resultados, principalmente ao se considerar o fato de que o sistema de controle é genérico, ou seja, não há qualquer conhecimento das especificidades do sistema dinâmico nesta primeira fase do voo.

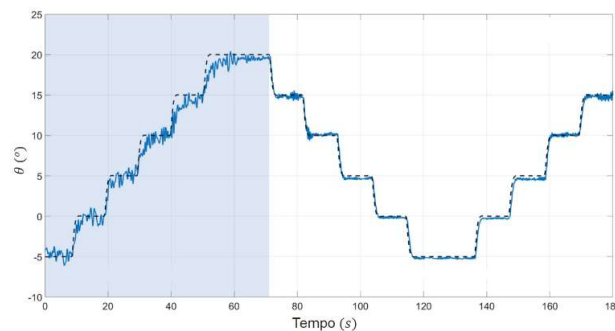


Figura 3.18: Desempenho dos controladores fuzzy e PD, gráfico azul, em relação ao sinal de referência, em preto, no controle do ângulo de arfagem (SCOTT e GONZÁLEZ, 2020).

Radhakrishnan e Swarup (2020) utilizam um sistema de controle PID fuzzy, cuja arquitetura é apresentada na Fig. 3.19, para controlar o sistema de arfagem de uma aeronave, em que $r(t)$ é o sinal de referência, $y(t)$ é a resposta do sistema, $e(t)$ é o sinal de erro, $u(t)$ é o sinal de controle, e K_p, K_i e K_d são os parâmetros do controlador PID, considerados neste trabalho como sendo parâmetros fuzzy.

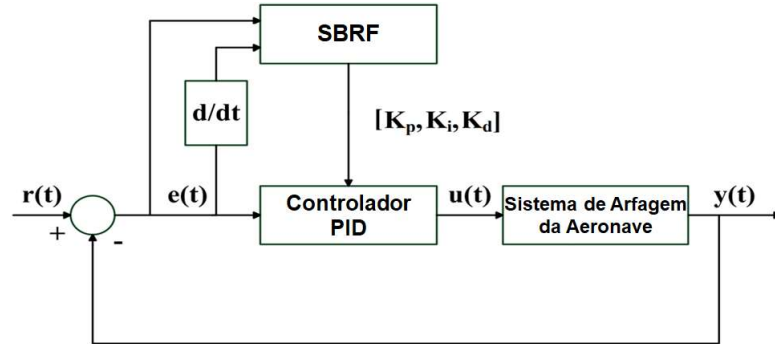


Figura 3.19: Arquitetura de controle PID fuzzy adotada em Radhakrishnan e Swarup (2020).

O sistema de controle desenvolvido no trabalho é comparado com os controladores PID tradicional, LQR (*Linear Quadratic Regulator*), LQR otimizado com AG (Algoritmo Genético) e com um controlador desenvolvido a partir da alocação de pólos e zeros, e os resultados são apresentados na Fig. 3.20 e na Tab. 3.9. É possível verificar, sobretudo na análise gráfica, a superioridade do controlador PID fuzzy frente às demais técnicas, fato também evidenciado pelos baixos valores de tempo de subida e de acomodação.

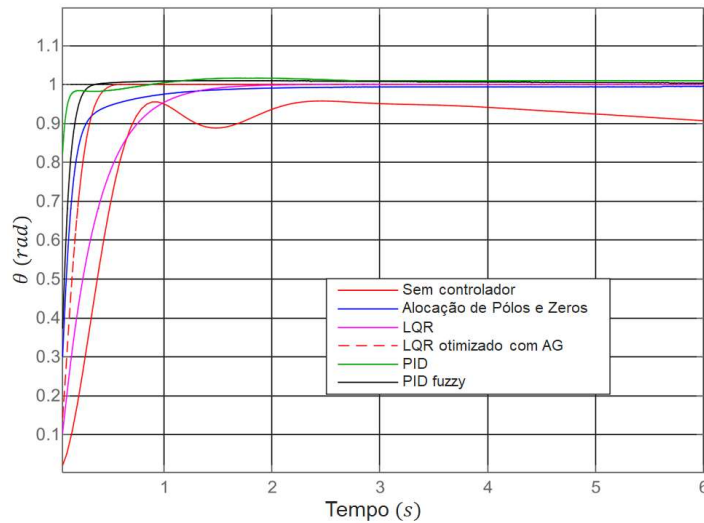


Figura 3.20: Resposta do sistema de arfagem da aeronave (RADHAKRISHNAN e SWARUP, 2020).

Tabela 3.9: Características dos Controladores no Domínio do Tempo (RADHAKRISHNAN e SWARUP, 2020).

Controladores	Tempo de Subida (s)	Tempo de Acomodação (s)
PID	0,0757	0,1558
PID fuzzy	0,1477	0,2536
LQR	0,6963	1,2574
LQR otimizado com AG	0,2612	0,4245
Alocação de pólos e zeros	0,2497	1,1543

Seguindo a mesma linha do trabalho anterior, Li (2020) propõe a utilização de um controlador PID fuzzy, mas neste caso no controle de um VANT de asa rotativa. A arquitetura de controle adotada é basicamente a mesma representada na Fig. 3.19, porém aplicada ao controle de altitude h e dos ângulos de rolagem ϕ , arfagem θ e guinada ψ do veículo aéreo. Os parâmetros K_p , K_i e K_d são considerados fuzzy, e na Fig. 3.21 são apresentadas as superfícies de resposta do SBRF deste controlador. A metodologia de controle adotada, cujos resultados são apresentados na Fig. 3.22, mostra-se eficaz no controle de atitude do veículo aéreo.

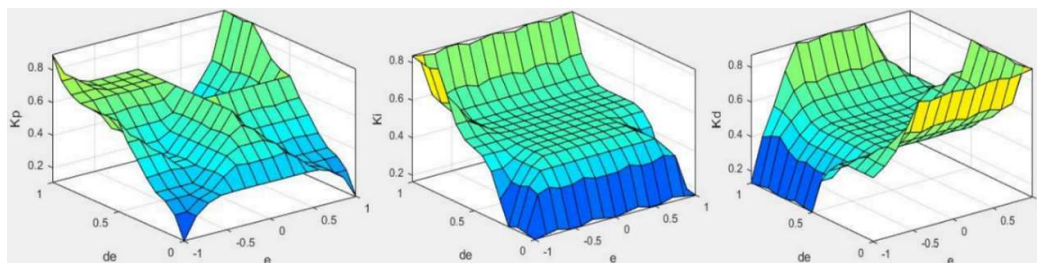


Figura 3.21: Superfícies de resposta do SBRF do controlador PID fuzzy (LI, 2020).

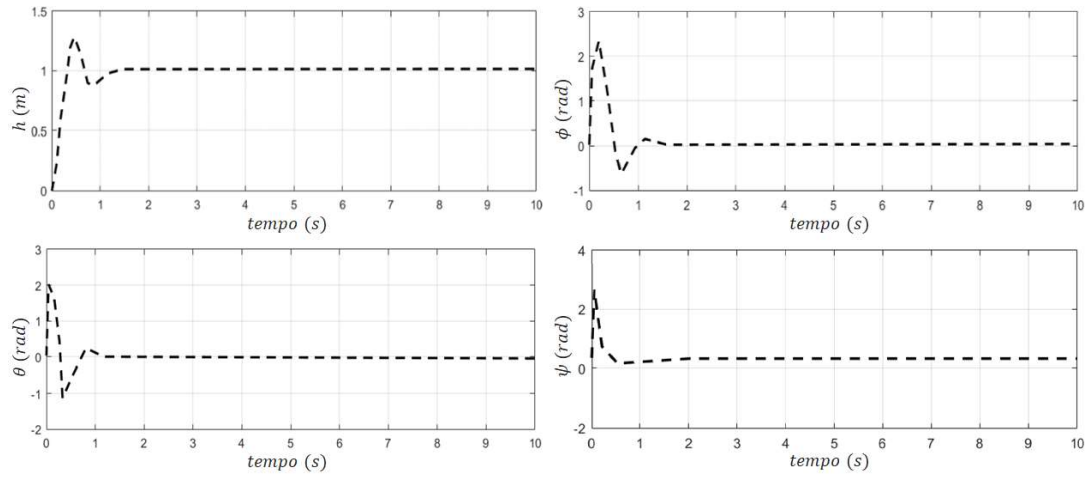


Figura 3.22: Desempenho do controlador PID fuzzy (LI, 2020).

Cui e Zhang (2021) apresentam a aplicação da teoria dos conjuntos fuzzy no controle de voo de um VANT submetido a velocidades próximas à velocidade do som, ou seja, número de mach Ma próximo a 1. A arquitetura de controle do VANT é ilustrada na Fig. 3.23, em que Ref é o valor de referência da velocidade do VANT, que subtraído à resposta do sistema aeronáutico gera o sinal de erro e , o qual gera as duas entradas do sistema (as quais passam pelo processo de fuzzificação FUZ), sendo uma delas a derivada Der do erro; a saída U do sistema de controle é obtida pelo processo de defuzzificação $Defuz$; e por fim, os parâmetros α_e , α_{ec} e α_u são obtidos através de algum método de otimização que minimize o valor do sinal de erro e ao longo do processo de simulação.

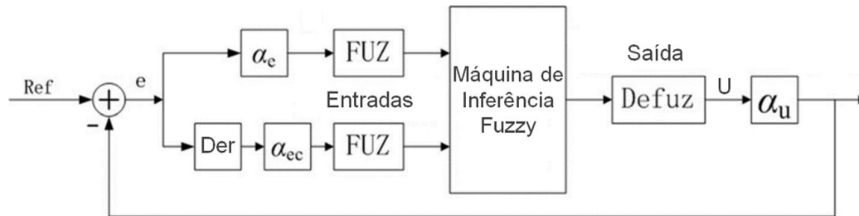


Figura 3.23: Arquitetura de controle fuzzy (CUI e ZHANG, 2021).

O desempenho do sistema de controle com os valores dos parâmetros α_e , α_{ec} e α_u arbitrados, e após o procedimento de otimização, é apresentado na Fig. 3.24, à esquerda. Verifica-se que em ambos os casos o sistema de controle é capaz de manter aeronave em voo em torno da velocidade de referência $Ma = 0,8$, embora após a otimização haja menor oscilação da resposta em torno do valor de referência.

Após o processo de otimização, Cui e Zhang (2021) implementam também uma simulação do tipo *hardware-in-the-loop*, no intuito de validar a técnica por meio da migração do processamento do sistema de controle à unidade de comando real do VANT. O resultado obtido, apresentado na Fig. 3.24 à direita, ratifica a capacidade do controlador em garantir um voo próximo às condições requeridas.

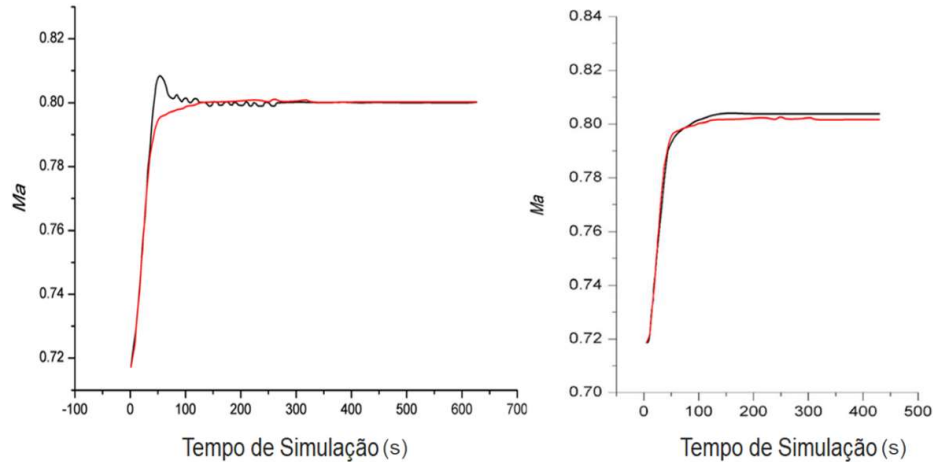


Figura 3.24: À esquerda, desempenho do controlador fuzzy não otimizado, em preto, e otimizado, em vermelho; à direita, resultado do *hardware-in-the-loop*, em preto, e do *software-in-the-loop*, em vermelho (CUI e ZHANG, 2021).

Ulus e Eski (2021) propõem analisar e comparar, em simulação, o desempenho de uma série de arquiteturas de controle fuzzy aplicadas ao controle de atitude de um VANT de asa fixa. Inicialmente é definido o modelo dinâmico linearizado a ser adotado no trabalho, e em seguida são propostas as seguintes arquiteturas de controle, a saber: Controlador PID cujos parâmetros são obtidos a partir do método de Ziegler-Nichols (ZN), controlador puramente fuzzy (FLC) que lança mão do método Mamdani de inferência fuzzy, controlador puramente ANFIS, controlador ANFIS em paralelo com PID (ANFIS+PID), e controlador híbrido PD-Fuzzy-PI. No intuito de ilustrar um dos processos de hibridização expostos no trabalho, na Fig. 3.25 é apresentada a arquitetura de controle ANFIS+PID, sendo $\phi(t)$ o ângulo de rolagem, $\psi(t)$ o ângulo de guinada, $p(t)$ e $r(t)$ são respectivamente as velocidades angulares de rolagem e guinada, e é o sinal de erro, e por fim $u(t)$ é a ação de controle a ser aplicada à aeronave.

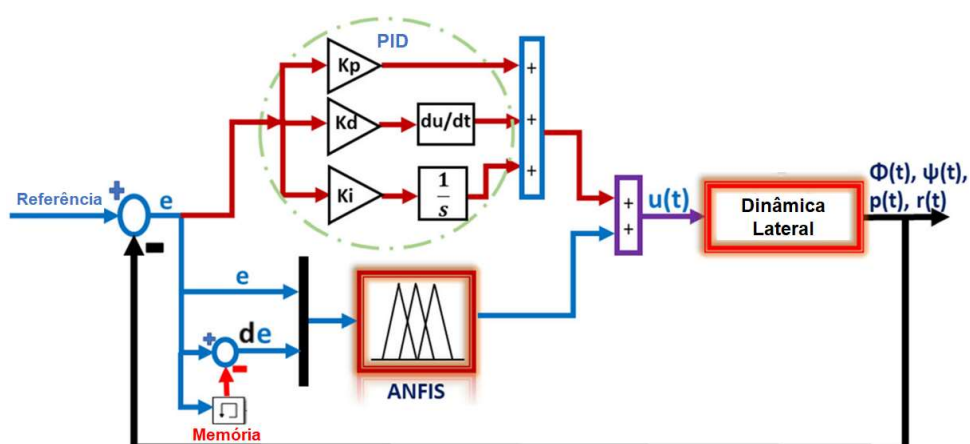


Figura 3.25: Arquitetura de controle ANFIS+PID (ULUS e ESKI, 2021).

A comparação do desempenho das estruturas mencionadas anteriormente é apresentada na Tab. 3.10 e na Fig. 3.26. É possível concluir que todas as arquiteturas conseguem satisfatoriamente controlar o VANT, sobretudo os controladores híbridos, a exemplo do ANFIS+PID e do PD+Fuzzy+PI, que mantêm em simulação os valores de sobressinal, tempo de acomodação e erro em regime permanente satisfatoriamente aceitáveis para a dinâmica lateral do VANT.

Tabela 3.10: Características dos Controladores no Domínio do Tempo (ULUS e ESKI, 2021).

Controladores	Ângulo de Rolagem			Ângulo de Guinada		
	Sobres-sinal (%)	Tempo de Acomodação (s)	Erro em Regime (%)	Sobres-sinal (s)	Tempo de Acomodação (s)	Erro em Regime (%)
PID (ZN)	51,3	0,88	0,01	15,6	0,15	1,1
ANFIS	14,8	0,73	2,65	18,1	0,09	0
ANFIS+PID	33,4	0,74	0,12	12,4	0,07	0
FLC	22,4	0,54	0,90	2,8	0,10	0,6
PD+Fuzzy+PI	1,7	0,10	0,01	1,9	0,11	2,0

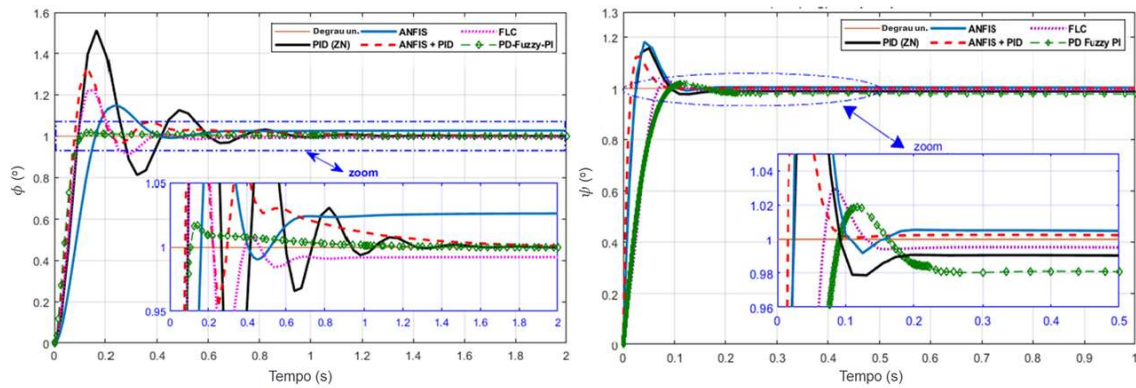


Figura 3.26: Desempenho das arquiteturas de controle com relação aos ângulos de rolagem, à esquerda, e guinada, à direita (ULUS e ESKI, 2021).

Esta tese, de forma semelhante a Ulus e Eski (2021), lança mão de uma arquitetura de controle baseada na teoria dos conjuntos fuzzy, porém inova ao propor uma nova técnica de inferência fuzzy para tal, que alia intuitividade, dispensa o uso de métodos de integração numérica, e que se presta não somente ao controle da dinâmica látero-direcional da aeronave, como em Ulus e Eski (2021), mas também ao controle da dinâmica longitudinal. Esta técnica é utilizada na programação computacional do sistema neuro-fuzzy desenvolvido neste trabalho, que é aplicada na arquitetura de controle da aeronave, cuja estabilidade é verificada numericamente. Além disso, realizam-se análises, em simulador de voo catalogado pela FAA, da capacidade do sistema de controle desenvolvido em controlar uma aeronave submetida a condições críticas de voo.

Após a revisão de literatura, no Capítulo IV é apresentada a metodologia adotada para a execução das atividades previstas para esta tese.

CAPÍTULO IV

METODOLOGIA

A metodologia adotada para a execução do trabalho é dividida em etapas, que são apresentadas na Fig. 4.1, e a seguir estas etapas são esquematizadas no intuito de facilitar a compreensão do leitor quanto aos principais procedimentos adotados para o cumprimento dos objetivos desta tese.

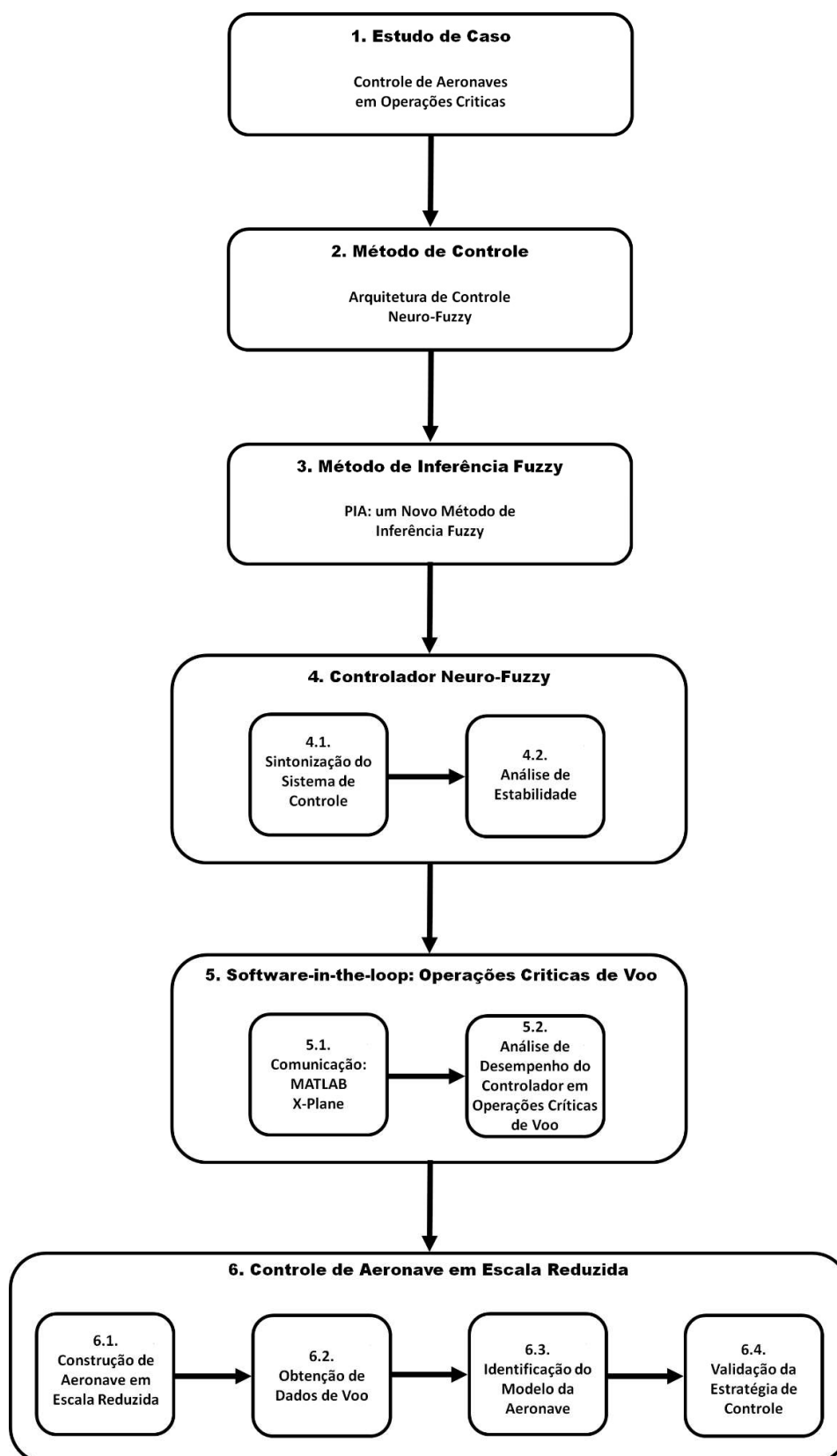


Figura 4.1: Esquema das etapas que compõem a metodologia da tese (autoria própria).

1. Estudo de Caso

Nesta etapa é definido o escopo das investigações científicas presentes neste trabalho, que busca aplicar o novo método PIA de inferência fuzzy no controle de aeronaves em operações críticas de voo.

Sabe-se que LOC-I é causa da grande maioria das fatalidades ocorridas na aviação comercial (IATA, 2020). Esse fato motiva o estudo do controle de aeronaves em operações críticas, a exemplo de condições meteorológicas severas, rajadas de vento, mau funcionamento da aeronave e o mau funcionamento ou falha do sistema propulsor.

Neste trabalho, para o desenvolvimento da arquitetura de controle, lança-se mão da inteligência artificial em função da viabilidade tecnológica para tal (existência no mercado de controladores de voo de alta capacidade de processamento), e sobretudo da versatilidade e adaptabilidade das técnicas existentes.

Para a realização das simulações necessárias para validação das técnicas, utiliza-se o software MATLAB em paralelo com o simulador de voo X-Plane 11, que é certificado pela FAA para o treinamento de pilotos. Nestas simulações, utiliza-se o Cessna 172 como objeto de estudo em função da grande disponibilidade de dados na literatura desta aeronave. Já para verificar a eficácia da técnica com relação à utilização de modelos obtidos a partir de dados experimentais, realizam-se ensaios de voo com uma aeronave Cessna 172 em escala reduzida.

O esquema apresentado na Fig. 4.2 é um resumo do estudo de caso utilizado neste trabalho.

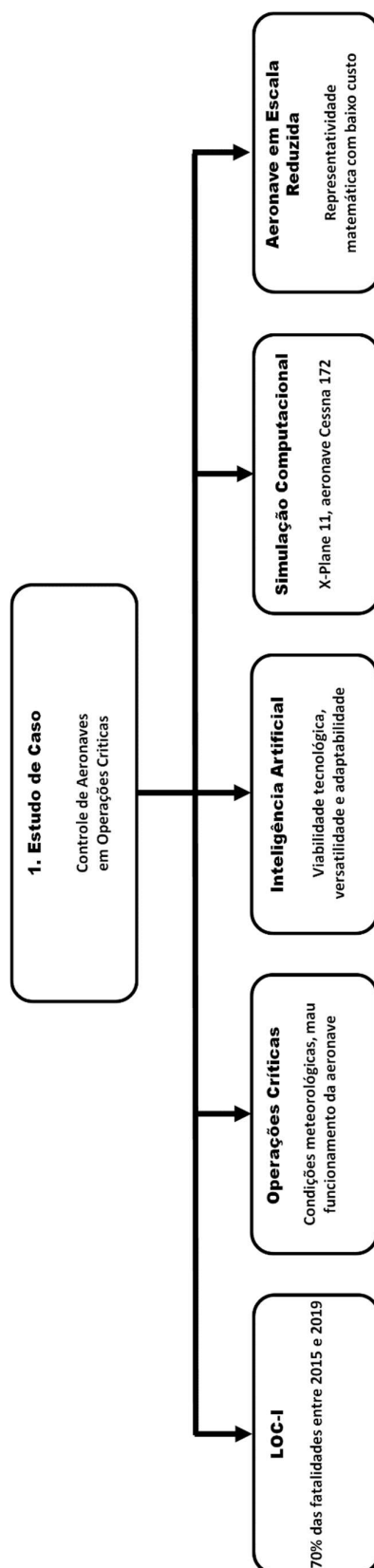


Figura 4.2: Esquema do estudo de caso desta tese (autoria própria).

2. Método de Controle

A partir da revisão bibliográfica apresentada no capítulo III, além de simulações previamente realizadas relativas a técnicas desenvolvidas por outros autores, é estabelecida uma arquitetura de controle neuro-fuzzy para aeronaves, técnica híbrida que alia a alta capacidade de adaptação das redes neurais, com a robustez e intuitividade da teoria dos conjuntos fuzzy.

Ressalta-se que a arquitetura de controle concebida é capaz de controlar as dinâmicas longitudinal e látero-direcional de uma aeronave. O esquema apresentado na Fig. 4.3 é um resumo do método de controle utilizado neste trabalho.

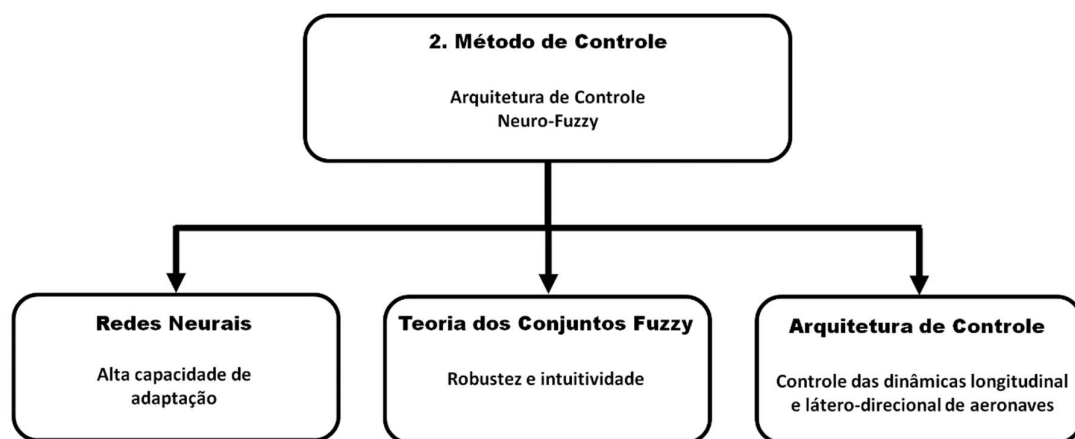


Figura 4.3: Esquema do método de controle (autoria própria).

3. Método de Inferência Fuzzy

A utilização dos métodos clássicos de inferência fuzzy mostra-se pouco eficiente no efetivo controle de um sistema aeronáutico. Dessa forma, no intuito de aliar a eficiência computacional do método de inferência fuzzy de Takagi-Sugeno, com a intuitividade do método de Mamdani, propõe-se o desenvolvimento de uma nova técnica de inferência fuzzy, denominada *Pondered Individual Analysis* (PIA).

O método PIA é baseado na análise e tradução matemática de cada proposição fuzzy que compõe a base de regras, ponderando o efeito de cada variável de entrada na saída de um SBRF. A definição dos conjuntos fuzzy de entrada e saída, bem como a base de regras

de um SBRF, segue o mesmo procedimento adotado no método de Mamdani, entretanto, o método pode ser entendido como um caso particular do método de Takagi-Sugeno. Isso ocorre porque as funções consequentes também são determinadas para cada subconjunto fuzzy da saída e, a partir da média ponderada ou de outro método de defuzzificação que dispense técnicas de integração, fornecem a resposta de um SBRF.

O esquema apresentado na Fig. 4.4 é um resumo do método PIA de inferência fuzzy.

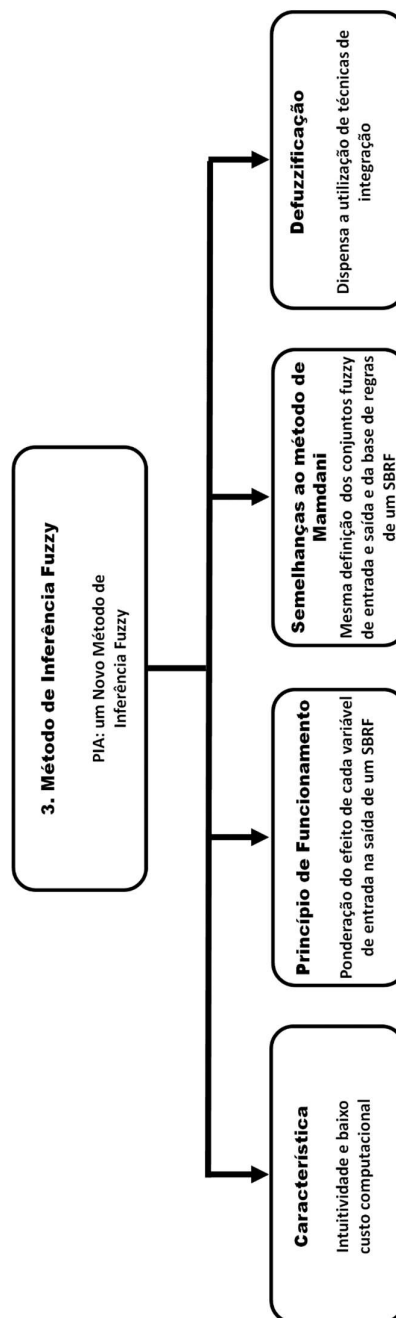


Figura 4.4: Esquema do método PIA de inferência fuzzy (autoria própria).

4. Controlador Neuro-Fuzzy

O controlador neuro-fuzzy desenvolvido passa por simulações no software MATLAB a fim de, a partir do modelo dinâmico do comportamento da aeronave em voo, sintonizar os parâmetros do controlador por meio de treinamento off-line e verificar numericamente a estabilidade do sistema.

O processo de sintonização do sistema de controle passa pela obtenção das médias e desvios padrão que definem as funções de pertinência das entradas e saídas dos SBRFs. O processo de otimização é conduzido por meio da simulação da operação da aeronave Cessna 172, em que são adotados valores de referência para as variáveis do processo, assim como condições iniciais e perturbações. O método de otimização utilizado é a evolução diferencial, aplicada à minimização do erro médio quadrático com relação aos valores de referência do sistema de controle. Consideram-se para essa simulação os modelos dinâmicos longitudinal e látero-direcional, linearizados através da teoria das pequenas perturbações, e o desempenho da técnica desenvolvida é comparado aos obtidos com os métodos PID e ANFIS (método de inferência fuzzy de Takagi-Sugeno).

O esquema apresentado na Fig. 4.5 é um resumo da metodologia aplicada à sintonização do controlador neuro-fuzzy.

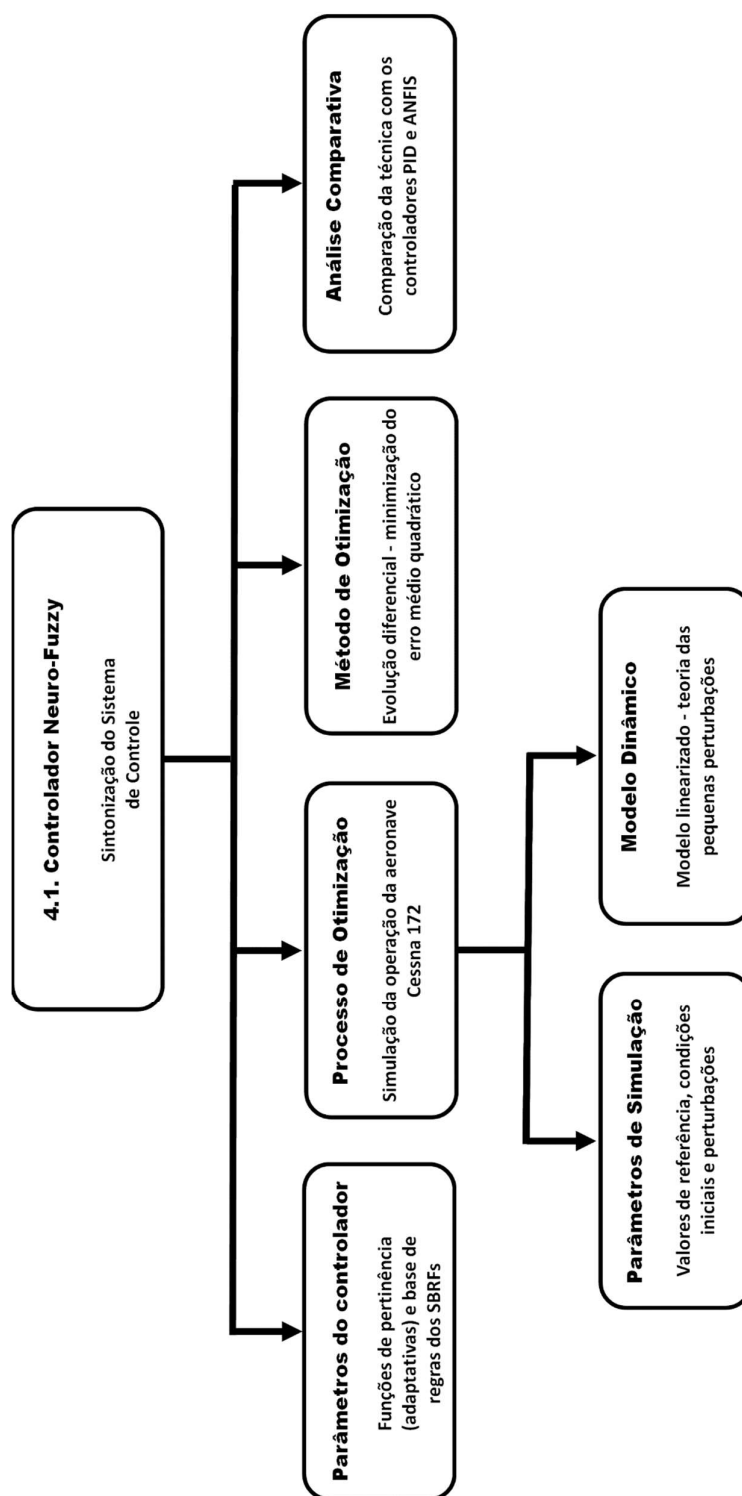


Figura 4.5: Esquema da sintonização do sistema de controle (autoria própria).

Após o processo de sintonização, principalmente por se tratar de um sistema aeronáutico, é fundamental que seja feita a análise da estabilidade do sistema controlado. Para a verificação de sua estabilidade assintótica no sentido de Lyapunov, realiza-se a análise da derivada da função de Lyapunov associada ao sistema, devendo o valor desta ser menor ou igual a zero para todo o domínio analisado.

Ressalta-se que a verificação da estabilidade é realizada neste trabalho de forma numérica, assim como em outros trabalhos (KIM, KANG e PARK, 1999; KIM, 2001), em função do elevado nível de complexidade de se reproduzir a mesma análise, porém de forma analítica. A análise numérica da estabilidade é desenvolvida considerando-se o modelo linearizado da aeronave, assim, discretiza-se o domínio do sistema de equações que rege a dinâmica da planta e avalia-se o valor da derivada da função de Lyapunov para cada condição testada.

O esquema apresentado na Fig. 4.6 é um resumo da metodologia aplicada à análise de estabilidade do controlador neuro-fuzzy.

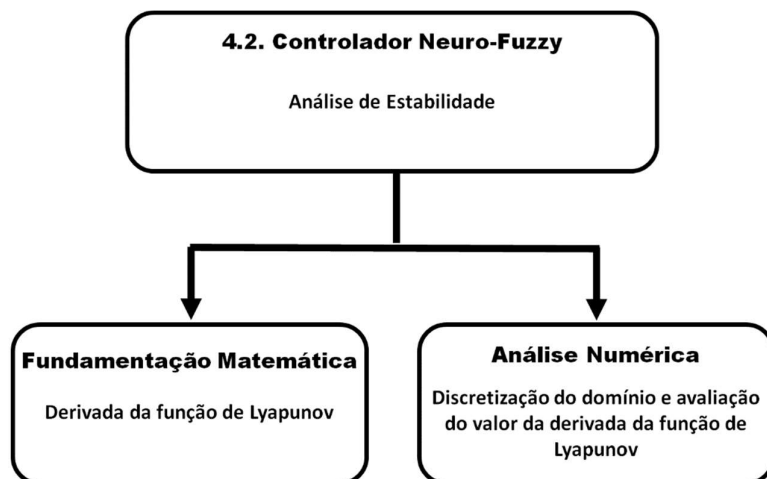


Figura 4.6: Esquema da análise de estabilidade do sistema de controle (autoria própria).

5. Software-in-Loop: Operações Críticas de Voo

A viabilidade de utilização do simulador de voo X-Plane se dá a partir do desenvolvimento de uma interface de comunicação entre o software MATLAB e o simulador de voo. Tal interface viabiliza a realização de simulações do tipo *software-in-the-loop* entre as duas plataformas, o que permite executar análises da capacidade da metodologia proposta em controlar uma aeronave submetida a operações críticas de voo.

A comunicação entre os softwares se dá em tempo real através do protocolo de comunicação UDP (*User Datagram Protocol*). No MATLAB, é realizada a implementação do controlador neuro-fuzzy e da interface de comunicação com o simulador de voo X-Plane 11, sendo que cada variável da aeronave no simulador de voo é recebida no MATLAB como sendo um pacote de dados pré-estruturados contendo 32 bytes, e a resposta do sistema de controle é calculada, transformada em um pacote de dados reconhecível pelo simulador e endereçada a este. No software X-Plane 11, realiza-se a simulação do comportamento dinâmico da aeronave. Os resultados obtidos são enviados em tempo real ao MATLAB através da especificação de IP e porta correspondentes.

O esquema apresentado na Fig. 4.7 é um resumo da metodologia aplicada à comunicação entre os softwares MATLAB e X-Plane.

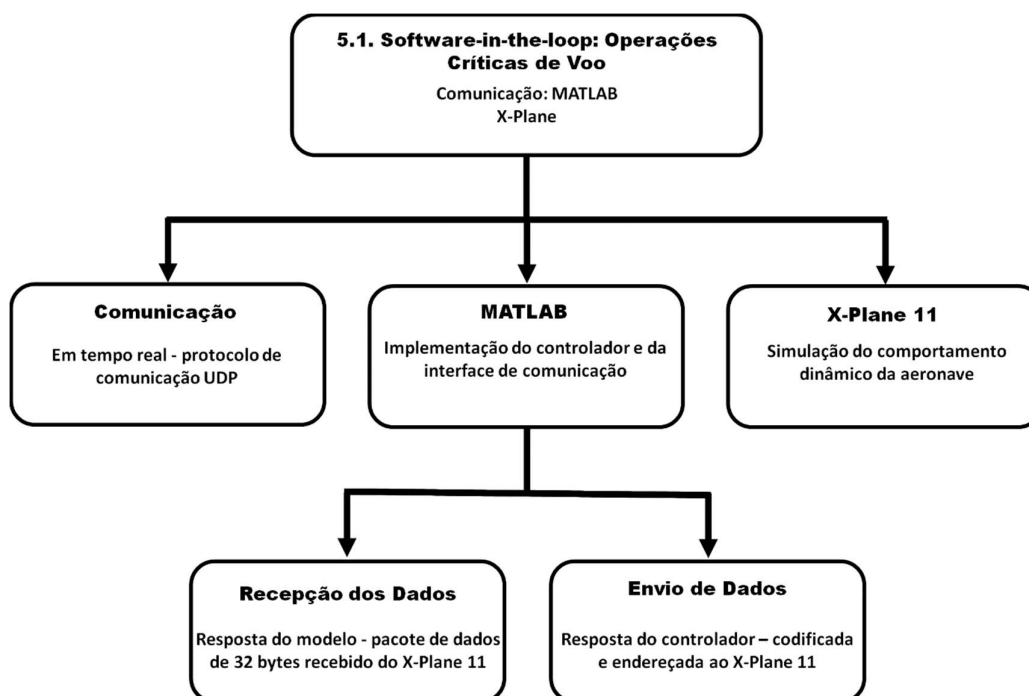


Figura 4.7: Esquema da comunicação entre MATLAB e X-Plane (autoria própria).

Após concluída a comunicação entre os aplicativos, busca-se analisar o desempenho do controlador, sintonizado na etapa anterior, em operações críticas de voo, sendo a análise gráfica e o desvio em relação ao sinal de referência os métodos utilizados na avaliação do método de controle. As operações críticas analisadas são: condições meteorológicas severas e rajadas de vento, mau funcionamento da aeronave, falha do sistema propulsor, e todas as condições anteriores simultaneamente.

O esquema apresentado na Fig. 4.8 é um resumo da metodologia utilizada na análise de desempenho do controlador em operações críticas de voo.

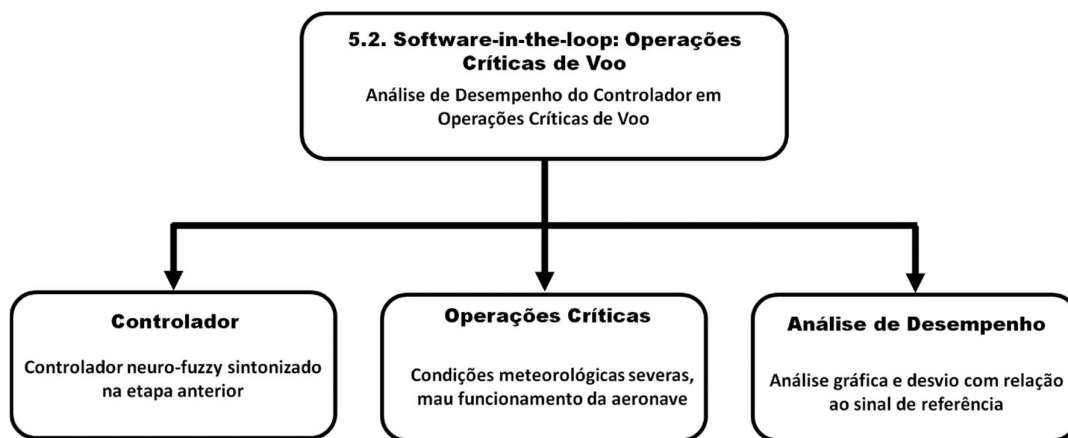


Figura 4.8: Esquema da análise de desempenho do controlador em operações críticas de voo (autoria própria).

6. Controle de Aeronave em Escala Reduzida

Nesta etapa é construído um protótipo de aeronave em escala reduzida, Fig. 4.9, e são realizados ensaios de voo no intuito de se extrair dados a serem utilizados na modelagem do VANT. Aeronaves em escala reduzida possibilitam a validação das técnicas de controle de forma experimental, e com baixo custo financeiro se comparados os ensaios aos testes em aeronaves comerciais. Utiliza-se nos ensaios experimentais uma aeronave Cessna 172 em escala reduzida de 1:10, associada a um controlador de voo Pixhawk.

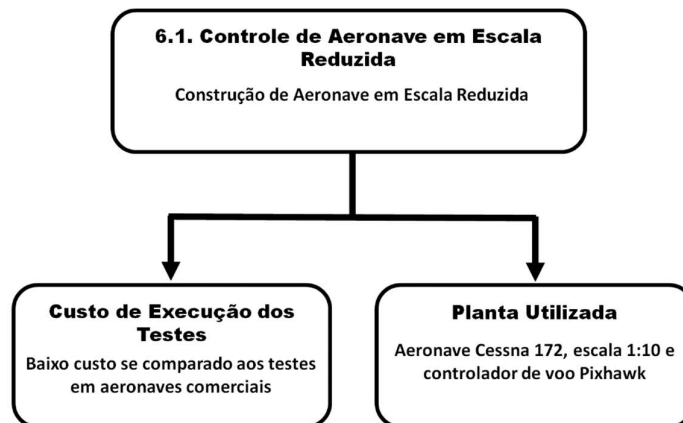


Figura 4.9: Esquema relativo à construção de aeronave em escala reduzida (autoria própria).

Para a extração dos dados experimentais, Fig. 4.10, estabiliza-se a aeronave em voo e aplicam-se perturbações à mesma, tudo por meio do piloto do aeromodelo. Durante os ensaios, o próprio controlador de voo Pixhawk é capaz de armazenar os dados extraídos em voo para posterior análise.

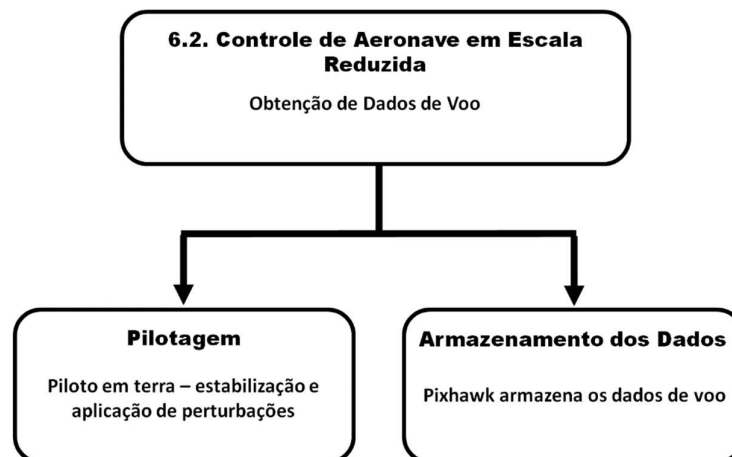


Figura 4.10: Esquema relativo à obtenção de dados de voo (autoria própria).

Após a obtenção dos dados de voo, realiza-se a identificação do modelo dinâmico da aeronave, Fig. 4.11, levando-se em conta neste caso apenas a alteração do ângulo de arfagem da aeronave em função da atuação do profundor, buscando-se assim um modelo do tipo SISO (*Single Input Single Output*). A identificação é obtida a partir do ajuste do modelo aos dados experimentais extraídos da aeronave (minimização do erro médio quadrático deste ajuste) a partir do método da evolução diferencial.

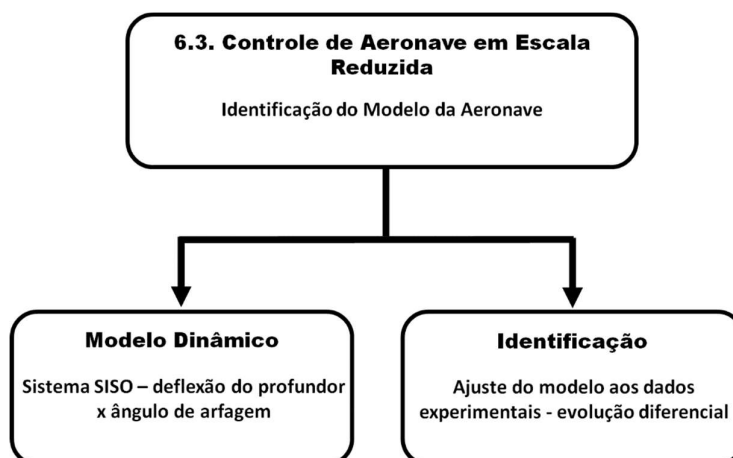


Figura 4.11: Esquema que resume a identificação do modelo da aeronave (autoria própria).

Por fim, chega-se novamente em um instante de validação do método PIA e da estratégia de controle adotada neste trabalho, Fig. 4.12. Utiliza-se, em uma simulação no MATLAB, um sistema neuro-fuzzy aliado ao método de inferência PIA no controle do aeromodelo. Apresenta-se, então, uma análise comparativa da técnica desenvolvida com os métodos PID e ANFIS.

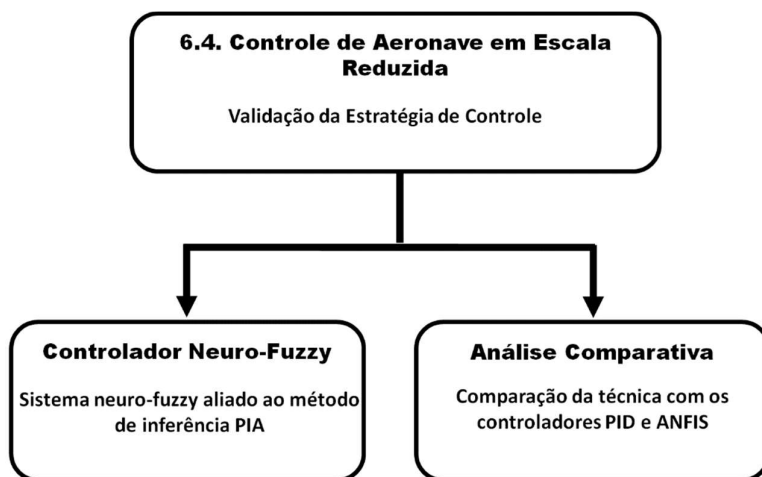


Figura 4.12: Esquema que ilustra a validação da estratégia de controle (autoria própria).

Após a apresentação da metodologia, inicia-se a exposição do desenvolvimento deste trabalho, a começar pelo PIA, novo método de inferência fuzzy proposto nesta tese e utilizado na concepção da arquitetura de controle de aeronaves.

CAPÍTULO V

PIA: UM NOVO MÉTODO DE INFERÊNCIA FUZZY

A análise individual ponderada, PIA (*Pondered Individual Analysis*), é um método de inferência fuzzy que combina baixo custo computacional e capacidade de ser utilizado na otimização de sistemas, e na modelagem e controle de processos. A técnica é baseada na análise e tradução matemática de cada proposição fuzzy que compõe a base de regras, ponderando o efeito de cada variável de entrada na saída de um SBRF.

A seguir, é apresentada uma breve descrição dos procedimentos para executar o método PIA.

1. A definição dos conjuntos fuzzy de entrada e saída, bem como a base de regras de um SBRF, segue o mesmo procedimento adotado no método Mamdani.
2. A união e a interseção de conjuntos fuzzy são realizadas respectivamente pelos operadores t-norma (por exemplo, operador máximo) e s-norma (por exemplo, operador mínimo).
3. Embora, no método PIA, as funções de pertinência sejam definidas para cada conjunto fuzzy das variáveis de saída, o método pode ser entendido como um caso particular do método de Takagi-Sugeno. Isso ocorre porque as funções consequentes f_{C_j} também são determinadas para cada conjunto fuzzy C_j da saída e, a partir da média ponderada ou de outro método de defuzzificação que não envolva integração numérica, fornecem a resposta de um SBRF.

A seção 5.1 apresenta a formalização conceitual e matemática do método PIA.

5.1. Método PIA: Descrição, Conceitos e Definições

As definições elaboradas neste trabalho e apresentadas a seguir estabelecem as funções consequentes e formalizam matematicamente o método PIA.

Definição 1: O ponto em que uma função de pertinência de um número fuzzy possui valor unitário é chamado de ponto central CP, Fig. 5.1. No caso de funções de pertinência que possuam mais de um ponto nessa condição, CP é definido através aplicação da média aritmética a esses pontos. Pode-se notar que qualquer função de pertinência, por exemplo, triangular, trapezoidal ou gaussiana, mesmo assimétrica, pode ser usada na construção de um SBRF para o método PIA.

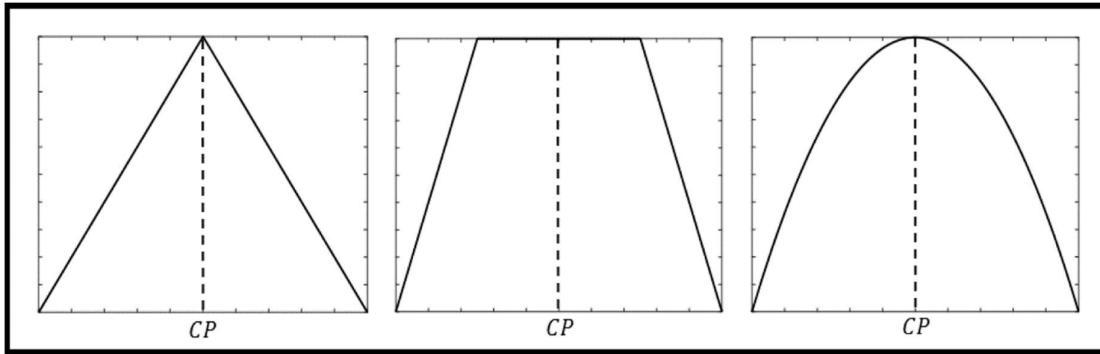


Figura 5.1: Definição dos pontos centrais das funções de pertinência (autoria própria).

Definição 2: Considerando dois conjuntos fuzzy S_1 e S_2 , respectivamente apresentando funções de pertinência com os pontos centrais CP_1 e CP_2 , e pertencentes ao mesmo conjunto universo U :

- Se $CP_1 < CP_2$, então $S_1 <_s S_2$;
- Se $CP_1 > CP_2$, então $S_1 >_s S_2$;
- Se $CP_1 = CP_2$, então $S_1 =_s S_2$;

em que os símbolos $<_s$, $>_s$ e $=_s$ significam, respectivamente, semanticamente menor que, semanticamente maior que, e semanticamente igual a.

A partir das definições descritas anteriormente, é possível elaborar novos conceitos para relacionar as variáveis de entrada e as variáveis de saída de um SBRF.

Definição 3: Para cada regra fuzzy ou proposição i de um SBRF, é verificado o efeito do aumento ou diminuição de cada variável de entrada sob a variável de saída. Diz-se que a relação entre uma variável de entrada IN , e uma variável de saída OUT de um SBRF é:

- **Direta**, representado por IN **Dir** OUT , quando o aumento e a diminuição da variável IN causam, respectivamente, o aumento e a diminuição da variável OUT ;
- **Inversa**, representada por IN **Inv** OUT , quando o aumento e a diminuição da variável IN causam, respectivamente, a diminuição e o aumento da variável OUT ;
- **Neutra**, representada por IN **Neut** OUT , quando não é possível inferir sobre a relação entre IN e OUT .

Um exemplo de base de regras é construído e apresentado na Tab. 5.1, em que X e Y são as entradas do SBRF e Z é a saída. Os conjuntos fuzzy de entrada são A_1, A_2, A_3, B_1, B_2 e B_3 ; e C_1, C_2 e C_3 são os conjuntos fuzzy de saída, sendo $A_1 <_s A_2 <_s A_3, B_1 <_s B_2 <_s B_3$ e $C_1 <_s C_2 <_s C_3$. Na Tab. 5.2 é exposta a relação entre as variáveis de entrada e saída para cada regra $R_i, i = 1, 2, \dots, 9$.

Tabela 5.1: Exemplo 1 - Base de Regras.

Regras	Se	X	E	Y	Então	Z
R_1		A_1		B_1		C_1
R_2		A_2		B_1		C_2
R_3		A_3		B_1		C_3
R_4		A_1		B_2		C_3
R_5		A_2		B_2		C_2
R_6		A_3		B_2		C_1
R_7		A_1		B_3		C_2
R_8		A_2		B_3		C_2
R_9		A_3		B_3		C_3

Tabela 5.2: Exemplo 1 - Relação Entre as Variáveis.

Regra	Relação: $X \rightarrow Z$	Justificativa	Relação: $Y \rightarrow Z$	Justificativa
R_1	$X \text{ Dir } Z$	$C_1 <_S C_2$	$Y \text{ Dir } Z$	$C_1 <_S C_3$
R_2	$X \text{ Dir } Z$	$C_2 >_S C_1, C_2 <_S C_3$	$Y \text{ Neut } Z$	$C_2 =_S C_2$
R_3	$X \text{ Dir } Z$	$C_3 >_S C_2$	$Y \text{ Inv } Z$	$C_3 >_S C_1$
R_4	$X \text{ Inv } Z$	$C_3 >_S C_2$	$Y \text{ Neut } Z$	$C_3 >_S C_1, C_3 >_S C_2$
R_5	$X \text{ Inv } Z$	$C_2 <_S C_3, C_2 >_S C_1$	$Y \text{ Neut } Z$	$C_2 =_S C_2$
R_6	$X \text{ Inv } Z$	$C_1 <_S C_2$	$Y \text{ Neut } Z$	$C_1 <_S C_3$
R_7	$X \text{ Neut } Z$	$C_2 =_S C_2$	$Y \text{ Inv } Z$	$C_2 <_S C_3$
R_8	$X \text{ Neut } Z$	$C_2 =_S C_2, C_2 <_S C_3$	$Y \text{ Neut } Z$	$C_2 =_S C_2$
R_9	$X \text{ Dir } Z$	$C_3 >_S C_2$	$Y \text{ Dir } Z$	$C_3 >_S C_1$

Ressalta-se que esta análise da relação entre as variáveis de entrada e saída pode ser, sem grandes dificuldades, programada por um profissional responsável pela construção de um eventual *toolbox* para o método. Em outras palavras, como as relações são unicamente dependentes da base de regras adotada, a construção das relações de um SBRF pode ser programada previamente, de uma forma tal que o usuário da técnica apenas se preocupe com a definição das variáveis de entrada e saída, funções de pertinência e base de regras, da mesma forma feita quando se lança mão do método de Mamdani.

Definição 4: Para cada regra R_i , pode-se calcular uma contribuição para a saída de um SBRF a partir do procedimento descrito a seguir:

1. Dada uma variável de entrada IN e seu valor in , o grau de pertinência de in em relação ao subconjunto de entrada mapeado pela regra R_i , μ , é projetado para o subconjunto fuzzy de saída. Assume-se que μ defina dois candidatos out_{cand} e out_{ca} da variável de saída OUT para compor a solução do SBRF, Fig. 5.2.

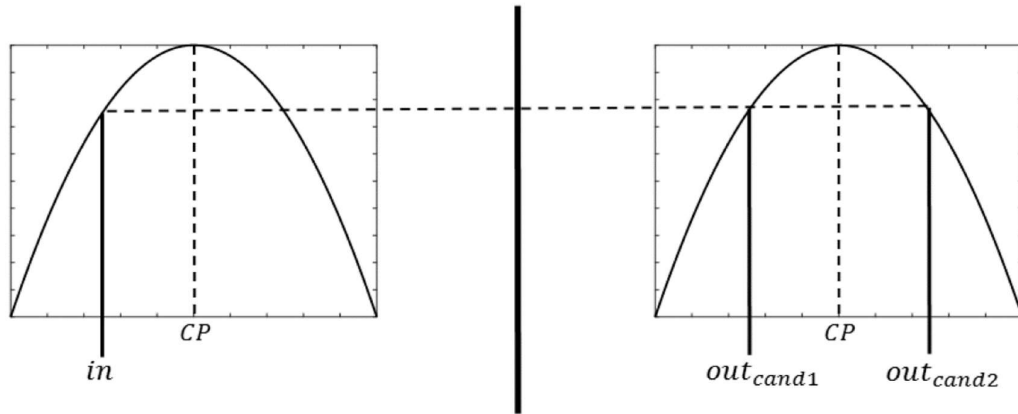


Figura 5.2: Valores dos candidatos (autoria própria).

2. A escolha de out_{cand1} ou out_{cand2} leva em consideração a base de regras do SBRF e está em função das seguintes condições:
 - SE $in < CP$ E IN **Dir** OUT, ENTÃO out_{cand1} ;
 - SE $in > CP$ E IN **Inv** OUT, ENTÃO out_{can} ;
 - SE $in < CP$ E IN **Inv** OUT, ENTÃO out_{cand} ;
 - SE $in > CP$ E IN **Dir** OUT, ENTÃO out_{cand2} ;
 - Em qualquer outro caso, a escolha do valor para compor a solução do SBRF é dada pela média aritmética de out_{cand1} e out_{cand2} .
3. Todo o processo é replicado para cada uma das n_{in} variáveis de entrada, de modo que para cada regra R_i , é possível definir uma contribuição de saída $cont_{Cjm}$, Eq. (5.1), dado que C_j é um conjunto fuzzy de saída, e m indica o número vezes que o conjunto C_j já foi relacionado na base de regras:

$$cont_{Cjm} = \frac{\sum_{k=1}^{n_{in}} \alpha_k out_{ki}}{\sum_{k=1}^{n_{in}} \alpha_k} \quad (5.1)$$

sendo out_{ki} o valor escolhido e α_k o peso associado à influência na variável de saída, ambos com relação à variável de entrada k . A variável α_k é implementada nesta equação para impor uma maior apreciação do efeito de certas variáveis de entrada na saída. Quando essas informações não estão disponíveis ou se todas as variáveis têm o mesmo efeito na saída do processo, assume-se que $\alpha_k = 1$.

A partir das definições apresentadas anteriormente, pode-se definir as funções consequentes f_{Cj} associadas a cada conjunto fuzzy de saída e, finalmente, determinar a

resposta do SBRF. A seguir, é apresentado um exemplo que ilustra a resolução de um SBRF pelo método PIA.

Considerando a base de regras exibida na Tab. 5.3, X e Y são entradas do SBRF e Z é a saída. Os conjuntos fuzzy de entrada são A_1 , A_2 , B_1 e B_2 ; e C_1 e C_2 são os conjuntos fuzzy de saída, sendo $A_1 <_s A_2$, $B_1 <_s B_2$ e $C_1 <_s C_2$.

Tabela 5.3: Exemplo 2 - Base de Regras.

Se	Se X	E Y	Então Z
R_1	A_1	B_1	C_1
R_2	A_2	B_1	C_2
R_3	A_1	B_2	C_1
R_4	A_2	B_2	C_1

A Tab. 5.4 apresenta a relação entre as variáveis de entrada e saída. Para exemplificar a obtenção dessas relações, é analisada a regra R_1 :

- X **Dir** Z , porque manter $Y =_s B_1$, e aumentar X de A_1 para A_2 , causa o aumento da variável Z de C_1 para C_2 ;
- Y **Neut** Z , porque manter $X =_s A_1$, e aumentar Y de B_1 para B_2 , não altera Z .

Tabela 5.4: Exemplo 2 - Relação Entre as Variáveis.

Regra	Relação: $X \rightarrow Z$	Relação: $Y \rightarrow Z$
R_1	X Dir Z	Y Neut Z
R_2	X Dir Z	Y Inv Z
R_3	X Neut Z	Y Neut Z
R_4	X Neut Z	Y Inv Z

Na Fig. 5.3, é resumido o procedimento para encontrar o peso, w_{Cjm} , e a contribuição de saída, $cont_{Cjm}$, de cada regra R_i . Como um exemplo, a regra R_1 é analisada:

- $z_{x1} = z_{cand2}$, porque $x > CP_{A1}$ e X *Dir* Z ;
- $z_{y1} = \frac{z_{cand1} + z_{cand2}}{2}$, porque Y *Neut* Z .

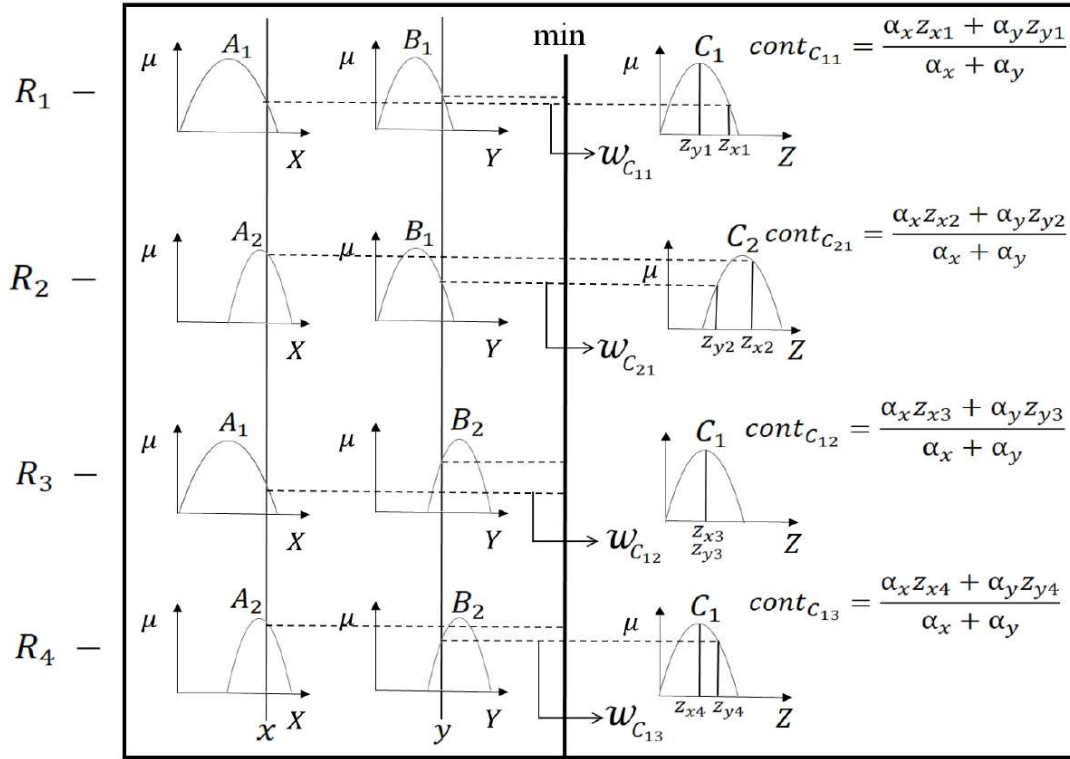


Figura 5.3: Ilustração do método PIA, em que α_x e α_y são os pesos associados, respectivamente, à influência de x e y na variável de saída (autoria própria).

Após definir os pesos w_{cjm} e as contribuições $cont_{cjm}$ de cada regra, o próximo passo é calcular o valor da função consequente f_{cj} de cada conjunto fuzzy C_j da saída, Eq. (5.2), em que T_j é o total de regras relacionadas ao conjunto C_j :

$$f_{cj} = \frac{\sum_{m=1}^{T_j} w_{cjm} cont_{cjm}}{\sum_{m=1}^{T_j} w_{cjm}}. \quad (5.2)$$

Neste exemplo, as funções consequentes são obtidas da seguinte maneira:

- $f_{C1} = \frac{w_{C11} cont_{C11} + w_{C12} cont_{C12} + w_{C13} cont_{C13}}{w_{C11} + w_{C12} + w_{C13}};$
- $f_{C2} = \frac{w_{C21} cont_{C21}}{w_{C21}} = cont_{C21}.$

Os pesos w_{cj} de cada conjunto fuzzy de saída podem ser obtidos pelo operador *max*:

$$w_{cj} = \max(w_{cjm}), m = 1, \dots, T_j. \quad (5.3)$$

Finalmente, a defuzzificação do SBRF é realizada a partir do método da média ponderada, Eq. (5.4), sendo n_c o número de conjuntos fuzzy que compõe a variável de saída Z , e o valor de saída, z , sendo dado por:

$$z = \frac{\sum_{j=1}^{n_c} W_{Cj} f_{Cj}}{\sum_{j=1}^{n_c} W_{Cj}} = \frac{W_{C1} f_{C1} + W_{C2} f_{C2}}{W_{C1} + W_{C2}}. \quad (5.4)$$

Dois fluxogramas que resumem a definição e a avaliação de um SBRF que usa o PIA como o método de inferência fuzzy são apresentados na Fig. 5.4.

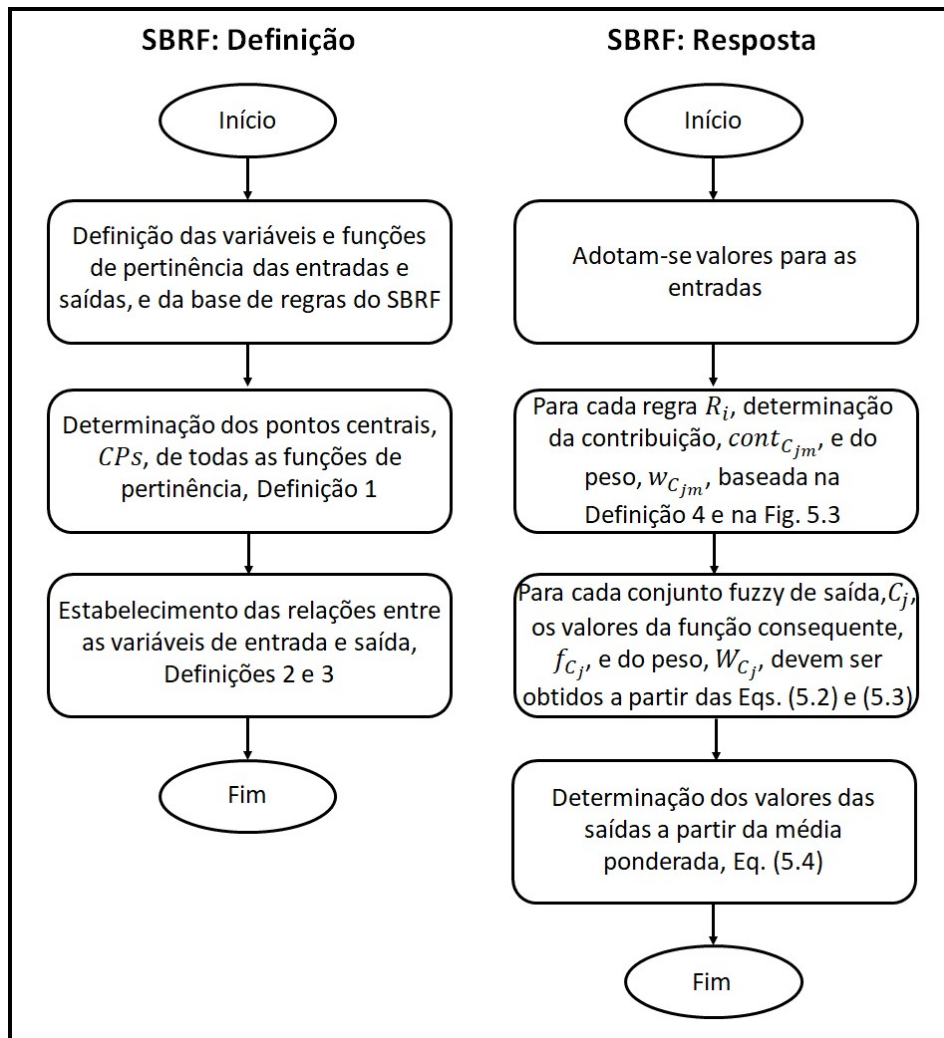


Figura 5.4: Fluxogramas do método PIA (autoria própria).

Após a apresentação do método PIA de inferência fuzzy, a seção 5.2 mostra um estudo de caso comparando o desempenho matemático e computacional dessa técnica com outras abordagens fuzzy.

5.2. Estudo de Caso: Modelagem Aerodinâmica

Esta seção explora a aplicação e a comparação do método PIA com os métodos de inferência fuzzy Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen na modelagem aerodinâmica de uma aeronave Cessna 172.

A direção relativa do vento, o ângulo de ataque da aeronave, α , a deflexão do profundor, δ_e , e as forças de sustentação, L , e arrasto, D , são mostrados na Fig. 5.5.



Figura 5.5: Variáveis envolvidas no processo de modelagem (autoria própria).

As forças de sustentação e arrasto da aeronave são sempre perpendiculares entre si e, para um voo em regime permanente, são definidas respectivamente pelas Eqs. (5.5) e (5.6):

$$L = \frac{1}{2} \rho V^2 S C_L, \quad (5.5)$$

$$D = \frac{1}{2} \rho V^2 S C_D, \quad (5.6)$$

onde ρ é a densidade do ar, V é a velocidade do vento relativo, S é a área de referência da aeronave, e C_L e C_D são respectivamente os coeficientes de sustentação e arrasto.

O coeficiente sustentação da aeronave depende de muitas variáveis. De acordo com Roskam (2001), o coeficiente de sustentação para um voo em regime permanente é definido como:

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{i_h}} i_h + C_{L_{\delta_e}} \delta_e, \quad (5.7)$$

em que C_{L_0} é o coeficiente linear da equação, C_{L_α} é o coeficiente de sustentação devido ao ângulo de ataque, $C_{L_{ih}}$ é o coeficiente de sustentação devido à deflexão do estabilizador horizontal, i_h , e $C_{L_{\delta e}}$ é o coeficiente de sustentação devido à deflexão do profundor.

O coeficiente de arrasto C_D , analogamente ao C_L , possui a mesma equação para um voo em regime permanente, definida como:

$$C_D = C_{D_0} + C_{D_\alpha} \alpha + C_{D_{ih}} i_h + C_{D_{\delta e}} \delta e. \quad (5.8)$$

A aeronave Cessna 172 possui estabilizador horizontal fixo, de modo que as equações para C_L e C_D podem ser simplificadas como:

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta e}} \delta e; \\ C_D &= C_{D_0} + C_{D_\alpha} \alpha + C_{D_{\delta e}} \delta e. \end{aligned} \quad (5.9)$$

Como a relação entre as entradas e saídas deste processo não pode ser extraída linearmente para todas as condições de voo devido às incertezas intrínsecas, este trabalho opta por considerar esse processo como um sistema fuzzy.

Em sequência é descrito o desenvolvimento de um SBRF em relação a C_D e C_L em função de α e δe . A base teórica para a concepção do sistema é a Eq. (5.9) e as informações extraídas em (ROSKAM, 2001).

Modelagem Fuzzy

O SBRF que trata da modelagem aerodinâmica da aeronave Cessna 172 é apresentado a seguir, incluindo a definição das funções de pertinência, base de regras e superfície de resposta para os métodos Mamdani, Takagi-Sugeno de ordem 0, Kosko-Mizumoto, Larsen e PIA.

Vale ressaltar que o suporte e o número de subconjuntos fuzzy, bem como a escolha das funções de pertinência, são determinados com base em (ROSKAM, 2001) e também a partir da análise dos dados extraídos do simulador de voo X-Plane 11.

Para ambas as variáveis de entrada (α e δe), as funções de pertinência são definidas de acordo com a Fig. 5.6.

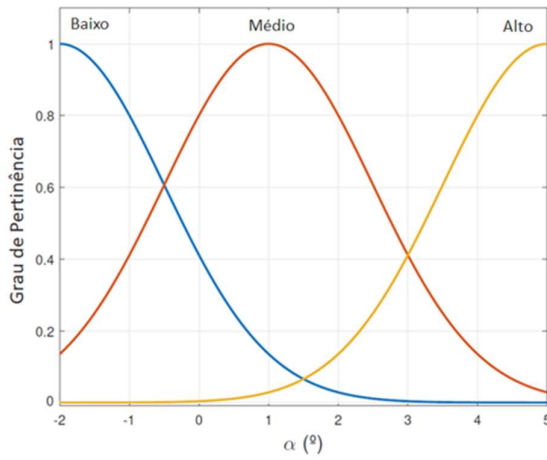
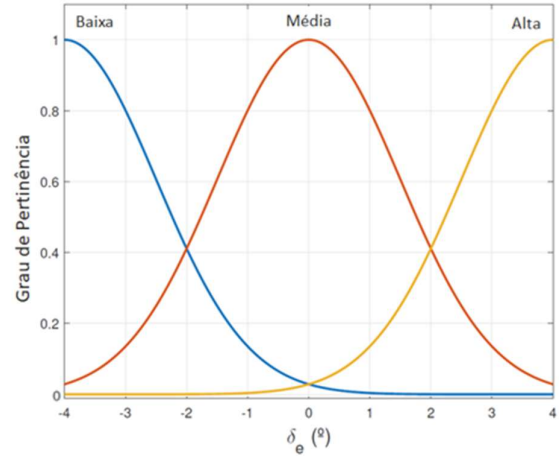
(a) Funções de pertinência de α .(b) Funções de pertinência de δe .

Figura 5.6: Funções de pertinência das entradas (autoria própria).

As funções de pertinência das variáveis de saída (C_D e C_L) para os métodos de inferência fuzzy Mamdani, Kosko-Mizumoto, Larsen e PIA são definidas de acordo com a Fig. 5.7. No método de Takagi-Sugeno, as funções de pertinência das variáveis de saída são *singleton*, cujos suportes correspondem à média de cada uma das funções de pertinência dos outros métodos de inferência fuzzy.

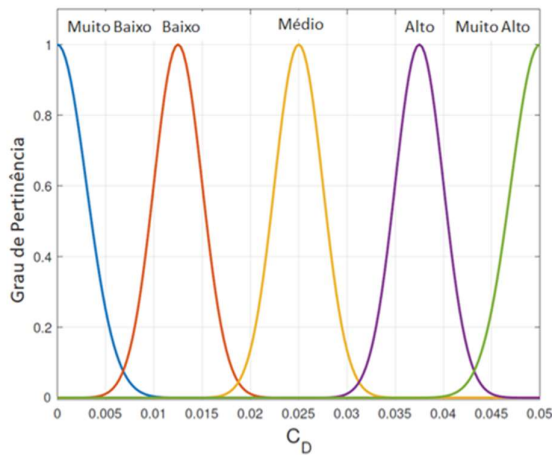
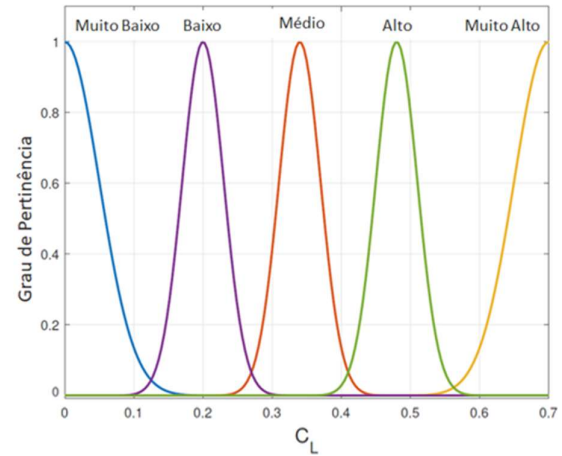
(a) Funções de pertinência de C_D .(b) Funções de pertinência de C_L .

Figura 5.7: Funções de pertinência das saídas (autoria própria).

A base de regras deste SBRF, apresentada na Tab. 5.5, é construída com base em (ROSKAM, 2001) e através do apoio de profissionais com experiência na área.

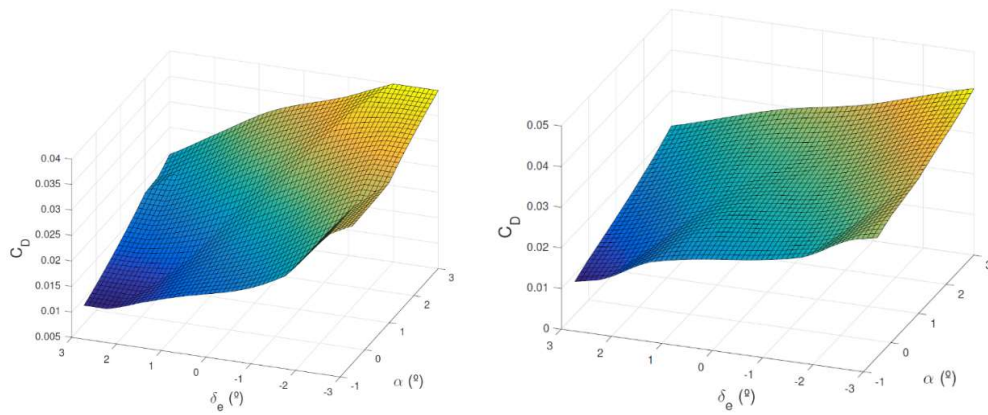
Tabela 5.5: Base de Regras.

Regras	Se	α	E	δe	Então	C_L	E	C_D
R_1		<i>baixo</i>		<i>baixa</i>		<i>médio</i>		<i>médio</i>
R_2		<i>médio</i>		<i>baixa</i>		<i>alto</i>		<i>alto</i>
R_3		<i>alto</i>		<i>baixa</i>		<i>muito alto</i>		<i>muito alto</i>
R_4		<i>baixo</i>		<i>média</i>		<i>baixo</i>		<i>baixo</i>
R_5		<i>médio</i>		<i>média</i>		<i>médio</i>		<i>médio</i>
R_6		<i>alto</i>		<i>média</i>		<i>alto</i>		<i>alto</i>
R_7		<i>baixo</i>		<i>alta</i>		<i>muito baixo</i>		<i>muito baixo</i>
R_8		<i>médio</i>		<i>alta</i>		<i>baixo</i>		<i>baixo</i>
R_9		<i>alto</i>		<i>alta</i>		<i>médio</i>		<i>médio</i>

As superfícies de resposta $C_D(\alpha, \delta e)$ para cada um dos métodos de inferência fuzzy são apresentadas nas Figs. 5.8 e 5.9. Além disso, as superfícies de resposta $C_L(\alpha, \delta e)$ para cada um dos métodos de inferência fuzzy são apresentadas nas Figs. 5.10 e 5.11.

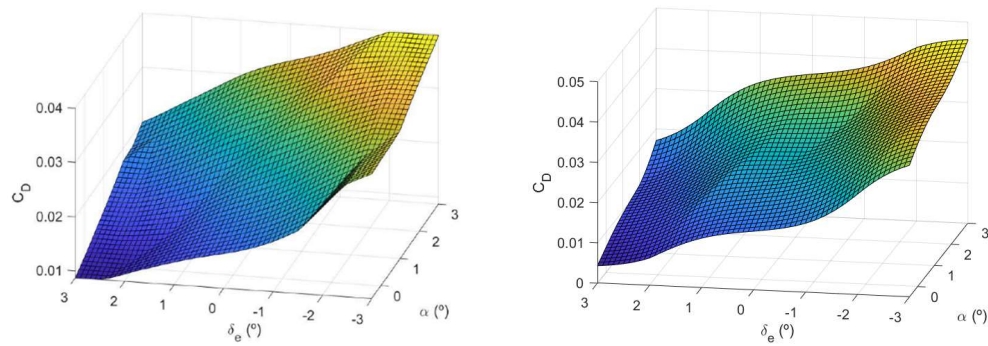
Pode-se notar que, para C_L e C_D , o método PIA garante consistência com a base de regras construída a priori, superfícies de resposta mais suaves que os outros métodos e maior semelhança com gráficos planares, de acordo com a Eq. (5.9).

A seguir, é apresentada uma análise numérica e estatística da ajustabilidade de um sistema fuzzy adaptativo aos dados numéricos-experimentais extraídos do simulador de voo X-Plane 11. Os resultados do método PIA de inferência fuzzy são comparados com os obtidos com os outros métodos.



(a) Método de inferência de Mamdani.

(b) Método de inferência de Takagi-Sugeno.



(c) Método de inferência de Kosko-Mizumoto.

(d) Método de inferência de Larsen.

Figura 5.8: Superfície de resposta C_D (α, δ_e) para os métodos de Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen (autoria própria).

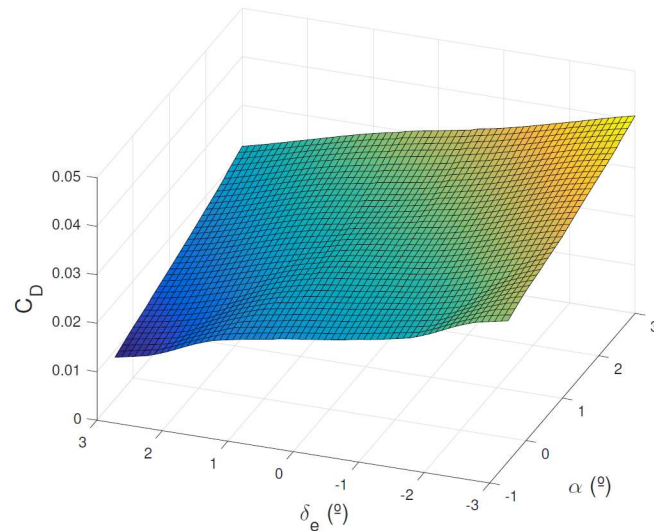
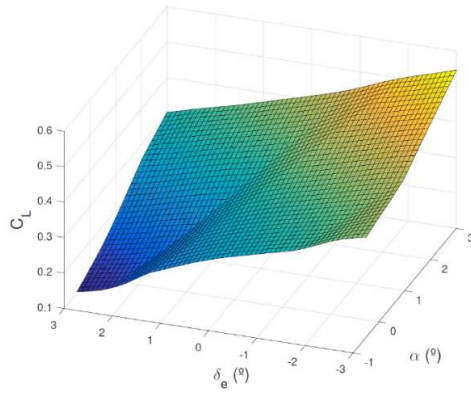
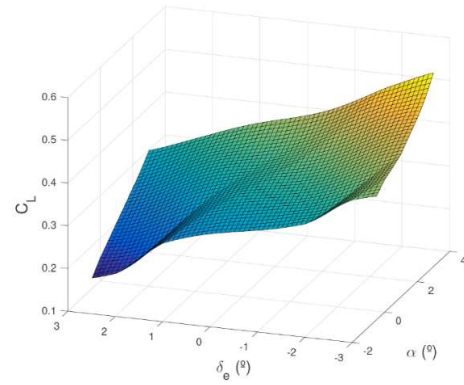


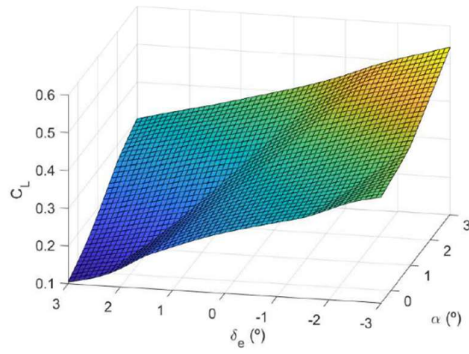
Figura 5.9: Superfície de resposta C_D (α, δ_e) para o método PIA (autoria própria).



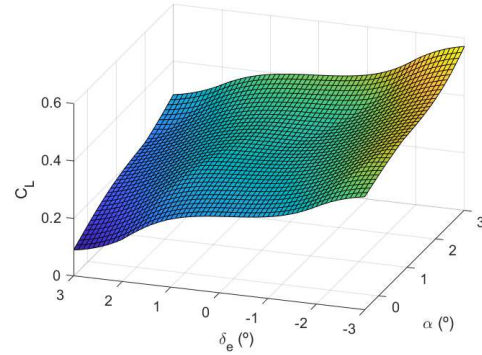
(a) Método de inferência de Mamdani.



(b) Método de inferência de Takagi-Sugeno.



(c) Método de inferência de Kosko-Mizumoto.



(d) Método de inferência de Larsen.

Figura 5.10: Superfície de resposta $C_L(\alpha, \delta_e)$ para os métodos de Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen (autoria própria).

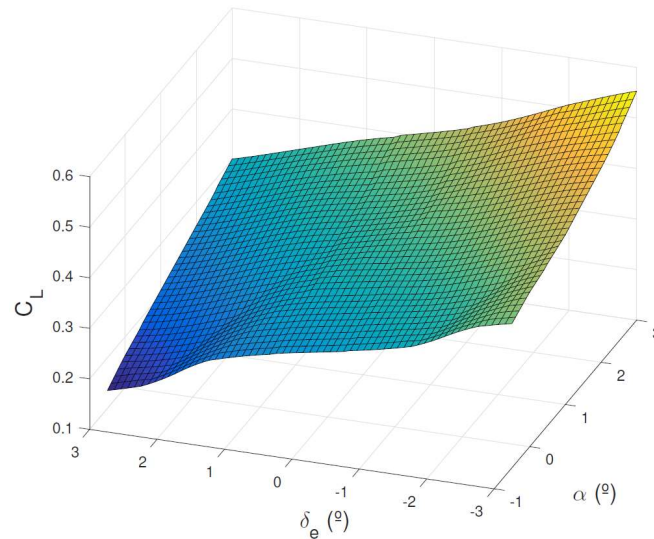


Figura 5.11: Superfície de resposta $C_L(\alpha, \delta_e)$ para o método PIA (autoria própria).

Análise Comparativa de Sistemas Fuzzy Adaptativos

Nesta etapa, um conjunto de dados, extraído de um voo virtual realizado no simulador de voo X-Plane11, é usado para analisar a capacidade de ajuste do PIA, e comparar a eficiência computacional desse método em relação aos métodos Mamdani, Takagi-Sugeno, Kosko-Mizumoto, Larsen e à abordagem híbrida HGFRD (BEMANI e AKBARZADEH, 2019).

Os sistemas fuzzy adaptativos são baseados no SBRF apresentado anteriormente nessa seção, ou seja, os mesmos possuem as três funções de pertinência gaussianas para cada uma das variáveis de entrada, Fig. 5.6, as mesmas cinco funções de pertinência para cada uma das variáveis de saída, Fig. 5.7, e também a base de regras fuzzy apresentada na Tab. 5.5. A única diferença é que os parâmetros que definem as funções de pertinência (média e desvio padrão) são adaptativos e ajustáveis por um método de otimização.

O método de otimização usado neste capítulo para ajustar os sistemas fuzzy aos dados numérico-experimentais é a evolução diferencial (STORN e PRICE, 1997). Os valores dos parâmetros utilizados na evolução diferencial são apresentados na Tab. 5.6.

Tabela 5.6: Parâmetros da Evolução Diferencial.

Parâmetro	Valor
Número de iterações	400
Número de vetores	150
Taxa de cruzamento	0,95
Fator de perturbação	0,4

O processo de otimização e os gráficos dos valores estimados de C_D e C_L , em azul, comparados a um conjunto de 300 dados extraídos durante 30 segundos (s) de simulação no X-Plane 11, em vermelho, são apresentados respectivamente nas Figs. 5.12, 5.13(a) e 5.13(b).

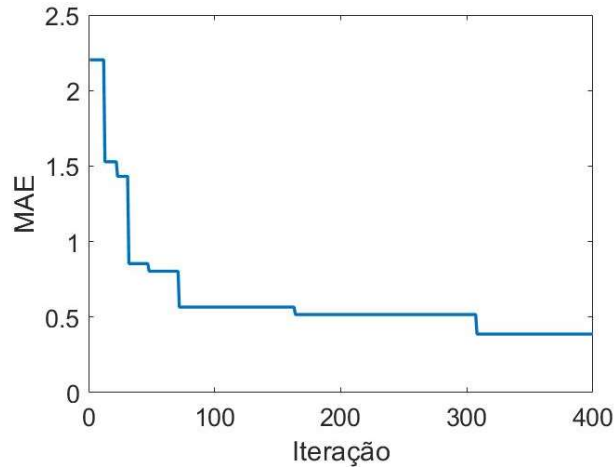
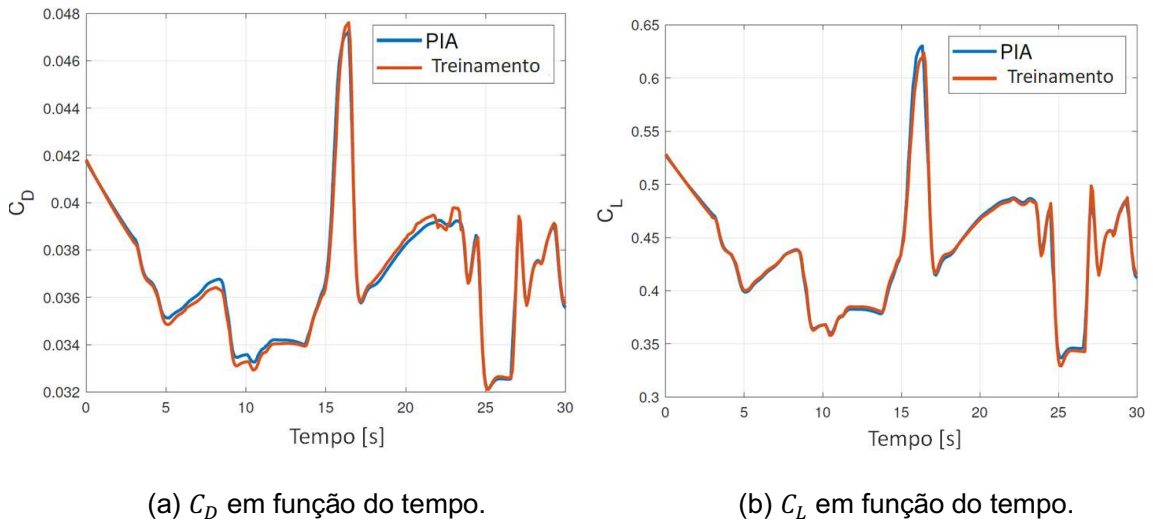


Figura 5.12: Processo de otimização do erro absoluto médio (MAE) da resposta do SBRF que utiliza o método PIA em função da iteração da evolução diferencial (autoria própria).



(a) C_D em função do tempo.

(b) C_L em função do tempo.

Figura 5.13: C_D e C_L em função do tempo com o método PIA (autoria própria).

Nas Figs. 5.13(a) e 5.13(b), pode-se verificar a capacidade de ajuste do método PIA; no entanto, para analisar e comparar quantitativamente o desempenho do PIA com as outras técnicas, é realizada uma análise estatística dos resultados obtidos por cada um dos métodos de inferência fuzzy.

O algoritmo de evolução diferencial é executado 30 vezes para cada método de inferência fuzzy, e a média, \bar{X} , e desvio padrão, S_D , do tempo de execução, Δt , e do erro absoluto médio da resposta, MAE (*Mean Absolute Error*), com relação aos dados numérico-experimentais de C_D e C_L extraídos no X -Plane 11, são apresentados respectivamente nas

Tabs. 5.7 e 5.8. Ressalta-se que todas as execuções do método de otimização ocorrem em uma máquina com processador intel i7 com clock de 4,5 GHz (execução realizada em apenas um núcleo do processador), memória RAM com 32 GB, disco rígido com capacidade de armazenamento de 1 TB, e memória SSD contendo 256 GB de espaço de armazenamento.

Tabela 5.7: Estatísticas dos Métodos de Inferência Fuzzy para C_D .

	<i>MAE</i>		Δt (s)	
	\bar{X}	S_D	\bar{X}	S_D
Mamdani	0,0383	0,0093	9052	3951
Takagi-Sugeno	0,0292	0,0024	45	13
Kosko-Mizumoto	0,0386	0,0022	5400	2055
Larsen	0,0398	0,0032	4213	1677
HGFRD	0,0280	0,0039	1752	34
PIA	0,0278	0,0022	597	63

Tabela 5.8: Estatísticas dos Métodos de Inferência Fuzzy para C_L .

	<i>MAE</i>		Δt (s)	
	\bar{X}	S_D	\bar{X}	S_D
Mamdani	0,5590	0,1020	6139	1436
Takagi-Sugeno	0,4285	0,0625	25	9
Kosko-Mizumoto	0,4687	0,1169	5859	1965
Larsen	0,5492	0,0814	5689	2840
HGFRD	0,3904	0,0453	2377	40
PIA	0,3883	0,0469	544	23

As estatísticas apresentadas nas Tabs. 5.7 e 5.8 indicam que o método PIA é o que mais reduz o erro absoluto médio em relação aos dados numérico-experimentais. Para verificar a superioridade do método PIA nesse critério, tanto para C_D quanto para C_L , são elaborados testes de hipóteses comparando o PIA com cada um dos outros métodos, cujas hipóteses nula, H_0 , e alternativa, H_a , (MONTGOMERY e RUNGER, 2002) são apresentadas a seguir:

- H_0 : O MAE do método em análise é igual ao MAE do método PIA;
- H_a : O MAE do método em análise é maior do que o MAE do método PIA.

Com 98,5% de nível de confiança, com base na distribuição t-Student aplicada a variáveis aleatórias heterocedásticas, o teste rejeitou a hipótese nula para os métodos de Mamdani, Takagi-Sugeno, Kosko-Mizumoto e Larsen, ou seja, para resolver esse problema, o método PIA é o que obtém o menor valor de MAE , e consequentemente a maior capacidade de ajuste aos dados numérico-experimentais. Embora o valor médio de MAE obtido pelo PIA tenha sido um pouco melhor que o obtido pelo HGFRD, não há elementos estatísticos que rejeitem a hipótese de desempenho numérico equivalente de ambas as técnicas.

Quanto ao tempo de execução de cada método, Takagi-Sugeno é o que possui o menor custo computacional, seguido pelo PIA, que por sua vez é mais rápido em termos computacionais do que os métodos de Mamdani, Kosko-Mizumoto, Larsen e HGFRD.

Os resultados sugerem a eficácia do método PIA, que combina capacidade de ajuste e baixo tempo de execução em relação a algumas técnicas, características típicas do método de Takagi-Sugeno, com a intuitividade de utilizar o mesmo processo de definição das funções de pertinência do método de Mamdani.

Após a explanação do método PIA, no próximo capítulo é apresentado o principal estudo de caso utilizado para a validação do método PIA de inferência fuzzy: controle neuro-fuzzy para aeronaves.

CAPÍTULO VI

CONTROLE NEURO-FUZZY PARA AERONAVES

Neste capítulo são apresentados os métodos utilizados de ajuste dos parâmetros dos controladores neuro-fuzzy, bem como os resultados obtidos via simulação e durante a operação do veículo para cada um dos métodos.

6.1. Arquitetura de Controle Neuro-Fuzzy

O modelo dinâmico de uma aeronave é composto por uma série de não-linearidades e acoplamentos. Existe, porém, a possibilidade de se trabalhar com razoável fidelidade à dinâmica real da aeronave a partir de seu modelo linearizado, que despreza sobretudo os acoplamentos entre as dinâmicas longitudinal e látero-direcional, mas sem deixar de considerar a aplicação de perturbações ao sistema (ROSKAM, 2001).

A partir do entendimento da aeronave como sendo submetida a dois comportamentos dinâmicos desacoplados entre si, controladores de voo são desenvolvidos levando em consideração essa natureza do sistema.

Esse é o caso do sistema de controle da empresa Micropilot, líder no desenvolvimento de controladores de voo para veículos aéreos não-tripulados. Na Fig. 3.12 é apresentada uma representação da arquitetura de controle longitudinal desenvolvida pela Micropilot, baseada

No controle longitudinal da aeronave, h é a altitude, θ é o ângulo de arfagem, q é a velocidade angular de arfagem, U é a velocidade longitudinal, w é a velocidade vertical, δ_T é a porcentagem relativa à aceleração máxima da aeronave, e δ_E é a deflexão do profundor.

Já no controle látero-direcional, ϕ é o ângulo de rolagem, ψ é o ângulo de guinada, p é a velocidade angular de rolagem, r é a velocidade angular de guinada, U continua sendo a velocidade longitudinal, v é a velocidade lateral, δ_A é a deflexão dos ailerons da aeronave, e por fim δ_R é a deflexão do leme.

Na arquitetura de controle, k indica a iteração atual, e *set* e *err* são respectivamente os indicativos dos sinais de referência (*setpoint*) e erro. Os blocos K e K/s são, nesta ordem, controladores proporcionais e integradores cujas respostas são somadas às saídas dos SBRFs. Todo o sistema de controle é composto por 6 SBRFs, cada um com suas respectivas entradas e saídas, e tendo o PIA como método de inferência fuzzy. As entradas e saídas de cada um dos SBRFs são apresentadas na Tab. 6.1.

Tabela 6.1: Entradas e Saídas dos Sistemas Baseados em Regras Fuzzy.

SBRF	$SBRF_1$	$SBRF_2$	$SBRF_3$	$SBRF_4$	$SBRF_5$	$SBRF_6$
Entradas	$h_{err,k}$	$\theta_{err,k}$ e U_k	$h_{err,k}$ e $U_{err,k}$	$\psi_{err,k}$	$\phi_{err,k}$ e U_k	$\psi_{err,k}$ e U_k
Saída	$\theta_{set,k}$	$\delta_{E1,k}$	δ_{Tk}	$\phi_{set,k}$	$\delta_{A1,k}$	$\delta_{R1,k}$

Os termos linguísticos e a base de regras de cada um dos SBRFs, obtidos a partir de Roskam (2001) e do conhecimento de especialistas na área de engenharia aeronáutica, são apresentados nas Tabs. 6.2 a 6.7.

$SBRF_1$:

- Termos linguísticos de $h_{err,k}$: *Baixo*, *Médio* e *Alto*;
- Termos linguísticos de $\theta_{set,k}$: *Baixo*, *Médio* e *Alto*.

Tabela 6.2: $SBRF_1$ - Base de Regras.

Regras	Se $h_{err,k}$	Então $\theta_{set,k}$
R_1	<i>Baixo</i>	<i>Baixo</i>
R_2	<i>Médio</i>	<i>Médio</i>
R_3	<i>Alto</i>	<i>Alto</i>

$SBRF_2$:

- Termos linguísticos de $\theta_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de U_k : *Baixa, Média e Alta*;
- Termos linguísticos de $\delta_{E1,k}$: *Baixíssima, Muito Baixa, Baixa, Média, Alta, Muito Alta, Altíssima*.

Tabela 6.3: $SBRF_2$ - Base de Regras.

Regras	Se	$\theta_{err,k}$	E	U_k	Então	$\delta_{E1,k}$
R_1		<i>baixo</i>		<i>baixa</i>		<i>Altíssima</i>
R_2		<i>médio</i>		<i>baixa</i>		<i>Média</i>
R_3		<i>alto</i>		<i>baixa</i>		<i>Baixíssima</i>
R_4		<i>baixo</i>		<i>média</i>		<i>Muito Alta</i>
R_5		<i>médio</i>		<i>média</i>		<i>Média</i>
R_6		<i>alto</i>		<i>média</i>		<i>Muito Baixa</i>
R_7		<i>baixo</i>		<i>alta</i>		<i>Alta</i>
R_8		<i>médio</i>		<i>alta</i>		<i>Média</i>
R_9		<i>alto</i>		<i>alta</i>		<i>Baixa</i>

 $SBRF_3$:

- Termos linguísticos de $h_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de $U_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de δ_{T_k} : *Muito Baixa, Baixa, Média, Alta e Muito Alta*.

Tabela 6.4: $SBRF_3$ - Base de Regras.

Regras	Se $h_{err,k}$	E $U_{err,k}$	Então δ_{T_k}
R_1	<i>baixo</i>	<i>baixo</i>	<i>Muito Baixa</i>
R_2	<i>médio</i>	<i>baixo</i>	<i>Baixa</i>
R_3	<i>alto</i>	<i>baixo</i>	<i>Média</i>
R_4	<i>baixo</i>	<i>médio</i>	<i>Baixa</i>
R_5	<i>médio</i>	<i>médio</i>	<i>Média</i>
R_6	<i>alto</i>	<i>médio</i>	<i>Alta</i>
R_7	<i>baixo</i>	<i>alto</i>	<i>Média</i>
R_8	<i>médio</i>	<i>alto</i>	<i>Alta</i>
R_9	<i>alto</i>	<i>alto</i>	<i>Muito Alta</i>

 $SBRF_4$:

- Termos linguísticos de $\psi_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de $\phi_{set,k}$: *Baixo, Médio e Alto*.

Tabela 6.5: $SBRF_4$ - Base de Regras.

Regras	Se $\psi_{err,k}$	Então $\phi_{set,k}$
R_1	<i>Baixo</i>	<i>Baixo</i>
R_2	<i>Médio</i>	<i>Médio</i>
R_3	<i>Alto</i>	<i>Alto</i>

 $SBRF_5$:

- Termos linguísticos de $\phi_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de U_k : *Baixa, Média e Alta*;
- Termos linguísticos de $\delta_{A1,k}$: *Baixíssima, Muito Baixa, Baixa, Média, Alta, Muito Alta, Altíssima*.

Tabela 6.6: $SBRF_5$ - Base de Regras.

Regras	Se	$\phi_{err,k}$	E	U_k	Então	$\delta_{A1,k}$
R_1		<i>baixo</i>		<i>baixa</i>		<i>Baixíssima</i>
R_2		<i>médio</i>		<i>baixa</i>		<i>Média</i>
R_3		<i>alto</i>		<i>baixa</i>		<i>Altíssima</i>
R_4		<i>baixo</i>		<i>média</i>		<i>Muito Baixa</i>
R_5		<i>médio</i>		<i>média</i>		<i>Média</i>
R_6		<i>alto</i>		<i>média</i>		<i>Muito Alta</i>
R_7		<i>baixo</i>		<i>alta</i>		<i>Baixa</i>
R_8		<i>médio</i>		<i>alta</i>		<i>Média</i>
R_9		<i>alto</i>		<i>alta</i>		<i>Alta</i>

 $SBRF_6$:

- Termos linguísticos de $\psi_{err,k}$: *Baixo, Médio e Alto*;
- Termos linguísticos de U_k : *Baixa, Média e Alta*;
- Termos linguísticos de $\delta_{R1,k}$: *Baixíssima, Muito Baixa, Baixa, Média, Alta, Muito Alta, Altíssima*.

Tabela 6.7: $SBRF_6$ - Base de Regras.

Regras	Se	$\psi_{err,k}$	E	U_k	Então	$\delta_{R1,k}$
R_1		<i>baixo</i>		<i>baixa</i>		<i>Altíssima</i>
R_2		<i>médio</i>		<i>baixa</i>		<i>Média</i>
R_3		<i>alto</i>		<i>baixa</i>		<i>Baixíssima</i>
R_4		<i>baixo</i>		<i>média</i>		<i>Muito Alta</i>
R_5		<i>médio</i>		<i>média</i>		<i>Média</i>

R_6	<i>alto</i>	<i>média</i>	<i>Muito Baixa</i>
R_7	<i>baixo</i>	<i>alta</i>	<i>Alta</i>
R_8	<i>médio</i>	<i>alta</i>	<i>Média</i>
R_9	<i>alto</i>	<i>alta</i>	<i>Baixa</i>

As funções de pertinência envolvidas neste sistema de controle são adotadas como sendo gaussianas, logo, cada uma das quais é definida por meio da adoção, no processo de sintonização do controlador, de valores para a média e desvio padrão, sendo este o tema da seção a seguir.

6.2. Sintonização e Estabilidade do Sistema de Controle

O processo de sintonização do controlador neuro-fuzzy passa pela simulação da operação da aeronave Cessna 172 submetida a perturbações ao longo do tempo de simulação.

Da mesma forma que em Thums, Torres e Palhares (2012), o processo de sintonização, ou seja, o cálculo dos valores dos parâmetros do controlador (neste caso, a média e desvio padrão associados a cada função de pertinência), é dividido em procedimentos, os quais são descritos a seguir.

1. Adota-se como estudo de caso a aeronave Cessna 172, em função de sua grande utilização e disponibilidade de informações a respeito de suas características dinâmicas.
2. O modelo utilizado é o exposto nas Eqs. (2.46) a (2.49), e os valores dos parâmetros, para a aeronave Cessna 172, são extraídos em Roskam (2001) e apresentados na Tab 6.8.

$$\begin{aligned}
& \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & -X_{\dot{q}} \\ -Z_{\dot{u}} & m - Z_{\dot{w}} & -Z_{\dot{q}} \\ -M_{\dot{u}} & -M_{\dot{w}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \\
& \begin{bmatrix} -X_u & mQ_0 - X_w & mW_0 - X_q \\ -mQ_0 - Z_u & -Z_w & -mU_0 - Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} mg \cos(\theta_0) \\ mg \sin(\theta_0) \cos(\phi_0) \\ 0 \end{bmatrix} \theta = \\
& \begin{bmatrix} X_{\delta_T} & X_{\delta_E} & X_{\delta_F} \\ Z_{\delta_T} & Z_{\delta_E} & Z_{\delta_F} \\ M_{\delta_T} & M_{\delta_E} & M_{\delta_F} \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_E \\ \delta_F \end{bmatrix}. \tag{2.46}
\end{aligned}$$

$$\dot{\theta} = q. \tag{2.47}$$

$$\begin{aligned}
& \begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{p}} & -Y_{\dot{r}} \\ -L_{\dot{v}} & I_x - L_{\dot{p}} & -I_{xz} - L_{\dot{r}} \\ -N_{\dot{v}} & -I_{xz} - N_{\dot{p}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} + \\
& \begin{bmatrix} -Y_v & -mW_0 - Y_p & mU_0 - Y_r \\ -L_v & -I_{xz}Q_0 - L_p & (I_z - I_y)Q_0 - L_r \\ -N_v & (I_y - I_x)Q_0 - N_p & I_{xz}Q_0 - N_r \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} -mg \cos(\theta_0) \cos(\phi_0) \\ 0 \\ 0 \end{bmatrix} \phi = \\
& \begin{bmatrix} Y_{\delta_A} & Y_{\delta_R} \\ L_{\delta_A} & L_{\delta_R} \\ N_{\delta_A} & N_{\delta_R} \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix}. \tag{2.48}
\end{aligned}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta_0) \\ 0 & 1/\cos(\theta_0) \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix}. \tag{2.49}$$

Tabela 6.8: Valores dos Parâmetros para a Aeronave Cessna 172 (ROSKAM, 2001).

m	1202 kg	$X_{\dot{u}}$	0	$X_{\dot{w}}$	0	$X_{\dot{q}}$	0
$Z_{\dot{u}}$	0	$Z_{\dot{w}}$	-10,41 Ns^2/m	$Z_{\dot{q}}$	0	$M_{\dot{u}}$	0
$M_{\dot{w}}$	-66,34 Ns^2/m	I_y	1825 kgm^2	$M_{\dot{q}}$	0	X_u	-22,69 Ns/m
Q_0	0	X_w	45,86 Ns/m	W_0	0	X_q	0

Z_u	$-151,4$ Ns/m	Z_w	-1095 Ns/m	U_0	$68,00$ m/s	Z_q	$-716,5$ Ns/rad
M_u	0	M_w	$-513,1$ Ns	M_q	-7732 Nms/rad	g	$-9,807$ m/s^2
θ_0	0	ϕ_0	0	X_{δ_T}	2030 N	X_{δ_E}	0
X_{δ_F}	0	Z_{δ_T}	0	Z_{δ_E}	-16508 N/rad	Z_{δ_F}	0
M_{δ_T}	0	M_{δ_E}	-64179 Nm/rad	M_{δ_F}	0	$Y_{\dot{v}}$	0
$Y_{\dot{p}}$	0	$Y_{\dot{r}}$	0	$L_{\dot{v}}$	0	I_x	1285 kgm^2
$L_{\dot{p}}$	0	I_{xz}	0	$L_{\dot{r}}$	0	$N_{\dot{v}}$	0
$N_{\dot{p}}$	0	I_z	2667 kgm^2	$N_{\dot{r}}$	0	Y_v	$-220,8$ Ns/m
Y_p	$-220,8$ Ns/rad	U_0	$68,00$ m/s	Y_r	$659,4$ Ns/rad	L_v	$-568,8$ Ns
L_p	-16360 Nms/rad	L_r	2697 Nms/rad	N_v	$361,7$ Ns	N_p	$-939,7$ Nms/rad
N_r	-3167 Nms/rad	Y_{δ_A}	0	Y_{δ_R}	7179 N/rad	L_{δ_A}	9644 Nm/rad
L_{δ_R}	6191 Nm/rad	N_{δ_A}	-9097 Nm/rad	N_{δ_R}	-27163 Nm/rad		

3. A otimização do sistema de controle é desenvolvida a partir do método da evolução diferencial, cujos parâmetros são apresentados na Tab. 6.9, que busca minimizar o erro médio absoluto MAE com relação ao comportamento (decaimento) imposto ao sistema de controle. No processo de busca por soluções que atendam ao requisito do sistema, impõe-se que as funções de pertinência que ocupam os limites dos intervalos de busca apresentem o mesmo valor de desvio padrão e tenham os valores de média iguais aos próprios limites de busca. Quanto às demais funções de pertinência, impõe-se igualdade de valor de desvio padrão e simetria com relação ao centro do intervalo de busca. O processo de otimização é subdividido em 4 etapas, descritas a seguir:

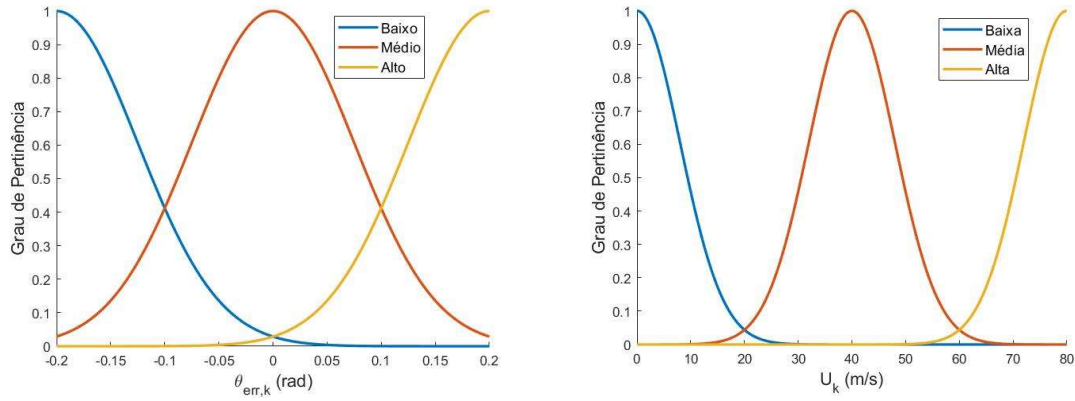
Tabela 6.9: Parâmetros da Evolução Diferencial.

Parâmetro	Valor
Número de iterações	100
Número de vetores	50
Taxa de cruzamento	0,95
Fator de perturbação	0,4

- a. Otimização dos parâmetros do $SBRF_2$ e consequentemente a definição de suas funções de pertinência, Figs. 6.3 e 6.4. Esta otimização se realiza a partir da simulação da operação da aeronave Cessna 172 submetida aos valores de referência, condições iniciais e perturbações apresentados na Eq. (6.1).

$$\left\{ \begin{array}{l} \theta_{set,k} = 0, 0 \leq t < 90 \text{ s} \\ \theta_k(t = 0) = 10^\circ \\ \theta_k(t = 15 \text{ s}) = -10^\circ \\ \theta_k(t = 30 \text{ s}) = 10^\circ \\ \theta_k(t = 45 \text{ s}) = -10^\circ \\ \theta_k(t = 60 \text{ s}) = 10^\circ \\ \theta_k(t = 75 \text{ s}) = -10^\circ \\ U_{set,k} = U_k = 30 \text{ m/s}, 0 \leq t < 30 \text{ s} \\ U_{set,k} = U_k = 68 \text{ m/s}, 30 \leq t < 60 \text{ s} \\ U_{set,k} = U_k = 100 \text{ m/s}, 60 \leq t < 90 \text{ s}. \end{array} \right. \quad (6.1)$$

- $SBRF_2$ – Entradas



(a) Funções de pertinência de $\theta_{err,k}$.

(b) Funções de pertinência de U_k .

Figura 6.3: Funções de pertinência das entradas do $SBRF_2$ (autoria própria).

- $SBRF_2$ – Saída

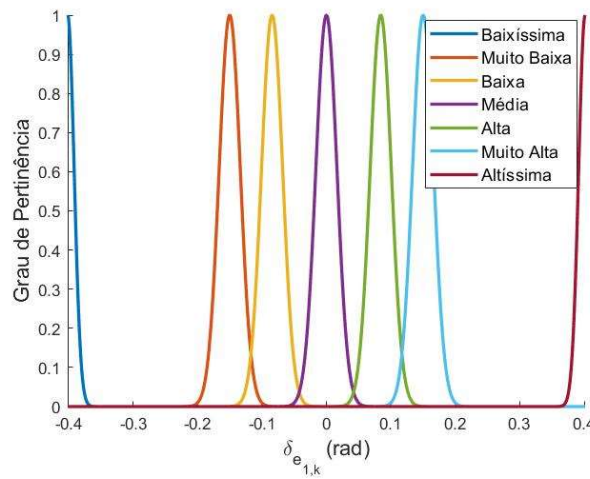


Figura 6.4: Funções de pertinência da saída do $SBRF_2$ (autoria própria).

- b. Otimização dos parâmetros de $SBRF_1$ e $SBRF_3$, e consequentemente a definição de suas funções de pertinência, Figs. 6.5 a 6.8. Esta otimização se realiza por meio da simulação da operação da aeronave Cessna 172 submetida aos valores de referência, condições iniciais e perturbações apresentados na Eq. (6.2).

$$\begin{cases} h_{set,k} = 50 \text{ m}, 0 \leq t < 30 \text{ s} \\ h_{set,k} = 0 \text{ m}, 30 \leq t < 100 \text{ s} \\ U_{set,k} = 68 \text{ m/s}, 0 \leq t < 30 \text{ s} \\ U_{set,k} = 73 \text{ m/s}, 30 \leq t < 50 \text{ s} \\ U_{set,k} = 68 \text{ m/s}, 50 \leq t < 100 \text{ s}. \end{cases} \quad (6.2)$$

- $SBRF_1$ – Entrada

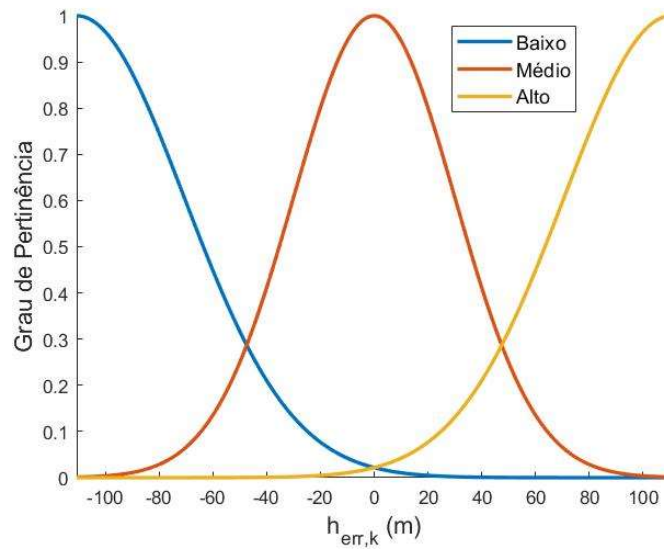


Figura 6.5: Funções de pertinência da entrada do $SBRF_1$ (autoria própria).

- $SBRF_1$ – Saída

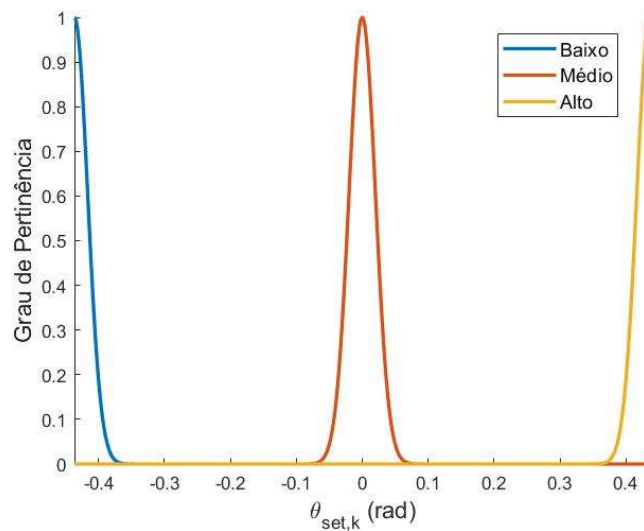
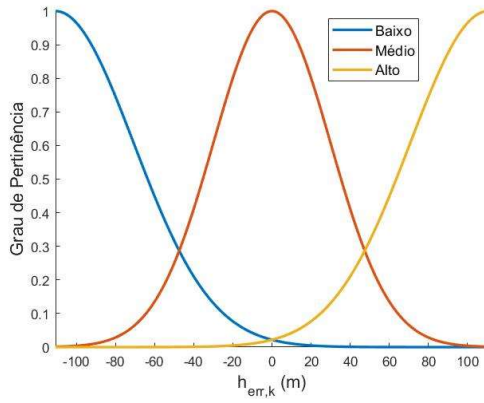
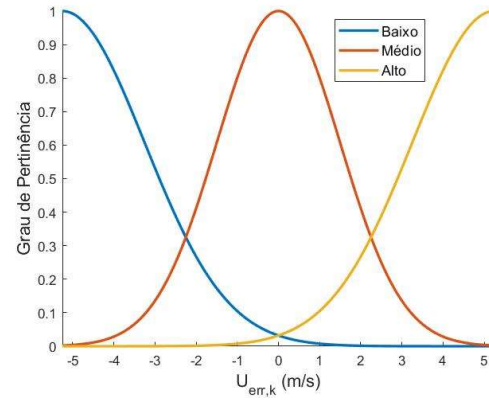


Figura 6.6: Funções de pertinência da saída do $SBRF_1$ (autoria própria).

- $SBRF_3$ – Entradas



(a) Funções de pertinência de $h_{err,k}$.



(b) Funções de pertinência de $U_{err,k}$.

Figura 6.7: Funções de pertinência das entradas do $SBRF_3$ (autoria própria).

- $SBRF_3$ – Saída

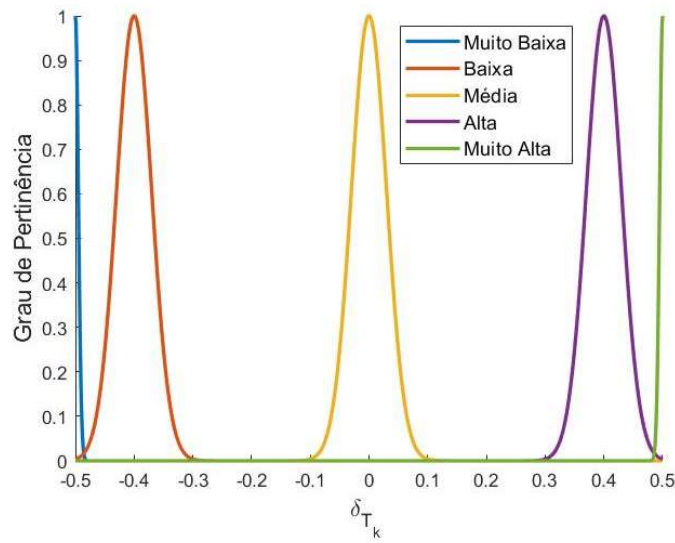
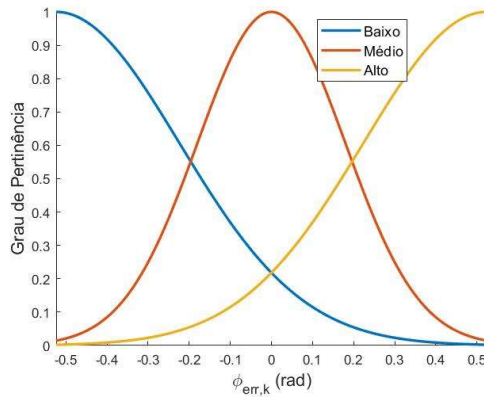


Figura 6.8: Funções de pertinência da saída do $SBRF_3$ (autoria própria).

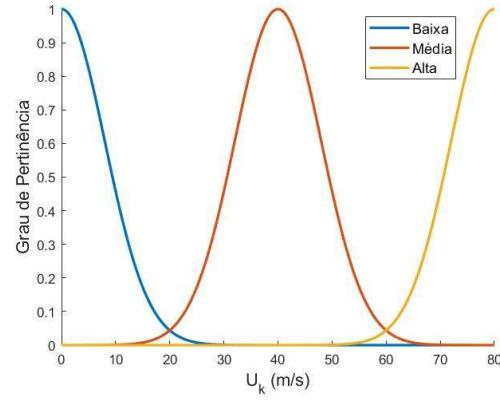
- c. Otimização dos parâmetros do $SBRF_5$ e assim a definição de suas funções de pertinência, Figs. 6.9 e 6.10. Esta otimização se dá a partir novamente da simulação da operação da aeronave Cessna 172 submetida aos valores de referência, condições iniciais e perturbações apresentados na Eq. (6.3).

$$\left\{ \begin{array}{l} \phi_{set,k} = 0, 0 \leq t < 90 \text{ s} \\ \phi_k(t = 0) = 10^\circ \\ \phi_k(t = 15 \text{ s}) = -10^\circ \\ \phi_k(t = 30 \text{ s}) = 10^\circ \\ \phi_k(t = 45 \text{ s}) = -10^\circ \\ \phi_k(t = 60 \text{ s}) = 10^\circ \\ \phi_k(t = 75 \text{ s}) = -10^\circ \\ U_{set,k} = U_k = 30 \text{ m/s}, 0 \leq t < 30 \text{ s} \\ U_{set,k} = U_k = 68 \text{ m/s}, 30 \leq t < 60 \text{ s} \\ U_{set,k} = U_k = 100 \text{ m/s}, 60 \leq t < 90 \text{ s}. \end{array} \right. \quad (6.3)$$

- $SBRF_5$ – Entradas



(a) Funções de pertinência de $\phi_{err,k}$.



(b) Funções de pertinência de U_k .

Figura 6.9: Funções de pertinência das entradas do $SBRF_5$ (autoria própria).

- $SBRF_5$ – Saída

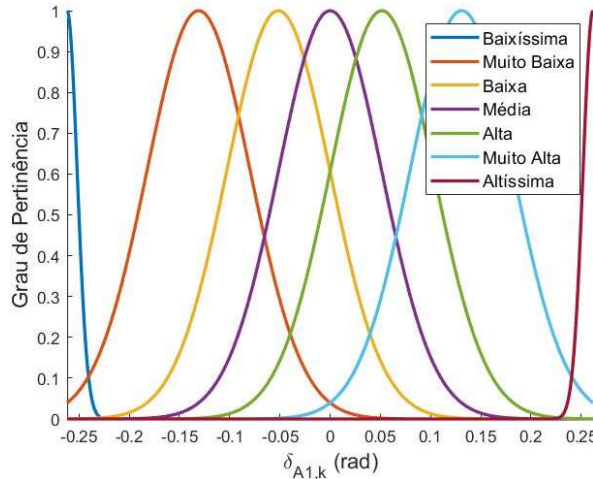


Figura 6.10: Funções de pertinência da saída do $SBRF_5$ (autoria própria).

- d. Por fim, a otimização dos parâmetros de $SBRF_4$ e $SBRF_6$, e logo a definição de suas funções de pertinência, Figs. 6.11 a 6.14, a partir da simulação da operação da aeronave Cessna 172 submetida aos valores de referência, condições iniciais e perturbações apresentados na Eq. (6.4).

$$\left\{ \begin{array}{l} \psi_{set,k} = 90^\circ, 0 \leq t < 50 \text{ s} \\ \psi_{set,k} = 0^\circ, 50 \leq t < 100 \text{ s} \\ \psi_{set,k} = 90^\circ, 100 \leq t < 150 \text{ s} \\ \psi_{set,k} = 0^\circ, 150 \leq t < 200 \text{ s} \\ \psi_{set,k} = 90^\circ, 200 \leq t < 250 \text{ s} \\ \psi_{set,k} = 0^\circ, 250 \leq t < 300 \text{ s} \\ U_{set,k} = U_k = 30 \text{ m/s}, 0 \leq t < 100 \text{ s} \\ U_{set,k} = U_k = 60 \text{ m/s}, 100 \leq t < 200 \text{ s} \\ U_{set,k} = U_k = 100 \text{ m/s}, 200 \leq t < 300 \text{ s}. \end{array} \right. \quad (6.4)$$

- $SBRF_4$ – Entrada

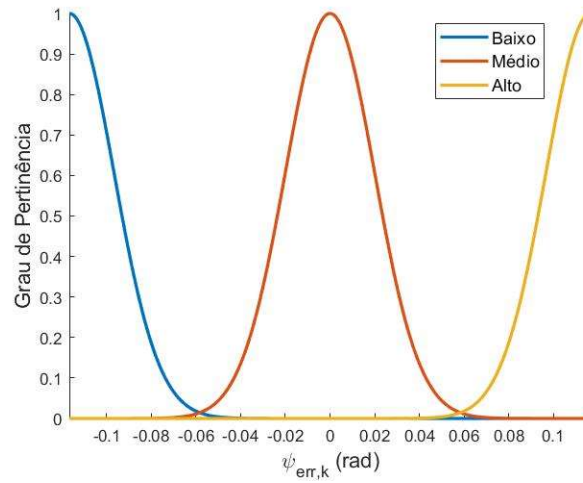


Figura 6.11: Funções de pertinência da entrada do $SBRF_4$ (autoria própria).

- $SBRF_4$ – Saída

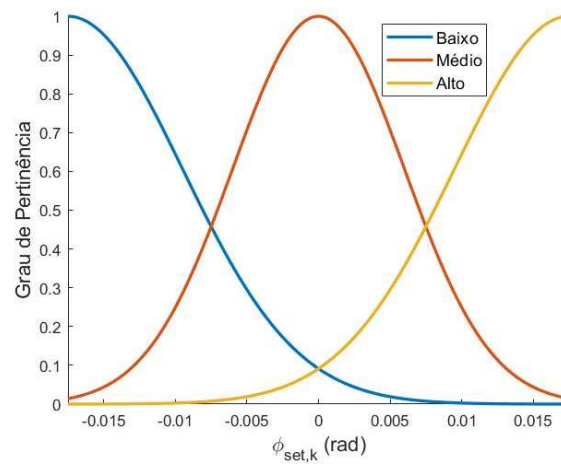
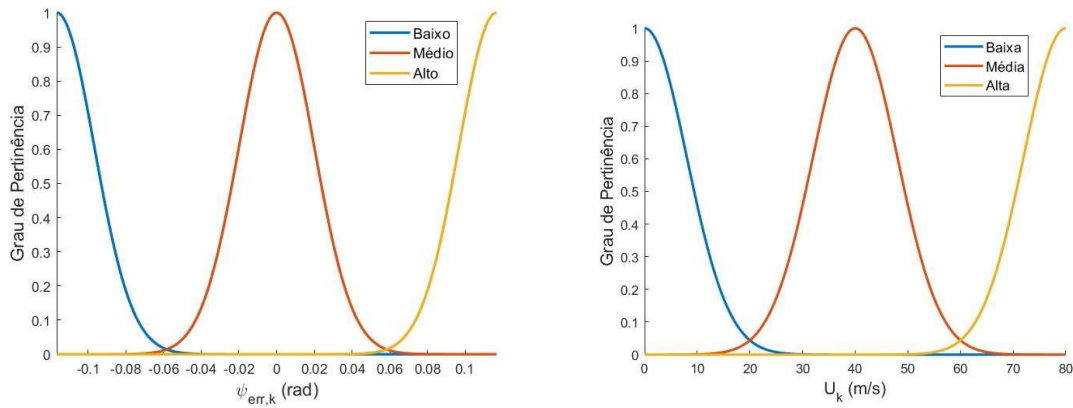


Figura 6.12: Funções de pertinência da saída do $SBRF_4$ (autoria própria).

- $SBRF_6$ – Entradas



(a) Funções de pertinência de $h_{err,k}$.

(b) Funções de pertinência de U_k .

Figura 6.13: Funções de pertinência das entradas do $SBRF_6$ (autoria própria).

- $SBRF_6$ – Saída

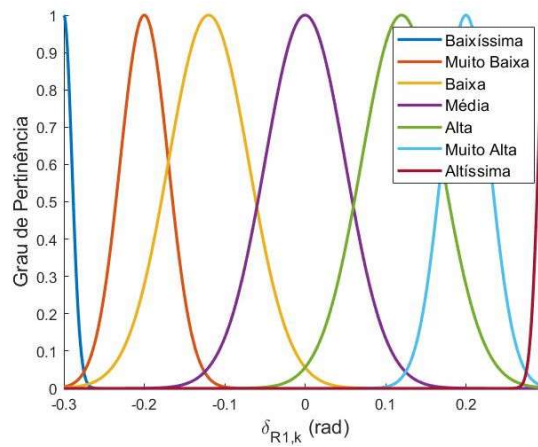


Figura 6.14: Funções de pertinência da saída do $SBRF_6$ (autoria própria).

O desempenho do sistema aeronáutico, controlado a partir da arquitetura de controle neuro-fuzzy proposta neste trabalho (PIA), é apresentado nas Figs. 6.15 a 6.19, assim como o desempenho dos controladores PID e neuro-fuzzy com o método de inferência de Takagi-Sugeno de ordem 0 (TS), sendo este último dotado da mesma arquitetura de controle e mesma técnica de otimização utilizada no controlador PIA. Nessas figuras, também são apresentados os gráficos dos sinais de referência set no tempo discreto k , e o comportamento (decaimento) considerado para a resposta, requisito este que é função das dimensões da aeronave e está relacionado ao conforto dos passageiros a bordo, sendo mais suave que o valor de referência.

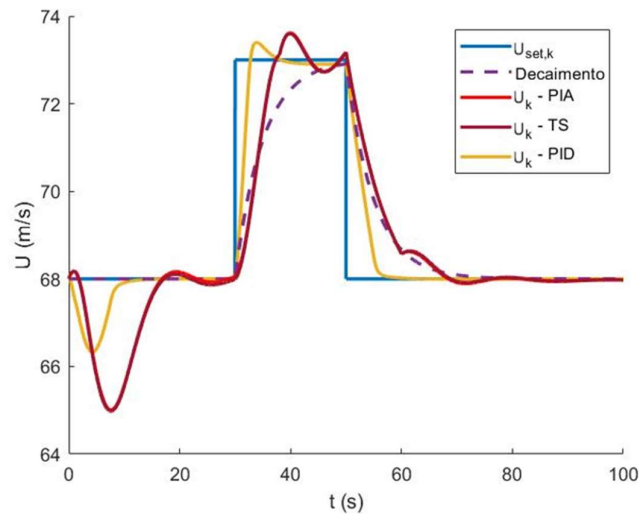


Figura 6.15: Velocidade da aeronave $U(m/s)$ em função do tempo de simulação $t(s)$ (autoria própria).

Na análise dos gráficos da Fig. 6.15, é possível verificar que a resposta do sistema com o controlador PID apresenta menor sobressinal se comparada às respostas dos demais controladores. A aderência da resposta da aeronave com relação ao decaimento desejado de 0 a 50 segundos é melhor com o controlador PID, porém, de 50 a 100 segundos, os controladores TS e PIA apresentam desempenho superior a este.

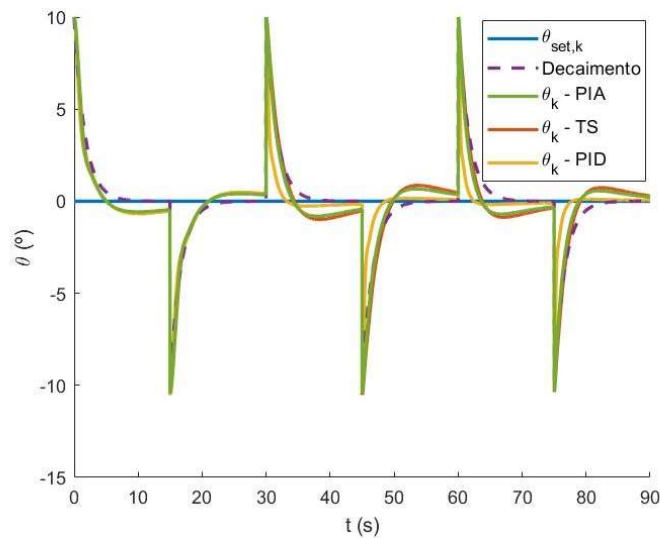


Figura 6.16: Ângulo de arfagem da aeronave $\theta(^{\circ})$ em função do tempo de simulação $t(s)$ (autoria própria).

No gráfico da Fig. 6.16, verifica-se o desempenho semelhante dos sistemas de controle, porém o controlador PIA, na maior parte da simulação, apresenta maior capacidade de seguir o decaimento pré-estipulado.

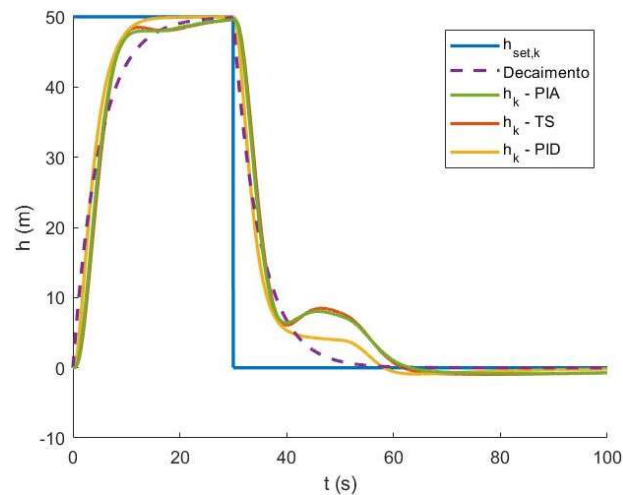


Figura 6.17: Altitude da aeronave em relação ao ponto de linearização $h(m)$ em função do tempo de simulação $t(s)$ (autoria própria).

Nos gráficos da Fig. 6.17, nota-se que a altitude da aeronave em simulação é controlada de forma semelhante utilizando quaisquer dos controladores testados. O controlador PID, porém, obteve desempenho mais satisfatório sobretudo no intervalo de tempo de 40 a 60 segundos.

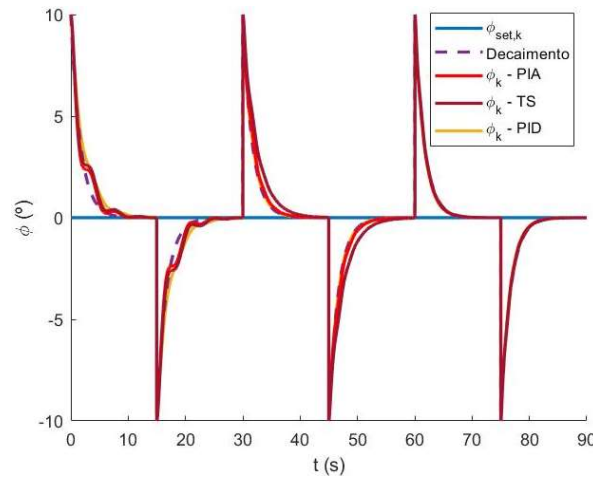


Figura 6.18: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo de simulação $t(s)$ (autoria própria).

Os controladores empregados em simulação, mais uma vez, desempenham-se de forma satisfatória e semelhante, conforme é possível verificar na Fig. 6.18. Nota-se, porém, que em praticamente toda a simulação, o controlador PIA apresenta maior capacidade de seguir o comportamento pré-estipulado.

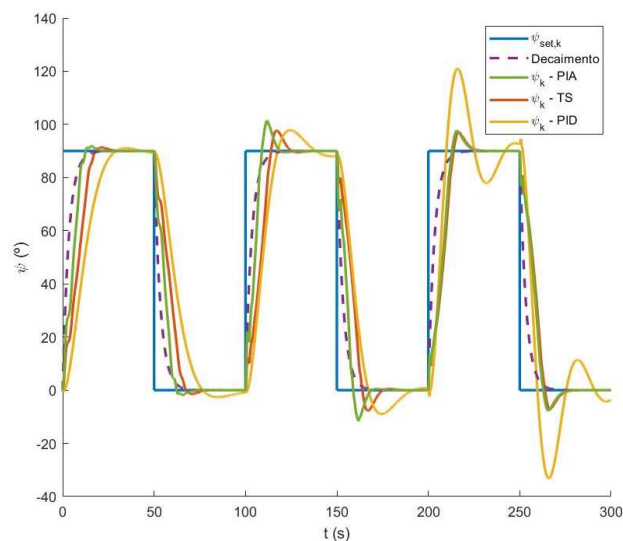


Figura 6.19: Ângulo de guinada da aeronave $\psi(^{\circ})$ em função do tempo de simulação $t(s)$ (autoria própria).

Por fim, ao se analisar os gráficos da Fig. 6.19 e levar em consideração a velocidade de cruzeiro adotada para a aeronave durante a simulação, a saber: 30 m/s no intervalo [0,100) s, 60 m/s no intervalo [100,200) s e 100 m/s no intervalo [200,300) s; é possível verificar novamente que todos os controladores conseguem manter o ângulo de guinada do Cessna 172 próximo ao comportamento desejado, sobretudo o controlador PIA. Ressalta-se também que este controlador, se comparado ao PID, mostra ser consideravelmente menos suscetível aos efeitos do aumento da velocidade de cruzeiro.

As ações de controle geradas pelo sistema neuro-fuzzy que lança mão do método PIA de inferência fuzzy, e que são responsáveis pelas respostas anteriormente apresentadas, são compostas pela deflexão do profundor δE , taxa de aceleração δT , deflexão do aileron δA e deflexão do leme δR , e são exibidas respectivamente nas Figs. 6.20(a), 6.20(b), 6.21(a) e 6.21(b).

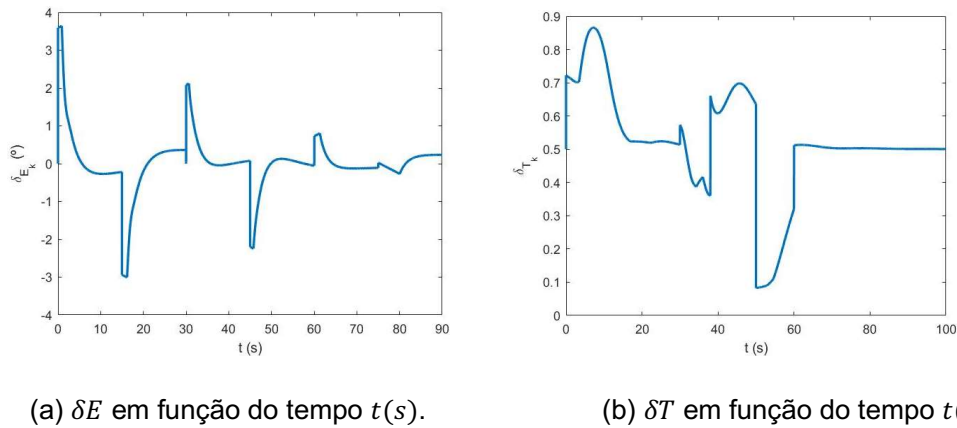


Figura 6.20: Ações de controle associadas à dinâmica longitudinal da aeronave (autoria própria).

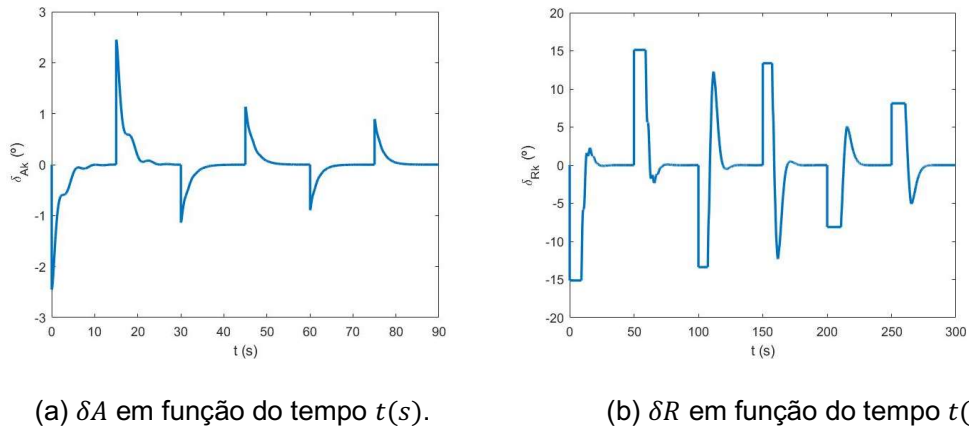


Figura 6.21: Ações de controle associadas à dinâmica látero-direcional da aeronave (autoria própria).

Através dos gráficos apresentados, é possível verificar que as ações de controle tendem a zero à medida em que o sistema se aproxima de seus valores de referência, com exceção à taxa de aceleração, responsável por manter o equilíbrio dinâmico da aeronave em um voo reto e nivelado, que oscila em torno de 0,5 (50% de carga do propulsor).

Diante dos resultados expostos graficamente nas Figs. 6.15 a 6.19, é possível verificar a capacidade de todas as técnicas de controlar efetivamente a aeronave Cessna 172. Em função do semelhante desempenho dos sistemas de controle, apresentam-se na Tab. 6.10, para cada uma das metodologias testadas, os dados relativos ao erro absoluto médio (MAE) da resposta do sistema controlado com relação ao decaimento de referência.

Tabela 6.10: Erro MAE da Resposta do Sistema Controlado.

Controlador Variável	PIA	TS	PID
$U(m/s)$	0,4196	0,4234	0,3748
$\theta(^{\circ})$	0,5858	0,6320	0,7313
$h(m)$	2,2105	2,3969	1,5356
$\phi(^{\circ})$	0,1357	0,2623	0,1588
$\psi(^{\circ})$	5,1227	8,2976	14,0976

É possível concluir, dentro das condições em teste e por meio dos resultados expostos na Tab. 6.10, que o controlador neuro-fuzzy que utiliza o método PIA de inferência fuzzy apresenta o melhor desempenho geral dentre as metodologias testadas, tendo obtido resultados superiores às demais técnicas em 3 das 5 variáveis de saída do sistema. Esse resultado indica que o método PIA se mostra, também, eficaz como estrutura matemática de um sistema de controle.

No intuito de verificar a estabilidade do sistema de controle sintonizado, as Eqs. (2.46), (2.47), (2.48) e (2.49) são aglutinadas em uma única equação matricial na forma $M\dot{x} + Kx = Du$, Eq. (6.5), em que x é o estado do sistema, u é a ação de controle aplicada ao mesmo e M , K e D são as matrizes que armazenam os coeficientes aerodinâmicos da aeronave:

$$\begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & -X_{\dot{q}} & 0 & 0 & 0 & 0 & 0 & 0 \\ -Z_{\dot{u}} & m - Z_{\dot{w}} & -Z_{\dot{q}} & 0 & 0 & 0 & 0 & 0 & 0 \\ -M_{\dot{u}} & -M_{\dot{w}} & I_y - M_{\dot{q}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m - Y_{\dot{v}} & -Y_{\dot{p}} & -Y_{\dot{r}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -L_{\dot{v}} & I_x - L_{\dot{p}} & -I_{xz} - L_{\dot{r}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -N_{\dot{v}} & -I_{xz} - N_{\dot{p}} & I_z - N_{\dot{r}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} +$$

$$\begin{bmatrix} -X_u & mQ_0 - X_w & mW_0 - X_q & mg\cos(\theta_0) & 0 & 0 & 0 & 0 & 0 \\ -mQ_0 - Z_u & -Z_w & -mU_0 - Z_q & mgsin(\theta_0)\cos(\phi_0) & 0 & 0 & 0 & 0 & 0 \\ -M_u & -M_w & -M_q & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -Y_v & -mW_0 - Y_p & mU_0 - Y_r & -mg\cos(\theta_0)\cos(\phi_0) & 0 \\ 0 & 0 & 0 & 0 & -L_v & -I_{xz}Q_0 - L_p & (I_z - I_y)Q_0 - L_r & 0 & 0 \\ 0 & 0 & 0 & 0 & -N_v & (I_y - I_x)Q_0 - N_p & I_{xz}Q_0 - N_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} =$$

$$\begin{bmatrix} X_{\delta_T} & X_{\delta_E} & X_{\delta_F} & 0 & 0 \\ Z_{\delta_T} & Z_{\delta_E} & Z_{\delta_F} & 0 & 0 \\ M_{\delta_T} & M_{\delta_E} & M_{\delta_F} & 0 & 0 \\ 0 & 0 & 0 & Y_{\delta_A} & Y_{\delta_R} \\ 0 & 0 & 0 & L_{\delta_A} & L_{\delta_R} \\ 0 & 0 & 0 & N_{\delta_A} & N_{\delta_R} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_T \\ \delta_E \\ \delta_F \\ \delta_A \\ \delta_R \end{bmatrix}. \quad (6.5)$$

Ao se multiplicar a Eq. (6.5) por M^{-1} e isolar \dot{x} , é possível reescrever a dinâmica do sistema no formato disposto na Eq. (2.6), em que $f(x)$ e $b(x)$ são funções que descrevem a dinâmica do sistema, e $x(t_0)$ é o estado inicial no tempo t_0 .

$$\dot{x}(t) = f(x) + b(x)u(x), \quad x(t_0) = x_o. \quad (2.6)$$

Define-se a função de Lyapunov $V(x) = x^T Px$, tomando P como sendo uma matriz identidade, e em seguida determina-se a derivada da função de Lyapunov \dot{V} , Eq. (2.7):

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T Px + x^T P \dot{x} = \\ \dot{V}(x) &= (f(x) + b(x)u(x))^T Px + x^T P(f(x) + b(x)u(x)) = \\ \dot{V}(x) &= F(x) + B(x), \end{aligned} \quad (2.7)$$

em que:

$$F(x) = f(x)^T Px + x^T P f(x), \quad B(x) = u^T(x) b^T(x) Px + x^T P b(x) u(x). \quad (2.8)$$

Se $\dot{V}(x) < 0 \forall x$, tem-se que o sistema em malha fechada, composto pelo controlador e pelo processo não-linear, será localmente assintoticamente estável no sentido de Lyapunov na origem x_0 .

Em função da complexidade matemática do modelo e do controlador neuro-fuzzy desenvolvido neste trabalho, abre-se mão de uma avaliação analítica, e realiza-se então a avaliação numérica da derivada da função da Lyapunov \dot{V} dentro dos limites pré-definidos para as variáveis do processo. Esses limites são estabelecidos de modo a contemplar o envelope de voo (limites operacionais) da aeronave e são apresentados na Tab. 6.11. Ressalta-se que os intervalos de avaliação de \dot{V} são discretizados, gerando um total de 1679616 avaliações, Fig. 6.22, o que é um indicativo de confiabilidade na inferência da estabilidade do sistema controlado.

Tabela 6.11: Intervalo de Avaliação da Função de Lyapunov \dot{V} .

Variável	Limite Inferior	Limite superior
$u(m/s)$	-25	25
$v(m/s)$	-25	25
$w(m/s)$	-25	25
$p(rad/s)$	-0,05	0,05
$q(rad/s)$	-0,05	0,05
$r(rad/s)$	-0,05	0,05
$\phi(^{\circ})$	-8	8
$\theta(^{\circ})$	-8	8

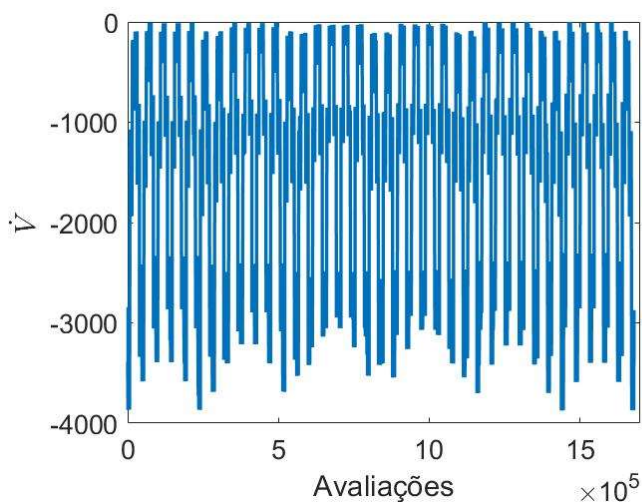


Figura 6.22: Avaliação numérica da derivada da função de Lyapunov \dot{V} (autoria própria).

É possível verificar através do gráfico apresentado anteriormente que o valor da derivada da função de Lyapunov para todas as condições testadas é negativo, sendo o maior valor observado igual a $-0,3634$, o que é um forte indício numérico de que o sistema controlado em malha fechada em questão é localmente assintoticamente estável no sentido de Lyapunov na origem x_0 .

Após a sintonização e análise de estabilidade do sistema aeronáutico controlado através do controlador neuro-fuzzy associado ao método PIA de inferência fuzzy, na próxima seção é apresentada a aplicação da técnica desenvolvida no controle de aeronave em operações críticas de voo.

6.3. *Software-in-the-Loop* – Operações Críticas de Voo

Esta seção apresenta o desempenho do sistema de controle aplicado a operações críticas de voo, porém, para viabilizar o *software-in-loop* realizado entre o MATLAB e o X-Plane 11, é necessário o desenvolvimento de uma interface de comunicação entre os dois softwares, assunto a ser abordado na etapa a seguir.

Comunicação UDP entre MATLAB e X-Plane 11

O X-Plane 11 é um simulador de voo produzido pela empresa Laminar Research (FIGUEIREDO e SAOTOME, 2012) caracterizado pela sua versatilidade e fidedignidade. Conta com modelos de aeronaves comerciais e militares, e um cenário virtual que abrange aeroportos e regiões do mundo todo. Além disso, o X-Plane 11 possibilita a reprodução de operações críticas de voo, relacionadas tanto ao funcionamento da aeronave, quanto às condições meteorológicas. Durante a simulação, é possível submeter a aeronave a uma série de avarias como falhas nos motores, sensores, instrumentos de navegação, controles, trens de pouso, dentre outros componentes. Da mesma forma também é possível alterar, a qualquer momento durante a simulação, as condições climáticas de forma a inserir perturbações como rajadas de vento, zonas de turbulência, chuvas fortes, dentre outras condições.

A agência americana FAA e a Agência Nacional de Aviação Civil (ANAC) reconhecem o X-Plane 11 como um software de base para simulações utilizadas no treinamento de pilotos profissionais, uma vez que este é capaz de reproduzir com alta fidelidade o ambiente de voo e a resposta da aeronave às ações de controle aplicadas pelo piloto. Além disso, o simulador possui ferramentas que permitem que o desempenho do piloto seja avaliado após o encerramento das simulações (ALMEIDA e CORREA, 2017).

Há na literatura trabalhos que propõem a utilização do Simulink e do X-Plane 11 na realização de simulações do tipo *software-in-the-loop* (BITTAR, 2011; FIGUEIREDO e SAOTOME, 2012; BITTAR *et al.*, 2014). No entanto, a literatura carece de trabalhos que apresentem a elaboração de um código puramente desenvolvido em linguagem de programação, e não em programação gráfica como o Simulink, o que garantiria que técnicas de controle mais complexas pudessem ser implementadas de forma mais direta e computacionalmente eficiente. Assim, este trabalho propõe o desenvolvimento de um algoritmo, através do uso do MATLAB, que possibilite o estabelecimento de uma comunicação entre este e o simulador de voo X-Plane 11, baseada no protocolo UDP (*User Datagram Protocol*).

Durante as simulações realizadas no X-Plane 11, são gerados dados em tempo real referentes à atitude, à localização e às ações de controle da aeronave. Caso o simulador seja configurado adequadamente, estes dados podem ser enviados através de uma rede UDP, a um IP (*Internet Protocol*) e a uma porta especificados, o que permite que outros softwares possam ter acesso aos dados gerados durante a simulação. Além disso, pacotes de comandos podem ser endereçados ao X-Plane 11 através da rede UDP estabelecida, de

forma que a aeronave possa ser controlada através da atuação sobre as superfícies de controle e sobre o sistema de propulsão, configurando assim uma simulação do *tipo software-in-the-loop*.

Para tratar os dados gerados durante a simulação, é necessário fazer com que o X-Plane 11 os envie para algum outro software que fique responsável pelo seu processamento e armazenamento. Neste trabalho, a análise dos dados é realizada através da utilização do MATLAB, e a comunicação entre este e o simulador é feita por meio de uma conexão baseada no protocolo UDP. Por fim, é importante destacar que o simulador e o algoritmo desenvolvido no MATLAB devem ser executados simultaneamente e no mesmo computador.

Para que a comunicação entre o MATLAB e o X-Plane 11 seja estabelecida da forma adequada, é necessário antes disso realizar um estudo acerca do simulador, e de como este constrói os pacotes de dados que são enviados através da rede. Este estudo é essencial para que o algoritmo aqui proposto seja desenvolvido, uma vez que o código deve levar em conta a estrutura dos pacotes que recebe do X-Plane 11, visto que é responsável pelo tratamento dos dados contidos nestes pacotes, e deve ser capaz de extrair deles as informações referentes à simulação.

Basicamente, as mensagens enviadas pelo X-Plane podem conter quaisquer informações geradas durante a simulação. Estas informações normalmente estão relacionadas ao estado atual da aeronave e dizem respeito a variáveis como por exemplo as velocidades lineares, as velocidades angulares, as acelerações, a deflexão das superfícies de comando, os ângulos de Euler, as coordenadas geográficas da aeronave, entre outras (BITTAR, 2011; BITTAR *et al.*, 2014). Desta forma, é necessário escolher quais variáveis estarão por fim contidas na mensagem enviada pelo X-Plane11, e esta configuração é realizada no menu *Settings*, na aba *Data Output*, Fig. 6.23.

Index	Data to Output
0	Frame rate
1	Times
2	Sim stats
3	Speeds
4	Mach, VVI, g-load
5	Weather
6	Aircraft atmosphere
7	System pressures
8	Joystick aileron/elevator/rudder
9	Other flight controls
10	Artificial stability
11	Flight controls aileron/elevator/rudder
12	Wing sweep & thrust vectoring
13	Trim, flap, slats, & speedbrakes
14	Gear & brakes
15	Angular moments
16	Angular velocities
17	Pitch, roll, & headings

Figura 6.23: Variáveis enviadas pelo X-Plane (IASBECK, PEREIRA e SANCHES, 2020).

Como o controle dos ângulos de Euler da aeronave é realizado durante a simulação, é preciso que o MATLAB receba do simulador pacotes que contenham informações acerca da atitude do avião. Assim sendo, o conjunto de variáveis selecionado na aba *Data Output* do X-Plane 11 é o de índice 17 (*Pitch, roll & headings*). Uma vez selecionado apenas este conjunto de variáveis, as mensagens enviadas pelo X-Plane 11 devem conter, em graus, o ângulo de arfagem (θ), o ângulo de rolagem (ϕ), e a diferença entre a direção para a qual aponta o avião e o norte (ψ ou *headings*).

As mensagens enviadas pelo X-Plane 11 são compostas por uma série de *bytes* e seguem a seguinte estrutura (LEWIS, 2009):

- Cabeçalho: 5 *bytes*;
- Índice correspondente ao conjunto de dados contido na mensagem: 4 *bytes*;
- Dados da mensagem: 32 *bytes* (8 números do tipo *float* com 4 *bytes* cada).

O cabeçalho da mensagem enviada pelo X-Plane possui cinco *bytes* e cada um deles representa um caractere. As mensagens enviadas pelo simulador têm seu cabeçalho composto pelas letras D, A, T e A, que indicam que a mensagem se trata de um pacote de dados, e além delas o caractere *null*. No entanto, como a mensagem enviada pelo X-Plane 11 é um conjunto de valores numéricos, sendo cada *byte* um valor que varia entre 0 e 255, serão observados como seus 5 primeiros *bytes* os números [68 65 84 65 0], correspondentes

na tabela ASCII (*American Standard Code for Information Interchange*) aos caracteres [D A T A null] (ASCIITABLE, 2010). No momento em que o MATLAB envia comandos ao simulador, a mesma estrutura é considerada na construção das mensagens.

Logo após o cabeçalho, encontra-se o índice correspondente ao conjunto de dados contido na mensagem. O mesmo possui comprimento de quatro *bytes*. Neste trabalho foi escolhido inicialmente apenas um conjunto de variáveis: o de índice 17. Assim sendo, é possível observar logo após o cabeçalho o conjunto de *bytes* [0 0 0 17].

Por fim, os valores dos ângulos θ , ϕ e ψ (variáveis relativas ao conjunto de índice 17, discutido no parágrafo anterior) encontram-se ao fim da mensagem e são os dados que de fato devem ser processados pelo algoritmo responsável pelo controle da aeronave. Cada pacote construído pelo X-Plane 11 pode conter até oito variáveis, sendo cada uma delas um número real do tipo *float* (*single precision*) composto por 4 *bytes* (32 *bits*). O primeiro *bit* indica o sinal s do número real representado, os próximos oito o expoente e , e os demais a mantissa m . Para que os vetores de *bytes* recebidos possam ser convertidos em números reais, a inversão destes se faz necessária, já que neste caso o computador utilizado implementa a arquitetura de representação de dados *little endian*.

Uma vez realizada a inversão dos *bytes*, os mesmos devem ser convertidos em suas correspondentes cadeias binárias que devem por fim ser concatenadas. Neste caso, como a conversão se dá a partir de vetores de 4 *bytes*, o valor binário gerado deverá possuir, por fim, 32 *bits* e a partir deste devem ser obtidos a mantissa m_{23-i} , $i = 1, \dots, 23$, o expoente e , e o *bit* de sinal s (STALLINGS, 2003). Para calcular o número real (ou *single*) correspondente aos dados recebidos, converte-se o vetor de *bits* que compõe o expoente em um número decimal, e_{10} , e em seguida utilizam-se as Eqs. (6.6) a (6.9) (LEWIS, 2009):

$$A = (-1)^s, \quad (6.6)$$

$$B = \left(1 + \sum_{i=1}^{23} m_{23-i} 2^{-i}\right), \quad (6.7)$$

$$C = 2^{e_{10}-12}, \quad (6.8)$$

$$single = ABC. \quad (6.9)$$

A conversão de um vetor de quatro *bytes* enviados no X-Plane em um número real, realizada a partir da aplicação das Eqs. (6.6) a (6.9), é implementada no software MATLAB por meio das funções a serem apresentadas nesta seção.

Caso vários conjuntos de variáveis sejam selecionados pelo usuário, todos eles deverão estar contidos na mensagem encaminhada pelo X-Plane. Assim sendo, a estrutura de um pacote que contenha mais de um conjunto de variáveis é mostrada a seguir.

- Cabeçalho: 5 *bytes*;
- Índice correspondente ao 1º conjunto de dados contido na mensagem: 4 *bytes*;
- 1º conjunto de dados da mensagem: 32 *bytes* (8 números do tipo *float* com 4 *bytes* cada);
- Índice correspondente ao 2º conjunto de dados contido na mensagem: 4 *bytes*;
- 2º conjunto de dados da mensagem: 32 *bytes* (8 números do tipo *float* com 4 *bytes* cada);
- E assim sucessivamente ...

À medida em que os dados são gerados pelo simulador, é interessante que eles sejam não somente enviados via UDP, mas também salvos em um arquivo e mostrados na tela durante a execução do algoritmo aqui desenvolvido, de forma que seja possível confirmar, tanto em tempo real quanto após o fim da simulação, que os dados recebidos pelo MATLAB realmente estejam corretos. Para tal, marca-se na tabela presente na aba *Data Output* (no menu de configurações do X-Plane 11) as caixas de seleção *Show in Cockpit*, *Disk (data.txt File)* e *Network via UDP*, Fig. 6.24. Desta forma, os dados da simulação serão não somente enviados através da rede, mas também mostrados durante a simulação no canto superior esquerdo da tela, Fig. 6.25, e salvos no arquivo *File.txt*, presente na pasta principal do X-Plane 11.

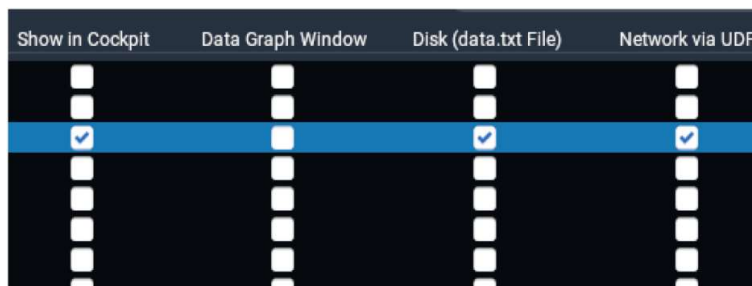


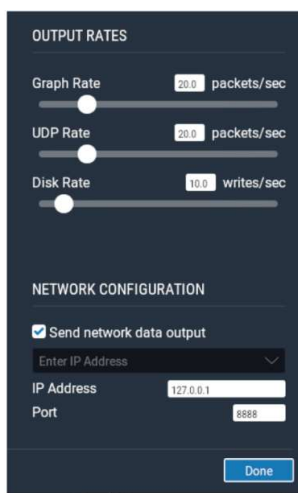
Figura 6.24: Caixas de seleção da tabela presente na aba *Data Output* (IASBECK, PEREIRA e SANCHES, 2020).



real	totl	miss	timer	zulu	local	hobbs
time	time	time	time	time	time	time
573.15	136.93	136.93	0.0000	20.038	12.038	0.0380
0.0000	0.0006	0.0006	0.0006	0.0000	0.0006	0.0006
ktas	ktas	ktas	ktas	mph	mph	mph
13.466	0.2287	3.5668				
ftlb	ftlb	ftlb				
0	0	0				
0.0001	0.0000	0.0001				
rad/s	rad/s	rad/s				
pitch	roll	yaw	nding			
2.9877	-1.8644	190.18	164.83			
deg	deg	true	deg			
mag	mag					
165.62	-15.343					
comp	deg					
X	Y	Z	VX	VY	VZ	dist
-23161	79.618	4007.7	0.0001	0.0001	0.0003	0.4955
m	m	m	m/s	m/s	m/s	ft
power						dist
3.4673						nm
1.1hp						

Figura 6.25: Dados de voo no X-Plane 11 (IASBECK, PEREIRA e SANCHES, 2020).

Ainda na aba *Data Output* é determinada a taxa de envio de dados através da rede UDP (número de pacotes enviados pelo X-Plane 11 a cada segundo), em *UDP Rate*, a taxa de escrita no arquivo *File.txt* (número de escritas no arquivo realizadas a cada segundo), em *Disk Rate*, e o IP do dispositivo para o qual o X-Plane enviará seus pacotes, em *IP Address*, Fig. 6.26. Ressalta-se que a taxa de envio de dados através da rede UDP, utilizada na simulação de forma equivalente à frequência de amostragem do sistema, deve ser determinada com base nas características dinâmicas da aeronave em estudo.



OUTPUT RATES

Graph Rate: 20.0 packets/sec

UDP Rate: 20.0 packets/sec

Disk Rate: 10.0 writes/sec

NETWORK CONFIGURATION

☒ Send network data output

Enter IP Address: [dropdown arrow]

IP Address: 127.0.0.1

Port: 8888

Done

Figura 6.26: Configurações da comunicação (IASBECK, PEREIRA e SANCHES, 2020).

Para o IP de destino dos pacotes enviados pelo X-Plane 11, é escolhido o local host. Este endereço representa o IP do dispositivo responsável por remeter a mensagem, o que significa que quando uma mensagem é enviada para ao local host, esta retorna ao dispositivo que a enviou. Neste caso, o X-Plane envia uma mensagem ao local host, e a mesma retorna para o computador em que ocorre a simulação. Uma vez que o MATLAB é executado

juntamente ao X-Plane e na mesma máquina, essa é uma das formas de fazer com que as mensagens enviadas pelo simulador cheguem até o MATLAB. O número da porta para onde as mensagens serão direcionadas pode ser estabelecido de forma arbitrária, porém é importante observar se nenhum outro software utiliza, durante a simulação, a porta selecionada. Neste trabalho, o número escolhido para a porta foi o 8888, e este parâmetro deve ser levado em consideração no desenvolvimento do código em MATLAB que deverá receber e tratar as mensagens.

Além de receber pacotes de dados do X-Plane 11, é possível também enviar ao simulador mensagens contendo comandos que possam alterar a deflexão das superfícies de comando da aeronave (ailerons, profundores e leme) ou a aceleração de seus motores. Desta forma é possível controlar o comportamento da aeronave por meio do envio de comandos através da rede. Como o MATLAB e o X-Plane 11 são executados simultaneamente e no mesmo computador, os pacotes que contêm os comandos são enviados ao local host (127.0.0.1), e à porta 49000 (utilizada por padrão pelo simulador para o recebimento de mensagens). Os pacotes enviados ao X-Plane devem possuir a mesma estrutura daqueles remetidos por ele: um cabeçalho de 5 bytes preenchido com os valores [68 65 84 65 0], em seguida o índice correspondente ao comando a ser enviado, e por fim os oito valores do tipo *float* devidamente representados como vetores de *bytes*.

Para enviar comandos ao simulador que ocasionem alterações na posição das superfícies de comando, o índice utilizado na construção do pacote encaminhado ao X-Plane deve ser o 11, Fig. 6.23. Assim sendo, a mensagem final é composta por cinco bytes de cabeçalho, [69 65 94 65 0], quatro de índice, [0 0 0 11], e trinta e dois de dados (os quatro primeiros referentes ao controle dos profundores, os quatro seguintes ao dos ailerons, e os próximos quatro ao do leme). Neste caso há apenas três informações a serem enviadas, sendo que o pacote encaminhado ao simulador deve conter oito valores *float*. Dessa forma, para satisfazer à estrutura de pacotes imposta pelo X-Plane 11, aos demais dados é atribuído o valor -999, ou mais precisamente, sua representação em bytes [0 192 121 196].

Os valores das variáveis referentes à deflexão das superfícies de comando, presentes no pacote de comandos enviado ao X-Plane, encontram-se sempre entre -1.0 e 1.0, visto que o simulador apenas recebe, através da rede UDP, comandos normalizados. Estes extremos representam as posições limite de cada uma das superfícies de comando. Caso o simulador receba uma mensagem contendo o valor referente à deflexão do profundor igual a 1.0, por exemplo, isso fará com que esta superfície de comando se incline para cima o máximo possível, enquanto que, sendo recebido o valor -1.0 nesta mesma variável, ocorrerá o oposto. Entre -1.0 e 1.0, todos os valores reais podem ser utilizados. Cabe ressaltar que uma vez que

uma dada posição tenha sido alcançada por alguma das superfícies de comando, esta será mantida até que um novo comando seja enviado ao X-Plane, ou até que a comunicação seja encerrada.

Durante a simulação, é possível que o usuário manobre o avião através de um joystick, ou utilizando o próprio mouse do computador. No entanto, uma vez que a comunicação tenha sido estabelecida e o MATLAB passe a enviar comandos ao X-Plane, o controle manual não poderá mais ser utilizado. Para que o usuário possa controlar novamente a aeronave, o programa desenvolvido no MATLAB deve enviar um pacote em que o valor -999 é atribuído a todo o conjunto de dados da mensagem, o que indica ao simulador que mais nenhuma mensagem será a ele encaminhada.

A Fig. 6.27 é um fluxograma que ilustra os processos envolvidos no *software-in-the-loop* realizado entre o MATLAB e o X-Plane.

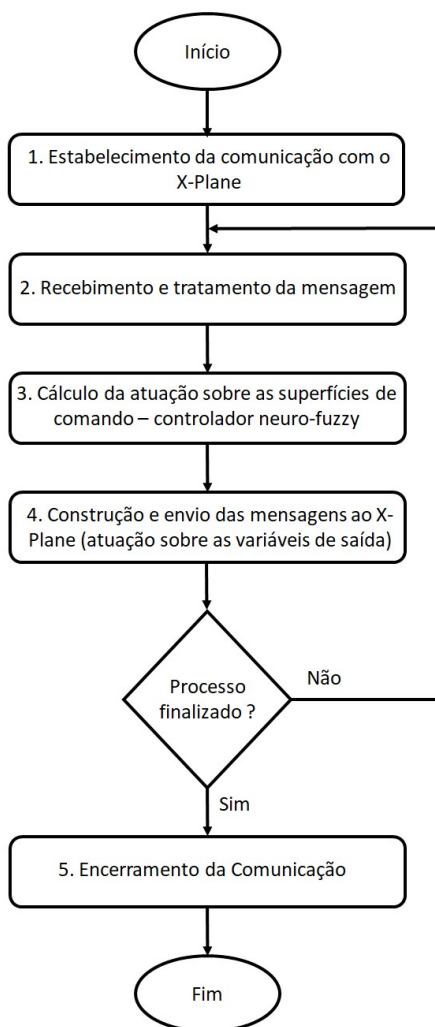


Figura 6.27: Fluxograma – *software-in-the-loop* entre MATLAB e X-Plane (autoria própria).

Os passos apresentados no fluxograma são detalhados a seguir:

1. Estabelecimento da comunicação com o X-Plane:

Para estabelecer a comunicação com o X-Plane 11, um *socket* UDP é criado e configurado para encaminhar mensagens ao IP 127.0.0.1 e à porta 49000, enquanto recebe mensagens na porta 8888. Depois de criado, o *socket* deve ser inicializado, como mostrado na Fig. 6.28.

```

1  % Inicializacao do socket UDP
2  display('Iniciando comunicação...');
3  sockUDP = udp('127.0.0.1', ...
               'RemotePort', 49000, 'LocalPort', ...
               8888);
4  fopen(sockUDP);
5  display('Comunicação UDP inicializada.');
```

Figura 6.28: Inicialização da comunicação (IASBECK, PEREIRA e SANCHES, 2020).

2. Recebimento e tratamento da mensagem:

Para que as mensagens enviadas pelo X-Plane 11 sejam recebidas pelo MATLAB, a leitura do *socket* UDP, inicializado no processo anterior, é realizada como mostrado na Fig. 6.29. A função *getData()*, Fig. 6.30, então extrai da mensagem recebida os 4 bytes referentes ao dado desejado (por exemplo, pitch, roll ou headings) e os repassa à função *bytes2single()*, Fig. 6.31. Esta função, por sua vez, transforma os bytes a ela repassados em uma única cadeia binária através da aplicação da função *data2bits()*, Fig. 6.32, para que, por fim, o número binário gerado seja então repassado à função *bits2single()*, Fig. 6.33, que converte este binário em um número real do tipo *float* (*single precision*) através da aplicação das Eqs. (6.6) a (6.9). Assim que obtidos os valores das variáveis de interesse, os mesmos são salvos em um arquivo para análise posterior.

```

1  msgRecv = fread(sockUDP);
2  pitch = getData(msgRecv, 2);
3  roll = getData(msgRecv, 3);
```

Figura 6.29: Recebimento das mensagens advindas do simulador (IASBECK, PEREIRA e SANCHES, 2020).

```

1 function data = getData(msgRecv, dataNum)
2     firstIndex = 6 + 4*(dataNum-1);
3     lastIndex = firstIndex + 3;
4     dataBytes = ...
        msgRecv(firstIndex:lastIndex);
5     data = bytes2single(dataBytes);
6 end

```

Figura 6.30: Função que obtém os dados, referentes ao comportamento da aeronave, contidos nos pacotes enviados pelo simulador (IASBECK, PEREIRA e SANCHES, 2020).

```

1 function single = bytes2single(bytes)
2     bits = data2bits(bytes);
3     single = bits2single(bits);
4 end

```

Figura 6.31: Função que converte um vetor de 4 bytes em um número do tipo *float – single precision* (IASBECK, PEREIRA e SANCHES, 2020).

```

1 function dataOut = data2bits(dataIn)
2     dataIn = dec2bin(dataIn,8);
3     % Note que o vetor de bits ...
        resultante é construído de ...
        trás para frente
4     % Isso é explicado pelo fato de o ...
        Windows utilizar o padrão little
5     % endian.
6     dataOut = [dataIn(4,:) ...
        dataIn(3,:) dataIn(2,:) ...
        dataIn(1,:)];
7 end

```

Figura 6.32: Função responsável pela conversão de um vetor de 4 bytes em uma cadeia binária formada pela concatenação dos números binários gerados a partir de cada um dos *bytes* (IASBECK, PEREIRA e SANCHES, 2020).

```

1 function dataOut = bits2single(dataIn)
2     bin = dataIn;
3     signal = (-1)^bin2dec(bin(1));
4     e = bin2dec(bin(2:9));
5     fraction = 1;
6     j = 1;
7     for i=10:32
8         fraction = fraction + ...
9             bin2dec(bin(i))*2^(-j);
10        j = j + 1;
11    end
12    value = signal*fraction*2^(e-127);
13    dataOut = value;
14 end

```

Figura 6.33: Função que converte um número binário de 32 bits em um número real do tipo *float – single precision* (IASBECK, PEREIRA e SANCHES, 2020).

3. Cálculo da atuação sobre as superfícies de comando – controlador neuro-fuzzy

Após o MATLAB ter acesso ao estado da aeronave, utiliza-se o controlador neuro-fuzzy para calcular os valores das deflexões das superfícies de comando e o nível de propulsão dos motores.

4. Construção e envio das mensagens ao X-Plane (atuação sobre as variáveis de saída)

A ação de controle obtida através do controlador neuro-fuzzy é enviada ao X-Plane através de funções como a *setElevator(...)*, Fig. 6.34, que recebe como argumentos a nova deflexão que o profundor deve assumir, e o *socket* UDP através do qual o pacote contendo esta ação de controle deve ser enviado. Dentro desta função, um pacote de dados é construído por meio da função *msgBuilder(...)*, Fig. 6.35, que por sua vez recebe como entradas todos os oito dados a serem enviados ao X-Plane, e o índice a eles correspondente, e converte cada um destes valores reais em um conjunto de quatro bytes, retornando o vetor de 32 *bytes* que compõe a mensagem final. Neste exemplo, como apenas a deflexão do profundor é alterada, é atribuído o valor -999 a todos os demais dados da mensagem. Por fim, o pacote de comando construído é enviado ao X-Plane através do *socket* UDP. A função responsável

por enviar ao X-Plane um valor normalizado correspondente, por exemplo, à nova deflexão dos ailerons, segue a mesma estrutura da função *setElevator(...)*.

```

1 function setElevator(value, sendUDP)
2     if value > 1
3         value = 1;
4     end
5     if value < -1
6         value = -1;
7     end
8     msgSend = ...
9         msgBuilder(11, value, -999, -999, ...
10             -999, -999, -999, -999, ...
11             -999);
12     fwrite(sendUDP, msgSend);
13 end

```

Figura 6.34: Função responsável por enviar ao X-Plane um pacote contendo um comando que altera a deflexão do profundor, em inglês, *elevator* (IASBECK, PEREIRA e SANCHES, 2020).

```

1 function msg = msgBuilder(dataType, ...
2     data1, data2, data3, data4, ...
3     data5, data6, data7, data8)
4     headerData = [68 65 84 65 0];
5     msgData = [single2bytes(data1)
6         single2bytes(data2)
7         single2bytes(data3)
8         single2bytes(data4)
9         single2bytes(data5)
10        single2bytes(data6)
11        single2bytes(data7)
12        single2bytes(data8)];
13     msg = [headerData ...
14         int2bytes(dataType) msgData];
15 end

```

Figura 6.35: Função responsável pela construção dos pacotes de comando enviados ao X-Plane durante a simulação (IASBECK, PEREIRA e SANCHES, 2020).

5. Encerramento da comunicação

Para que a comunicação entre o MATLAB e o X-Plane 11 seja encerrada, Fig. 6.36, é enviado ao simulador um pacote de comandos formado integralmente pelo valor -999. A função *leaveControl(...)*, Fig. 6.37, constrói este pacote e o envia ao X-Plane, para que, em seguida, o socket utilizado no estabelecimento da comunicação seja fechado.

```

1  leaveControl(sockUDP);
2  display('Encerrando comunicação UDP...');
3  fclose(sockUDP);
4  delete(sockUDP);
5  clear sockUDP;
6  display('Comunicação UDP encerrada.');
```

Figura 6.36: Seção do código que encerra a comunicação entre o MATLAB e o X-Plane 11 (IASBECK, PEREIRA e SANCHES, 2020).

```

1  function leaveControl(sockUDP)
2      msgSend = msgBuilder(11, -999, ...
                           -999, -999, -999, -999, -999, ...
                           -999, -999);
3      fwrite(sockUDP, msgSend);
4  end
```

Figura 6.37: Função que envia ao X-Plane 11 um pacote de comandos que informa ao simulador que nenhuma outra mensagem será remetida pelo MATLAB, e que a aeronave, a partir de então, passará a responder apenas a intervenções diretas do usuário (IASBECK, PEREIRA e SANCHES, 2020).

Após o desenvolvimento da interface de comunicação UDP, realiza-se a análise de desempenho do controlador neuro-fuzzy, sintonizado na seção anterior, em operações críticas de voo através de uma simulação do tipo *software-in-the-loop* envolvendo os softwares MATLAB e X-Plane 11.

Análise de Desempenho do Controlador em Operações Críticas de Voo

O relatório apresentado em IATA (2020) aponta que as condições meteorológicas críticas, o mau funcionamento da aeronave, e o mau funcionamento ou falha do sistema propulsor estão entre os principais fatores causadores da perda de controle da aeronave durante o voo, com taxas de ocorrência respectivamente iguais a 45, 35 e 20%, sendo que em muitos dos casos há a ocorrência simultânea de mais de um fator de risco. Dessa forma, no intuito de verificar a eficácia da técnica desenvolvida no controle de aeronave em operações críticas de voo, realizam-se cinco simulações do tipo *software-in-the-loop* entre o MATLAB e o X-Plane 11, a fim de avaliar o desempenho do sistema neuro-fuzzy no controle da aeronave Cessna 172: três simulações avaliando cada uma das condições mencionadas anteriormente de forma isolada; uma considerando apenas condição inicial não nula; e uma avaliando a ocorrência simultânea de todas as condições críticas citadas.

Na primeira simulação, Fig. 6.38, verifica-se a capacidade do sistema neuro-fuzzy de estabilizar a aeronave submetida apenas a uma condição inicial de voo não nula. Nas Figs. 6.39 e 6.40 são apresentados respectivamente os gráficos dos ângulos de arfagem θ e rolagem ϕ em função do tempo t .

No gráfico apresentado na Fig. 6.39, verifica-se que a simulação se inicia com $\theta = 3,7^\circ$, atinge sobressinal equivalente a aproximadamente $1,85^\circ$, e no tempo $t = 16\text{ s}$ já ocorre a acomodação do ângulo de arfagem da aeronave, que alcança o valor de $0,015^\circ$. No gráfico apresentado na Fig. 6.40, verifica-se que a simulação se inicia com $\phi = 4,6^\circ$, atinge sobressinal equivalente a aproximadamente $2,2^\circ$, e também no tempo $t = 16\text{ s}$ ocorre a acomodação do ângulo de rolagem, que alcança o valor de $0,001^\circ$. A partir da análise realizada, constata-se a eficácia do sistema neuro-fuzzy, com o método PIA de inferência fuzzy, em levar o erro com relação aos sinais de referência *set* para praticamente zero nesta condição de operação da aeronave.



Figura 6.38: Ilustração da primeira simulação realizada no simulador de voo X-Plane 11 (autoria própria).

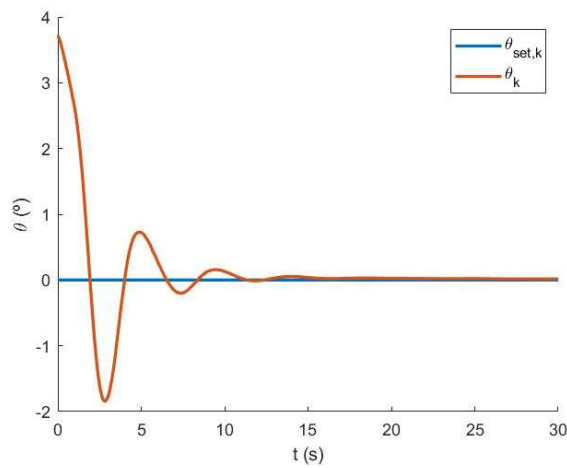


Figura 6.39: Ângulo de arfagem da aeronave $\theta(^{\circ})$ em função do tempo $t(s)$ na primeira simulação (autoria própria).

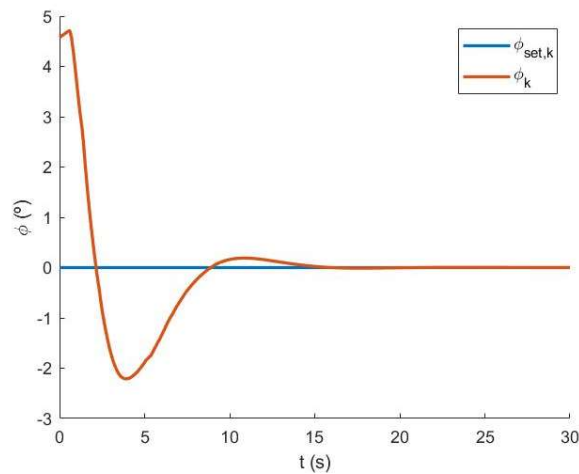


Figura 6.40: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo $t(s)$ na primeira simulação (autoria própria).

Na segunda simulação, verifica-se a capacidade do sistema neuro-fuzzy de estabilizar a aeronave submetida a mau funcionamento da semiasa direita. Nas Figs. 6.41 e 6.42 são apresentados respectivamente os gráficos dos ângulos de arfagem θ e rolagem ϕ em função do tempo t .

No gráfico apresentado na Fig. 6.41, verifica-se que a simulação se inicia com $\theta = -1,7^\circ$, atinge sobressinal equivalente a aproximadamente $0,75^\circ$, e no tempo $t = 16\text{ s}$ já ocorre a acomodação do ângulo de arfagem da aeronave, que alcança o valor de $-0,006^\circ$. No gráfico apresentado na Fig. 6.42, verifica-se que a simulação se inicia com $\phi = 7,7^\circ$, atinge sobressinal equivalente a aproximadamente $4,4^\circ$, e no tempo $t = 18\text{ s}$ ocorre a acomodação do ângulo de rolagem, que alcança o valor de $-0,015^\circ$. Constata-se então a eficácia do controlador neuro-fuzzy em levar o erro com relação aos sinais de referência *set* para zero para essa condição de mau funcionamento da aeronave, somente, porém, com um pequeno aumento do tempo de acomodação do ângulo de rolagem comparativamente à simulação anterior.

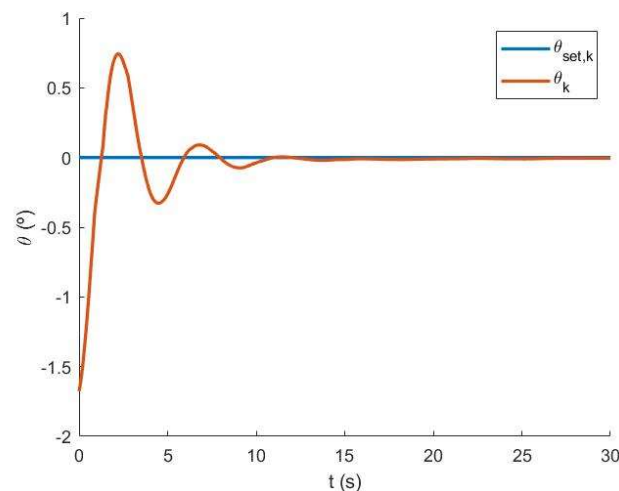


Figura 6.41: Ângulo de arfagem da aeronave $\theta(^\circ)$ em função do tempo $t(s)$ na segunda simulação (autoria própria).

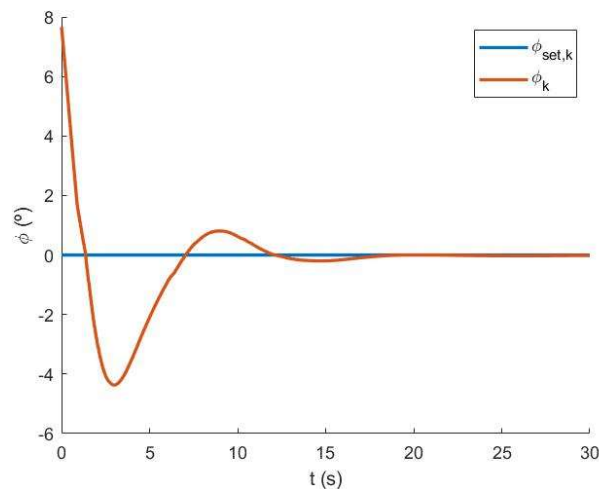


Figura 6.42: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo $t(s)$ na segunda simulação (autoria própria).

Na terceira simulação, analisa-se a eficácia do sistema neuro-fuzzy de controlar a aeronave Cessna 172 submetida à perda total de potência do sistema propulsor. Nas Figs. 6.43 e 6.44 são apresentados novamente, respectivamente, os gráficos dos ângulos de arfagem e rolagem em função do tempo.

No gráfico apresentado na Fig. 6.43, verifica-se que a simulação se inicia com $\theta = -7,6^{\circ}$, atinge sobressinal equivalente a aproximadamente $2,9^{\circ}$, e no tempo $t = 16\text{ s}$ já ocorre a acomodação do ângulo de arfagem da aeronave, que alcança o valor de $-0,042^{\circ}$. No gráfico apresentado na Fig. 6.44, verifica-se que a simulação se inicia com $\phi = 6,5^{\circ}$, atinge sobressinal equivalente a aproximadamente $2,3^{\circ}$, e também no tempo $t = 16\text{ s}$ ocorre a acomodação do ângulo de rolagem, que alcança o valor de $-0,005^{\circ}$. Embora seja verificada mais uma vez a competência do controlador desenvolvido nessa tese em minimizar o erro com relação aos sinais de referência, em função da falha do sistema propulsor e da inevitável perda gradual de altitude da mesma, há o surgimento de um pequeno erro em regime permanente para o ângulo de arfagem, mas que não compromete o desempenho do sistema.

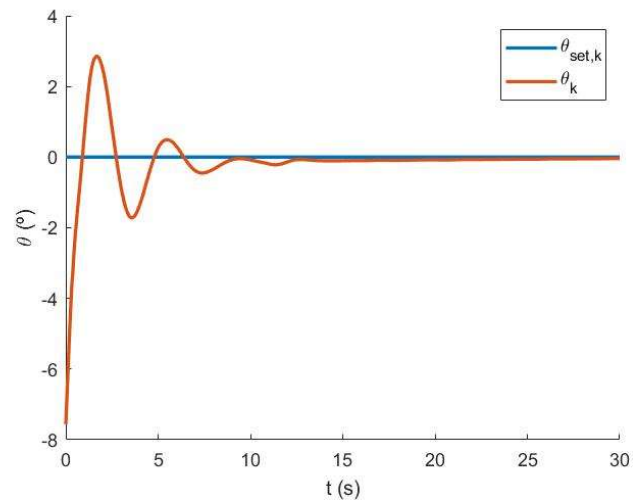


Figura 6.43: Ângulo de arfagem da aeronave $\theta(^{\circ})$ em função do tempo $t(s)$ na terceira simulação (autoria própria).

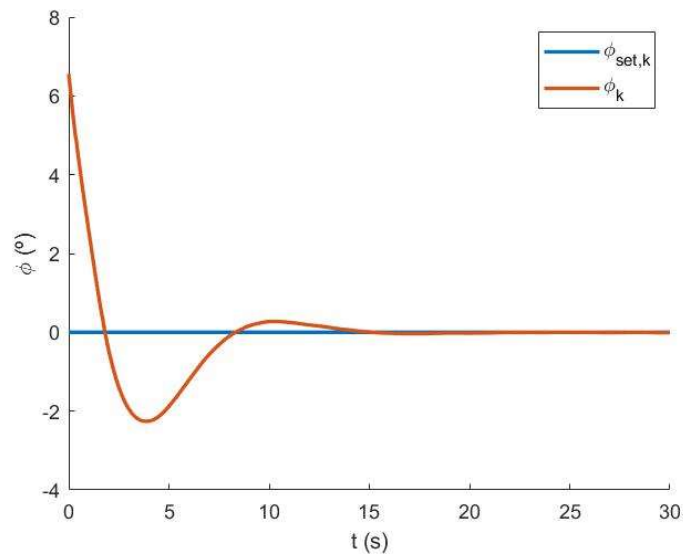


Figura 6.44: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo $t(s)$ na terceira simulação (autoria própria).

Na quarta simulação, Fig. 6.45, a capacidade do sistema neuro-fuzzy de controlar a aeronave submetida a condições climáticas severas (visibilidade inferior a 3 milhas, altura da base das nuvens mais baixas inferior a 1000 pés acima do nível do solo, forte precipitação e tempestade) é colocada a prova. Nas Figs. 6.46 e 6.47 são apresentados respectivamente os gráficos dos ângulos de arfagem e rolagem em função do tempo.

No gráfico apresentado na Fig. 6.46, verifica-se que a simulação se inicia com $\theta = 6,3^\circ$, e chega a alcançar até ângulo de arfagem aproximadamente igual a $7,8^\circ$. No gráfico apresentado na Fig. 6.47, nota-se que a simulação se inicia com $\phi = 2,0^\circ$, e chega a alcançar até ângulo de rolagem aproximadamente igual a $-6,6^\circ$. As condições climáticas críticas de fato impõem dificuldade para o sistema de controle da aeronave Cessna 172, porém é possível observar em ambos os gráficos a tendência de oscilação do sistema em torno do sinal de referência e também de redução da amplitude da resposta com o avanço do tempo.



Figura 6.45: Ilustração da quarta simulação realizada no simulador de voo X-Plane 11 (autoria própria).

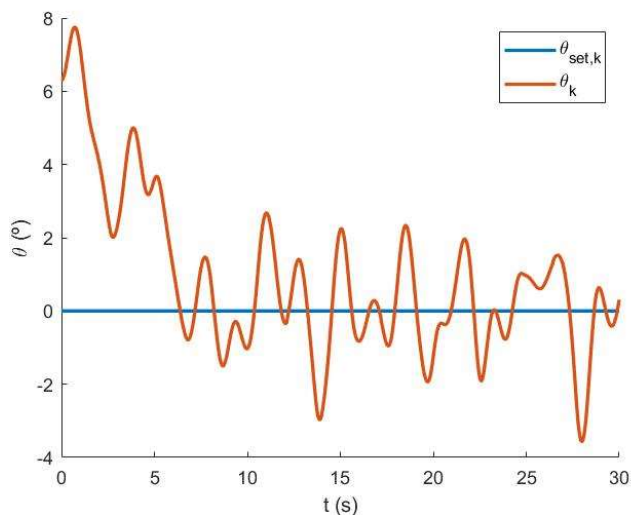


Figura 6.46: Ângulo de arfagem da aeronave θ ($^\circ$) em função do tempo t (s) na quarta simulação (autoria própria).

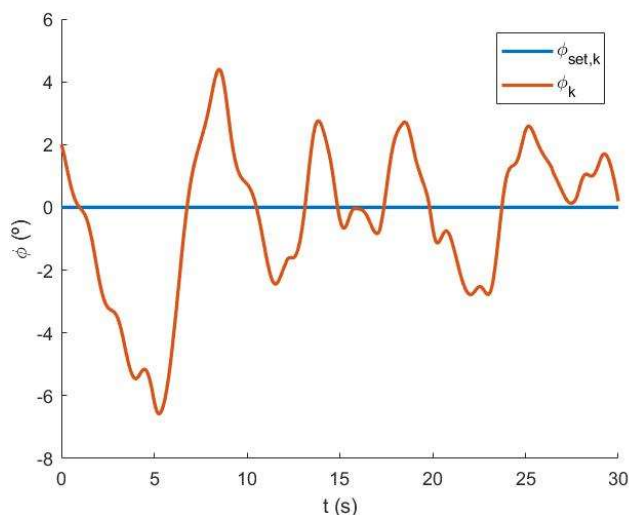


Figura 6.47: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo $t(s)$ na quarta simulação (autoria própria).

Por fim, na quinta simulação, submete-se a aeronave Cessna 172 a todas as condições iniciais e perturbações mencionadas anteriormente e de forma simultânea. Nas Figs. 6.48 e 6.49 são apresentados novamente, respectivamente, os gráficos dos ângulos de arfagem e rolagem em função do tempo, mas neste momento com uma janela de tempo de 100 segundos.

No gráfico apresentado na Fig. 6.48, verifica-se que a simulação se inicia com $\theta = -7,80^{\circ}$, e durante o processo alcança ângulo de arfagem aproximadamente igual a $-7,75^{\circ}$. Já no gráfico apresentado na Fig. 6.49, nota-se que a simulação se inicia com $\phi = -7,20^{\circ}$, e durante o processo alcança ângulo de rolagem aproximadamente igual a $6,80^{\circ}$. Em função das condições climáticas severas, há dificuldade por parte do sistema de controle de manter o voo reto e nivelado, e a falha do sistema propulsor causa inevitável perda gradual de altitude da aeronave e o consequente surgimento de um desvio da média da resposta com relação ao sinal de referência para o ângulo de arfagem. Apesar das condições críticas impostas e dos seus efeitos sobre o desempenho do sistema controlado, o sistema neuro-fuzzy é capaz de autonomamente manter a oscilação da aeronave, durante todo o tempo de simulação, ao redor do valor de referência para o ângulo de rolagem ϕ , e com um desvio de apenas -1° para o ângulo de arfagem θ (valor aceitável partindo do princípio de que na simulação não há geração de potência por parte do motor).

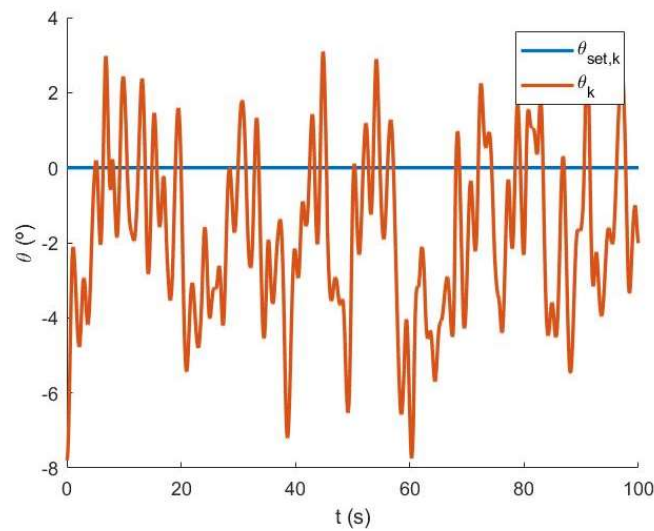


Figura 6.48: Ângulo de arfagem da aeronave $\theta(^{\circ})$ em função do tempo $t(s)$ na terceira simulação (autoria própria).

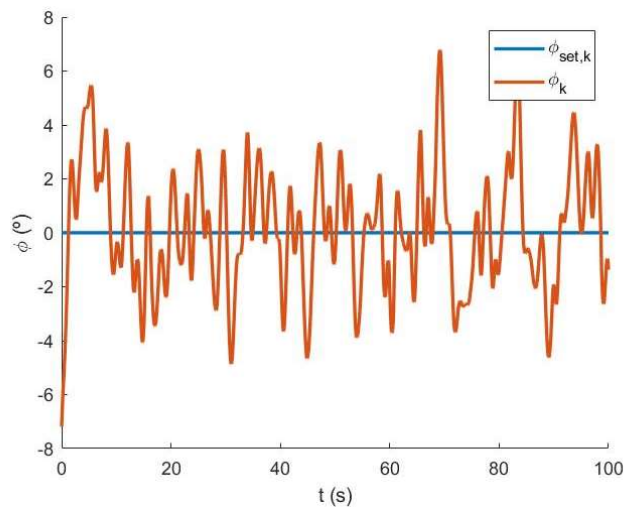


Figura 6.49: Ângulo de rolagem da aeronave $\phi(^{\circ})$ em função do tempo $t(s)$ na terceira simulação (autoria própria).

A validação da arquitetura de controle e do método PIA de inferência fuzzy é desenvolvida até o momento a partir de simulações considerando inicialmente o modelo linearizado da aeronave Cessna 172, e em seguida a modelagem disponibilizada pelo simulador de voo X-Plane. Na próxima seção, as técnicas propostas neste trabalho são validadas levando-se em consideração um modelo dinâmico obtido a partir de dados experimentais extraídos em ensaios de voo, os quais são realizados em uma aeronave Cessna 172 em escala reduzida.

6.4. Controle de Aeronave em Escala Reduzida

A validação das técnicas desenvolvidas nesta tese também passa pela utilização de dados experimentais, assim, constrói-se um protótipo de aeronave em escala reduzida e são realizados ensaios de voo no intuito de se extrair dados a serem utilizados na modelagem do VANT, e na posterior verificação, em simulação, da capacidade das técnicas de satisfatoriamente controlar o sistema em estudo. Esse último processo de validação pode ser subdividido em etapas, a saber: construção de aeronave em escala reduzida, obtenção de dados de voo, identificação do modelo da aeronave e validação da estratégia de controle.

Construção de Aeronave em Escala Reduzida

Aeronaves em escala reduzida possibilitam a validação das técnicas de controle de forma experimental, e com baixo custo financeiro se comparados os ensaios aos testes em aeronaves comerciais.

Utiliza-se nos ensaios experimentais uma aeronave Cessna 172 em escala reduzida de 1:10, sendo composta pela fuselagem, trem de pouso e hélice do aeromodelo Cessna 172 Class 500, Fig. 6.50, desenvolvida pela empresa Art-tech; por um motor elétrico do tipo *brushless* de 800 Kv (rpm/V); e por um controlador de voo (que substitui o original do aeromodelo) denominado Pixhawk-PX4, o qual é um hardware de baixo custo, de projeto aberto e desenvolvido pela comunidade acadêmica, Fig. 6.51.



Figura 6.50: Fuselagem, trem de pouso e hélice do aeromodelo Cessna 172 Class 500 (adaptada de SANT'ANA, 2019).



Figura 6.51: Controlador de voo Pixhawk e sua localização na aeronave (adaptada de SANT'ANA, 2019).

Após a construção da aeronave em escala reduzida, segue-se com o planejamento e execução dos ensaios de voo, e consequentemente com a obtenção dos dados experimentais.

Obtenção de Dados de Voo

O controlador de voo Pixhawk centraliza uma série de importantes funções, a saber:

- Fornecer a atitude da aeronave por meio do acelerômetro integrado à placa – para tal, é fundamental que esta esteja situada próxima ao centro de gravidade da aeronave, de onde parte o referencial baricêntrico do sistema;
- Receber os sinais dos sensores externos, como o módulo de GPS e o tubo de pitot, e também os comandos provindos do piloto do VANT;
- Filtrar os sinais através sobretudo do filtro de Kalman, cujos principais aspectos são descritos no Apêndice A deste trabalho;
- Processar os sinais filtrados, calcular o valor das saídas, e controlar o propulsor e os servomotores que defletem as superfícies de comando da aeronave;
- Armazenar os dados de voo em um cartão de memória SD inserido na placa.

Após o ajuste do controlador de voo, e a verificação de sua sintonia com o computador localizado em solo e com controle remoto do piloto, inicia-se o plano de voo. O plano de voo é desenvolvido no software Ardupilot instalado no computador em solo, em que são traçados pontos de destino (trajetória a ser seguida de maneira autônoma pela aeronave), em

seqüência o piloto retoma o controle manual da aeronave, induz perturbações à mesma no intuito de proporcionar dados de voo que propiciem a obtenção de um modelo mais representativo do comportamento dinâmico da aeronave, e por fim poussa-se esta.

O ensaio de voo tem como localização a entrada do Campus Glória da Universidade Federal de Uberlândia e possui tempo total de duração de 10 minutos. O plano de voo é executado de forma satisfatória e os dados são armazenados em cartão SD para que posteriormente sejam utilizados na identificação do modelo da aeronave. O software Ardupilot, em que é inserido o plano de voo do aeromodelo, e a foto da aeronave pronta para voo no Campus Glória são apresentados na Fig. 6.52.



Figura 6.52: Software Arduplane e aeronave Cessna 172 em escala reduzida pronta para voo (adaptada de SANT'ANA, 2019).

Identificação do Modelo da Aeronave

O processo de identificação do modelo dinâmico da aeronave se inicia com a avaliação dos dados obtidos em voo e armazenados no cartão de memória SD. Em função do alto grau de complexidade do processo de identificação do modelo completo da aeronave, e sobretudo em função do alto nível de ruído a que os dados extraídos estão submetidos (devido à turbulência gerada pela hélice, à vibração do motor, à baixa inércia da aeronave e à própria imprecisão intrínseca dos sensores), opta-se por identificar um modelo do tipo SISO (*Single Input Single Output*) que relaciona a deflexão do profundor δE ao ângulo de arfagem θ do Cessna 172 em escala reduzida. Nas Figs. 6.53(a) e 6.53(b) são apresentados os gráficos da deflexão do profundor e do ângulo de arfagem obtidos durante os 600 segundos de ensaio de voo, respectivamente.

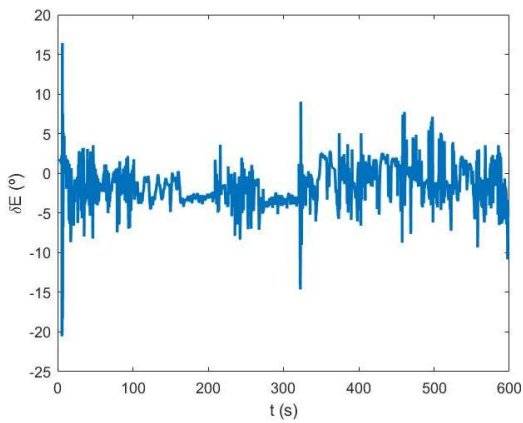
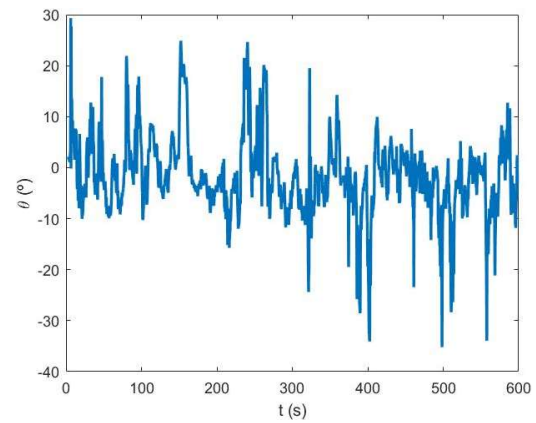
(a) δE em função do tempo $t(s)$.(b) θ em função do tempo $t(s)$.

Figura 6.53: Dados extraídos em voo (autoria própria).

Para a obtenção de um modelo dinâmico representativo do sistema aeronáutico, tomam-se duas amostras de 20 segundos cada, uma para a identificação e outra para a validação da estrutura matemática resultante, as quais são contidas no conjunto de dados ilustrados na Fig. 6.53. As amostras de identificação e de validação são apresentadas respectivamente nas Figs. 6.54 e 6.55.

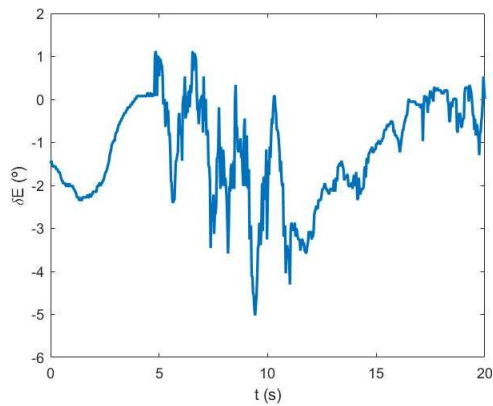
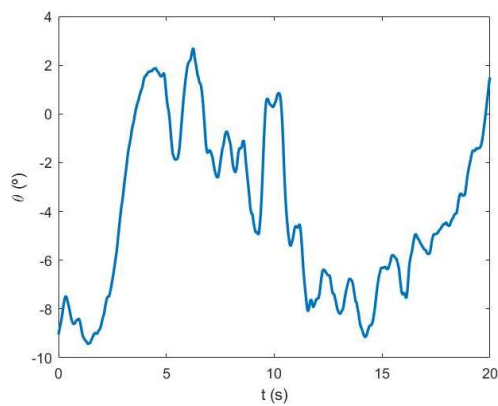
(a) δE em função do tempo $t(s)$.(b) θ em função do tempo $t(s)$.

Figura 6.54: Amostra de identificação do modelo dinâmico (autoria própria).

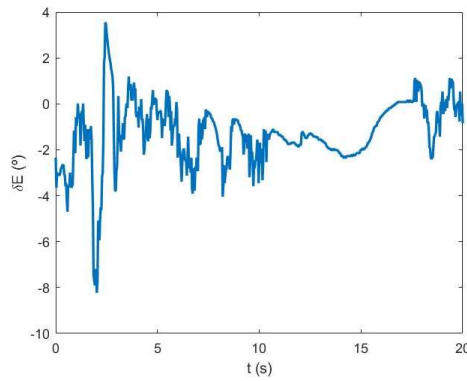
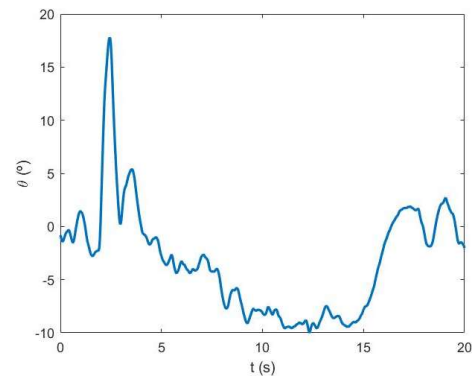
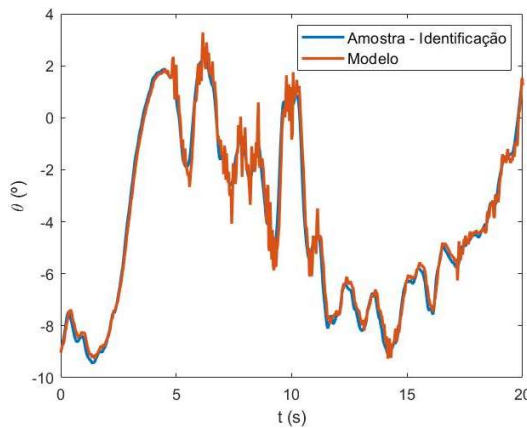
(a) δE em função do tempo $t(s)$.(b) θ em função do tempo $t(s)$.

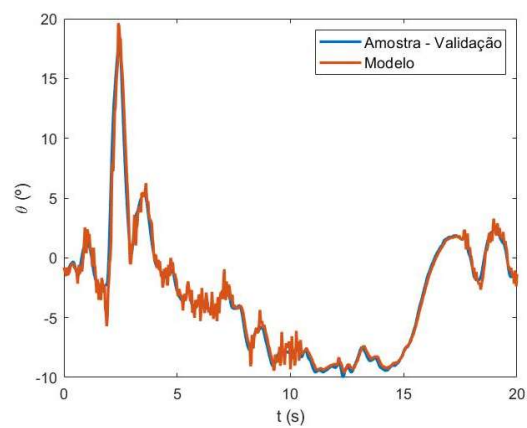
Figura 6.55: Amostra de validação do modelo dinâmico (autoria própria).

Utiliza-se para a determinação do modelo do sistema a evolução diferencial como método de otimização, cujos parâmetros são os mesmos apresentados na Tab. 6.9. A técnica busca minimizar o erro absoluto médio *MAE* (*Mean Absolute Error*) da resposta do modelo com relação aos dados experimentais da amostra de identificação. Adota-se, de forma intuitiva e por meio de tentativa e erro, um modelo em transformada z com 3 pólos e 2 zeros, e levando-se em consideração o tempo mínimo de amostragem da plataforma Pixhawk de 0,032 s. A Eq. (6.10) é o modelo matemático alcançado nesse processo de otimização, ressaltando-se que foram obtidas as taxas de ajuste aos dados experimentais das amostras de identificação, Fig. 6.56(a), e de validação, Fig. 6.56(b), iguais respectivamente a 92,3% e 90,3%:

$$G(z) = 0,82 \frac{(z-0,98)(z+0,94)}{(z-0,99)(z^2+0,38z+0,69)}. \quad (6.10)$$



(a) Amostra de identificação.



(b) Amostra de validação.

Figura 6.56: Ajuste do modelo dinâmico às amostras seleccionadas (autoria própria).

Após o processo de identificação do modelo dinâmico da aeronave Cessna 172 em escala reduzida, a seguir é apresentada mais uma aplicação e validação da estratégia de controle desenvolvida nesta tese.

Validação da Estratégia de Controle

A arquitetura de controle utilizada no controle do aeromodelo é basicamente a apresentada na Fig. 6.1, porém simplificada de modo a se ajustar a um sistema do tipo SISO. Aplicam-se, de forma semelhante à aeronave em escala real, imposições ao processo, conforme a Eq. (6.11):

$$\begin{cases} \theta_{set,k} = 0, 0 \leq t < 30 \text{ s} \\ \theta_k(t = 0) = 10^\circ \\ \theta_k(t = 15 \text{ s}) = -10^\circ, \end{cases} \quad (6.11)$$

de modo a viabilizar a otimização dos parâmetros do *SBRF* e consequentemente a definição de suas funções de pertinência.

O desempenho do sistema aeronáutico e as ações de controle geradas pela técnica proposta nesta tese são apresentados respectivamente nas Figs. 6.57(a) e 6.57(b). Pode-se notar na Fig. 6.57(a) que, partindo da condição inicial $\theta_k(t = 0) = 10^\circ$, rapidamente a aeronave é capaz de buscar o valor de referência $\theta_{set,k} = 0$, entrando em regime permanente após um intervalo de tempo de 2,5 s, sendo que o mesmo ocorre após a imposição do ângulo de arfagem $\theta_k(t = 15 \text{ s}) = -10^\circ$. Na Fig. 6.57(b), verifica-se que ocorre a deflexão do profundor nos instantes em que o ângulo de arfagem é diferente do valor de referência, havendo picos de deflexão de $8,5^\circ$ e $-8,5^\circ$ nos instantes $t = 0$ e $t = 15 \text{ s}$, respectivamente, seguidos de progressiva redução até o ângulo de arfagem atingir valor nulo.

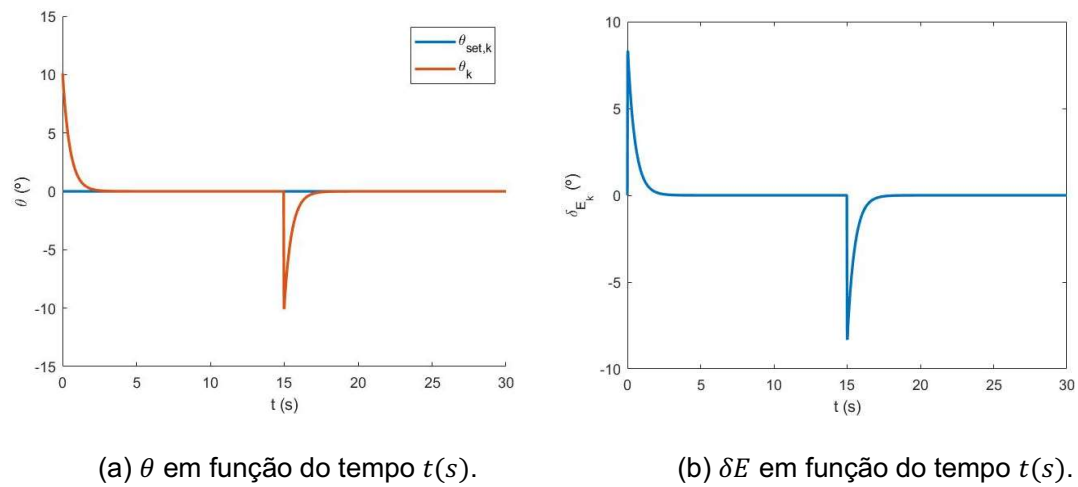


Figura 6.57: Resultados obtidos a partir do controlador neuro-fuzzy que lança mão do método PIA de inferência fuzzy (autoria própria).

No intuito de propiciar uma análise mais apurada dos resultados obtidos com o método PIA de inferência fuzzy, apresentam-se na Fig. 6.58 os gráficos do desempenho dos controladores PID, e neuro-fuzzy com o método de inferência de Takagi-Sugeno de ordem 0 (TS), assim como os gráficos dos sinais de referência set no tempo discreto k , e o decaimento requerido para a resposta.

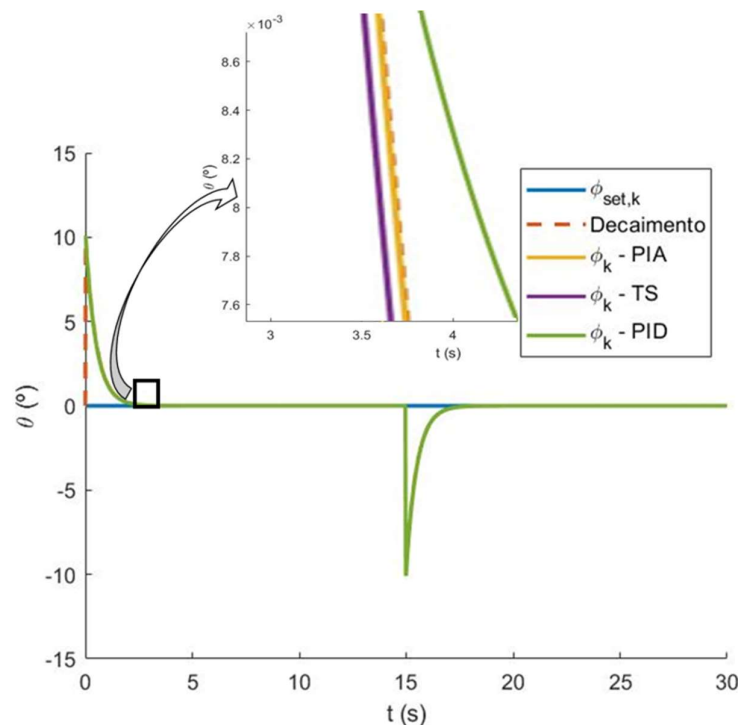


Figura 6.58: Análise comparativa das técnicas de controle (autoria própria).

A partir da Fig. 6.58, é possível verificar a capacidade de todas as técnicas de controlar efetivamente o ângulo de arfagem da aeronave Cessna 172 em escala reduzida, porém tendo o controlador que utiliza o método PIA, nesta análise comparativa de desempenho, uma pequena vantagem diante dos demais, no que diz respeito à proximidade da resposta obtida com relação ao sinal de decaimento desejado. Essa vantagem também se repete na análise do erro absoluto médio (MAE) da resposta do sistema controlado, com relação ao decaimento de referência, para cada uma das metodologias testadas: 0,0134 utilizando o sistema neuro-fuzzy com o método PIA; 0,0135 utilizando o sistema neuro-fuzzy com o método de TS; e 0,0159 utilizando o controlador PID. Os resultados obtidos nessa análise final ratificam a eficácia do método PIA também como estrutura matemática de um sistema de controle.

CAPÍTULO VII

CONCLUSÕES

O trabalho apresenta a aplicação de sistemas neuro-fuzzy no controle de aeronaves em operações críticas, com ênfase no desenvolvimento de um novo método de inferência fuzzy denominado PIA (*Pondered Individual Analysis*). Na intenção de facilitar a exposição dos principais pontos relativos às conclusões, perspectivas e resultados acadêmicos da tese, este capítulo é dividido em 3 seções, a saber: principais conclusões, perspectivas e trabalhos futuros, e principais publicações.

7.1. Principais Conclusões

Em linhas gerais, o método de inferência fuzzy PIA alia baixo custo computacional, intuitividade, consistência da resposta com ele obtida com relação à base de regras construída a priori, e superfícies de resposta suaves, sendo dessa forma uma técnica potencialmente indicada tanto para a modelagem, quanto para o controle de processos.

No primeiro teste de validação do novo método de inferência fuzzy, busca-se modelar os coeficientes de sustentação C_L e arrasto C_D em função do ângulo de ataque α e da deflexão do profundor δ_e de uma aeronave Cessna 172. Em uma análise qualitativa da superfície de resposta resultante do modelo utilizando os métodos de inferência fuzzy PIA, Mamdani e

Takagi-Sugeno, a nova técnica é a que obtém superfícies de resposta mais suaves e compatíveis com a base de regras construída a priori. Os resultados satisfatórios se repetem na análise quantitativa, haja vista que o método PIA, no ajuste a dados numérico-experimentais extraídos no simulador de voo X-Plane 11, garante o menor desvio aos dados com relação aos demais métodos, e com baixo custo computacional já que dispensa, no processo de defuzzificação, o uso de ferramentas de integração.

O segundo processo de validação conta com uma arquitetura de controle neuro-fuzzy inovadora, que também lança mão do método PIA de inferência fuzzy, que se propõe a controlar a velocidade, o ângulo de arfagem, a altitude, o ângulo de rolagem e o ângulo de guinada de uma aeronave Cessna 172. Através das simulações realizadas, é possível concluir que essa arquitetura de controle apresenta o melhor desempenho geral dentre as metodologias testadas, tendo obtido resultados superiores com relação aos controladores PID e neuro-fuzzy que utiliza o método de inferência fuzzy de Takagi-Sugeno em 3 das 5 variáveis de saída do sistema. Esse resultado indica o desempenho computacional e generalidade do método PIA, que mostra-se também eficaz como estrutura matemática de um sistema de controle.

Operações críticas de voo que dão origem a acidentes aéreos da categoria LOC-I (*Loss of Control In-Flight*), como condições meteorológicas críticas, mau funcionamento da aeronave, e o mau funcionamento ou falha do sistema propulsor, foram causa de aproximadamente 70% de todas as fatalidades ocorridas em aeronaves com massa de decolagem superior a 5.700 kg entre 2015 e 2019, assim, devido à relevância do tema, são utilizadas como objeto de estudo da terceira etapa de validação do novo método de inferência fuzzy, na intenção de também provocar a comunidade científica a buscar novos olhares no que diz respeito à utilização de técnicas da inteligência computacional no controle de aeronaves. Durante as simulações do tipo *software-in-the-loop* realizadas entre o MATLAB e o simulador de voo X-Plane, a capacidade da arquitetura de controle desenvolvida de controlar a aeronave submetida a condições severas é colocada a prova. As condições críticas de fato impõem dificuldade para o sistema de controle que lança mão do método PIA, porém é possível observar, em praticamente todas as simulações, a tendência de oscilação do sistema em torno do sinal de referência e também de redução da amplitude da resposta com o avanço do tempo.

Por fim, a última análise se dá a partir de um modelo dinâmico obtido a partir de dados experimentais extraídos em ensaios de voo, os quais são realizados em uma aeronave Cessna 172 em escala reduzida. Os resultados, quando comparados novamente aos obtidos com os controladores PID e neuro-fuzzy que utiliza o método de inferência fuzzy de Takagi-

Sugeno, indicam menor erro médio quadrático com relação ao comportamento (decaimento) desejado para a aeronave, e dessa forma confirmam a eficácia da nova estrutura matemática aplicada à área de controle.

7.2. Perspectivas e Trabalhos Futuros

O trabalho desenvolvido nesta tese serve potencialmente de suporte para uma série de estudos diretamente relacionados ao escopo desta pesquisa, ou até mesmo para outras áreas do conhecimento, assim, pode-se dividir as perspectivas e trabalhos futuros em dois grupos: continuidade das atividades, e perspectivas de aplicação do método de inferência fuzzy PIA.

Continuidade das Atividades

Diante do rigor necessário para a validação de uma arquitetura de controle a ser aplicada em um sistema aeronáutico, o seu teste em uma aeronave em escala reduzida pode ser mais um passo para viabilizar a realização de ensaios em aeronaves comerciais. A implementação da arquitetura de controle já se encontra em curso, está sendo desenvolvida em linguagem C++, e terá como hardware de execução de suas rotinas a própria plataforma Pixhawk.

O fenômeno do estol (perda abrupta de sustentação da aeronave durante o voo) é um potencial causador de perda de controle da aeronave durante o voo, porém, em função da alta complexidade matemática envolvida e dos altos custos relacionados aos ensaios de voo, apenas nos últimos anos ganha efetivo espaço nos trabalhos que envolvem o controle de aeronaves. Em função da viabilidade tecnológica dos novos controladores e sistemas computacionais, busca-se estudar a aplicação da arquitetura de controle desenvolvida também na mitigação dos efeitos do estol.

Por fim, dentro dos aspectos de continuidade das atividades desenvolvidas na tese, pretende-se realizar uma análise técnica do comportamento (decaimento) da aeronave em função de suas dimensões e das condições de operação, e também desenvolver uma análise de computabilidade do novo método de inferência fuzzy no intuito de tornar as comparações,

quanto ao tempo de execução dos métodos de inferência, independentes da máquina em que se realizam os testes.

Perspectivas de aplicação do método de inferência fuzzy PIA

O método PIA, assim como os demais métodos de inferência fuzzy, é caracterizado pela generalidade, assim, prospecta-se que o mesmo seja utilizado em uma série de aplicações, algumas das quais expostas a seguir:

- Inclusão do método em *toolboxes* comumente utilizadas quando da utilização da teoria dos conjuntos fuzzy, de modo a popularizar e difundir o método na comunidade científica;
- Utilização na modelagem e predição do mercado de criptomoedas;
- Aplicação do método na biomatemática, área em que a teoria dos conjuntos fuzzy é amplamente utilizada;
- Realização de comparações diversas entre o desempenho do método PIA e dos demais métodos de inferência fuzzy.

7.3. Principais Publicações

As principais publicações produzidas durante o período do doutorado são mencionadas a seguir e divididas em três categorias: trabalhos apresentados em congressos, trabalhos publicados como capítulos de livro e trabalhos em periódicos.

Trabalhos Apresentados em Congressos

IASBECK, A.H., PEREIRA, B.L. & SANCHES, L. UDP Communication Between MATLAB and X-Plane 11. **Anais do XXIII Encontro Nacional de Modelagem Computacional (ENMC)**, Palmas, 2020.

BERNARDES, R., PEREIRA, B.L., MACHINI, F.M.M. & FINZI, R.M. Neuro-Fuzzy Systems Applied on a Fixed-Wing Aircraft Control. **International Congress of Mechanical Engineering (COBEM)**, Uberlândia, 2019.

SANT'ANA, V., MACHINI, F.M.M., PEREIRA, B.L. & FINZI, R.M. Multi-Copter Forces and Moments Coefficients Estimation Using Neuro-Fuzzy. **International Congress of Mechanical Engineering (COBEM)**, Uberlândia, 2019.

CARVALHO, F.C., OLIVEIRA, M.V.F., PEREIRA, B.L., CAVALINI JR, A.A. & STEFFEN JR, V. Systematic Approach to the Design of Robust Controllers Applied to Supercritical Rotors Supported by Magnetic Bearings. **Proceedings of International Conference on Structural Engineering Dynamics (ICEDyn)**, Viana do Castelo, Portugal, 2019.

CARVALHO, F. C., OLIVEIRA, M. V. F., PEREIRA, B. L., CAVALINI JR, A. A. & STEFFEN JR, V. Modeling and Characterization of a Flexible Rotor Supported by Active Magnetic Bearings Using Model Reduction Techniques. **Proceedings of XVIII International Symposium on Dynamic Problems of Mechanics (DINAME)**, Búzios, 2019.

PEREIRA, B. L., LIMA, L., SILVA, G. D. & SANCHES, L. Aplicação da Evolução Diferencial na Identificação dos Parâmetros do Modelo de um Veículo de Duas Rodas do Tipo Pêndulo Invertido com Zona Morta. **Congresso Nacional de Engenharia Mecânica (CONEM)**, Salvador, 2018.

SILVA, G.D., PEREIRA, B.L., SOUSA, A.R., SANCHES, L., TAVARES, J.J.P.Z.S., MUROFUSHI, R. H. & CARDOSO, R.N. Nonlinear Modeling, Simulation and Control of a Two-Wheeled Inverted Pendulum. **International Congress of Mechanical Engineering (COBEM)**, Curitiba, 2017.

PEREIRA, B.L., SANCHES, L. & TAVARES, J.J.P.Z.S. Neuro-Fuzzy Control and Differential Evolution in the Compensation of the Actuators' Dead Zone of a Two-Wheeled Vehicle. **International Congress of Mechanical Engineering (COBEM)**, Curitiba, 2017.

Trabalhos Publicados como Capítulos de Livro

SANT'ANA, V.T., PEREIRA, B.L., MORAIS, T. & FINZI, R.M. Aplicação de Sistemas Neuro-Fuzzy na Predição do Coeficiente de sustentação do Aerofólio NACA 1412. **Impactos das Tecnologias na Engenharia Mecânica 2**. Ponta Grossa: Atena, 2019. p. 190-194.

MARCOS, W. P., MUROFUSHI, R.H. & PEREIRA, B. L. Identificação Experimental e Projeto de um PID para um Servomecanismo. **Impactos das Tecnologias na Engenharia Mecânica 2**. Ponta Grossa: Atena, 2019. p. 83-97.

Trabalhos em Periódicos

JAFELICE, R.M., PEREIRA, B.L., BERTONE, A.M.A & BARROS, L.C. An Epidemiological Model for HIV Infection in a Population Using Type-2 Fuzzy Sets and Cellular Automaton. **Computational & Applied Mathematics**, 2019. 38: 141. <https://doi.org/10.1007/s40314-019-0867-8>.

PEREIRA, B. L., DINIZ, G., SANCHES, L. & TAVARES, J.J.P.Z.S. Online Modified Differential Evolution Method: Static Stability Control of Two-Wheeled Inverted Pendulum Vehicle Case Study. **Brazilian Journal of Development**. (Artigo Aceito).

PEREIRA, B. L., SANT'ANA, V.T., JAFELICE, R.S.M. & FINZI, R.M. Pondered Individual Analysis: a New Fuzzy Inference Method - Application in Aerodynamic Modeling. **Neural Computing and Applications**. (Artigo Submetido).

PEREIRA, B. L., JAFELICE, R.S.M. & FINZI, R.M. An Approach of Pondered Individual Analysis Method in Aircraft Control. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**. (Artigo Submetido).

REFERÊNCIAS BIBLIOGRÁFICAS

ADIPRAWITA, W., AHMAD, A. & SEMBIRING, J. Automated Flight Test and System Identification for Rotary Wing Small Aerial Platform Using Frequency Responses Analysis. **Journal of Bionic Engineering**. 2008. 4. 237-244. 10.1016/S1672-6529(07)60037-7.

AHMAD, G., KHAN, M.A., ABBAS, S., ATHAR, A., KHAN, B.S. & ASLAM, M.S. Automated Diagnosis of Hepatitis B Using Multilayer Mamdani Fuzzy Inference System, **Journal of Healthcare Engineering**, vol. 2019, 11p., 2019. <https://doi.org/10.1155/2019/6361318>.

ALMEIDA, L. & CORREA, C. Percepções sobre os jogos de simulação de voo na formação de pilotos privados de avião. **Renote**. 2017.

ASCIITABLE, **ASCII Table and Description**, 2010. Acessado em: 7 de março de 2018. Disponível em: <https://www.asciitable.com>.

BARROS, L. C., BASSANEZI, R. C. & LODWICK, W. **A First Course in Fuzzy Logic, Fuzzy Dynamical Systems and Biomathematics Theory and Applications**. 1 ed. Springer, 2016.

BEMANI, A. & AKBARZADEH, M. A hybrid adaptive granular approach to Takagi-Sugeno-Kang fuzzy rule discovery. **Applied Soft Computing**. 81. 2019. 105491. 10.1016/j.asoc.2019.105491.

BERNARDES, R. **Sistemas Neuro-Fuzzy Aplicados no Controle de uma Aeronave de Asa Fixa**, Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2019.

BITTAR, A. **Piloto automático para VANT's: Tutorial X-Plane & Simulink**. Instituto Tecnológico de Aeronáutica (ITA). 2011.

BITTAR, A. & OLIVEIRA, N. M. F. Hardware-In-the-Loop Simulation of an Attitude Control with Switching Actuators for SUAV. **ICUAS**, p. 09, 2013.

BITTAR, A., FIGUEREIDO, H. V., GUIMARÃES, P. A. & MENDES, A. C. Guidance Software-in-The-Loop Simulation Using X-Plane and Simulink for UAVs. **ICUAS**, p. 10, 2014.

BITTAR, A., OLIVEIRA, N. M. F. & FIGUEREIDO, H. V. Hardware-In-the-Loop Simulation with X-Plane of Attitude Control of a SUAV Exploring Atmospheric Conditions. **Journal of Intelligent & Robotic Systems**, p. 17, 2014.

BLEJ, M. & AZIZI, M. **Comparison of Mamdani-type and Sugeno-type fuzzy inference systems for fuzzy real time scheduling**, 2016. 11. 11071-11075.

BRANDON, J. M. & MORELLI, E. A. **Nonlinear Aerodynamic Modeling From Flight Data Using Advanced Piloted Maneuvers and Fuzzy Logic**. NASA, p. 25, 2012.

CHEN, B. S & CHENG, Y. M. A Structure-Specified H^∞ Optimal Control Design for Practical Applications: A Genetic Approach. **IEEE Transactions On Control Systems Technology**, p. 12, 1998.

CHEN, C. H.; LIN, C. J. & LIN, C. T. Nonlinear System Control Using Adaptive Neural Fuzzy Networks Based on a Modified Differential Evolution. **IEEE Transactions On Systems, Man, And Cybernetics—part C: Applications And Reviews**, p. 15, 2009.

CHEN, M., SHI, P. & LIM, C. C. Adaptive Neural Fault-Tolerant Control of a 3-DOF Model Helicopter System. **IEEE Transactions On Systems, Man, And Cybernetics: Systems**, p.11, 2016.

CHEN, X.W., ZHANG, J.G. & LIU, Y.J. Research on the Intelligent Control and Simulation of Automobile Cruise System Based on Fuzzy System, **Mathematical Problems in Engineering**, p. 12, 2016. <https://doi.org/10.1155/2016/9760653>.

COX, D. E., CUNNINGHAM, K. & JORDAN, T. **Subscale Flight Testing for Aircraft Loss of Control: Accomplishments and Future Directions**. NASA, p. 13, 2012.

CRAIGHEAD, J., MURPHY, R., BURKE, J. & GOLDIEZ, B. A Survey of Commercial & Open-Source Unmanned Vehicle Simulators. **IEEE International Conference on Robotics and Automation**, p. 6, 2007.

CUI, G. & ZHANG, H. Hardware-in-the-loop simulation of flight mach fuzzy controller. **Wireless Networks**. 2021. 10.1007/s11276-019-02239-5.

CUI, R., YANG, C., LI, Y. & SHARMA, S. Adaptive Neural Network Control of AUVs With Control Input Nonlinearities Using Reinforcement Learning, **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, vol. 47, no. 6, pp. 1019-1029, 2017, doi: 10.1109/TSMC.2016.2645699.

EYOH, I & UMOH, U. A Comparative Analysis of Fuzzy Inference Engines in Context of Profitability Control. **West African Journal of Industrial & Academic Research**. 9. 13 – 27, 2013.

FAHMY, R., ZAHER, S. & KANDIL, A. A Comparison between Fuzzy Inference Systems for Prediction (with Application to Prices of Fund in Egypt). **International Journal of Computer Applications**, 109, 6-11, 2015. 10.5120/19246-0604.

FIGUEIREDO, H.V. & SAOTOME, O. Modelagem e simulação de veículos aéreos não tripulados (VANT) do tipo quadricóptero usando o simulador X-Plane e Simulink. **Anais do XIX Congresso Brasileiro de Automática (CBA)**, 2012.

FOSSEN, T.I. **Mathematical models for control of aircraft and satellites**. Department of Engineering Cybernetics Norwegian University of Science and Technology, 2011.

FRINK, T. N., MURPHY, P. C., ATKINS, H. L., VIKEN, S. A. & PETRILLI, J. L. **Status of Computational Aerodynamic Modeling Tools for Aircraft Loss-of-Control**. NASA, p. 25, 2017.

GARCIA, R., & BARNES, L. Multi-UAV simulator utilizing x-plane. **Journal of Intelligent and Robotic Systems**. 57. 393-406. 2010. 10.1007/978-90-481-8764-5_20.

GOMIDE, F. A. C., GUDWIN, R.R. & TANSCHKEIT, R. **Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações**, Unicamp, 2015, 39 p.

HAN, M.F., LIN, C. T. & CHANG, J.Y. Differential Evolution with Local Information for Neuro-Fuzzy Systems Optimization. **Knowledge-Based Systems**, Taiwan, v. 44, p. 78-89, 2013. <https://doi.org/10.1016/j.knosys.2013.01.023>.

HAYKIN, S. **Redes neurais: Princípios e Prática**. 2 ed. Canada: Bookman, 2008. 899 p.

HU, X., XU, B. & HU, C. Robust Adaptive Fuzzy Control for HFV With Parameter Uncertainty and Unmodeled Dynamics, **IEEE Transactions on Industrial Electronics**, vol. 65, no. 11, pp. 8851-8860, 2018, doi: 10.1109/TIE.2018.2815951.

IASBECK, A.H., PEREIRA, B.L & SANCHES, L. UDP Communication Between MATLAB and X-Plane 11. **Anais do XXIII Encontro Nacional de Modelagem Computacional (ENMC)**, Palmas, 2020.

IATA. **Loss of Control In-Flight: Accident Analysis Report**. p. 42, 2015.

IATA. **Safety Report 2019**. p. 262, 2020.

JAFELICE, R. S. M. **Modelagem Fuzzy para Dinâmica de Transferência de Soropositivos para HIV em Doença Plenamente Manifesta**, tese de doutorado, UNICAMP, 2003. 187p.

JAMSHIDI, A., CHAMZINI, A., YAKHCHALI, S. & KHALEGHI, S. Developing a new fuzzy inference system for pipeline risk assessment. **Journal of Loss Prevention in the Process Industries**. 26. 197–208. 2013. 10.1016/j.jlp.2012.10.010.

JANG, J.S. ANFIS Adaptive-Network-based Fuzzy Inference System. **IEEE Transactions on Systems, Man and Cybernetics**, 1993. 23. 665 - 685. 10.1109/21.256541.

JEE, T., TAY, K.M. & LIM, C. A New Two-Stage Fuzzy Inference System-Based Approach to Prioritize Failures in Failure Mode and Effect Analysis. **IEEE Transactions on Reliability**. 64. 869-877. 2015. 10.1109/TR.2015.2420300.

JHA, S. K., GAUR, P. & YADAV, A. K. Various Intelligent Control Techniques for Attitude Control of an Aircraft System. **3rd International Conference on Computing for Sustainable Global Development**, p. 5, 2016.

JORDAN, T. & BAILEY, R. **NASA Langley's AirSTAR Testbed: A Subscale Flight Test Capability for Flight Dynamics and Control System Experiments**, 2008. 10.2514/6.2008-6660.

JORDAN, T., FOSTER, J., BAILEY, R. & BELCASTRO, C. **AirSTAR: A UAV Platform for Flight Dynamics and Control System Testing**, 2006. 10.2514/6.2006-3307.

KHALIL, H. K. **Nonlinear Systems**, 3rd Edition, Prentice Hall, Englewood Cliffs, NJ, 2002.

KERK, Y.W., TAY, K.M. & LIM, C.P. Monotone Interval Fuzzy Inference Systems, **IEEE Transactions on Fuzzy Systems**, vol. 27, no. 11, pp. 2255-2264, 2019. doi: 10.1109/TFUZZ.2019.2896852.

KIM, E., KANG, H. J. & PARK, M. Numerical Stability Analysis of Fuzzy Control Systems Via Quadratic Programming and Linear Matrix Inequalities. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, 29(4), pp. 333-346, 1999, doi: 10.1109/3468.769752.

KIM, E. A New Approach to Numerical Stability Analysis of Fuzzy Control Systems. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, 31(1), pp. 107-113, 2001, doi: 10.1109/5326.923273.

KOSKO, B. Fuzzy Systems as Universal Approximators. **IEEE Transactions on Computers**, 43(11), 1994.

KURNAZ, S.; CETIN, O. & KAYNAK, O. Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. **Expert Systems with Applications**, p. 6, 2010.

KWONG, W. A., PASSINO, K. M., LAUKONEN, E. G. & YURKOVICH, S. Expert Supervision of Fuzzy Learning Systems for Fault Tolerant Aircraft Control. **Proceedings of the IEEE**, 1995. 18p.

LANDAU, I.D. *et al.* **Adaptive Control: Algorithms, Analysis and Applications**, EUA: Springs, 2011, 34 p. <https://doi.org/10.1007/978-0-85729-664-1>.

LARSEN, P.M. Industrial Applications of Fuzzy Logic Control. **International Journal of Man-Machine Studies**, 12:1, 1980. [https://doi.org/10.1016/S0020-7373\(80\)80050-2](https://doi.org/10.1016/S0020-7373(80)80050-2).

LEWIS, J. **X-plane, UDP, and Visual Basic, for X-Plane version 9**, 2009. Acessado em: 19 de janeiro de 2018. Disponível em: http://www.jefflewis.net/XPlaneUDP_9.html.

LI, D. Fuzzy PID Controller to control the attitude of Quadrotor UAV, **Gyancity Journal of Engineering and Technology**, v. 6, p. 1-11, 2020.

LI, H., WANG, J., WU, L., LAM, H.K. & GAO, Y. Optimal Guaranteed Cost Sliding Mode Control of Interval Type-2 Fuzzy Time-Delay Systems. **IEEE Transactions on Fuzzy Systems**. 2018. 10.1109/TFUZZ.2017.2648855.

LIU, D.M., NAADIMUTHU, G. & LEE, E. S. Trajectory Tracking in Aircraft Landing Operations Management Using the Adaptive Neural Fuzzy Inference System. **Computers and Mathematics with Applications**, 2008. 6p.

MACHINI, F.M.M. **Modelagem, Simulação e Controle de uma Aeronave Multirrotora**, Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2016.

MCLEAN, D. **Automatic Flight Control Systems**. Prentice Hall Inc. 1990.

MENDEL, J. M. **Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions**. 2 ed. Springer, 2017.

MIZUMOTO, M. Realization of PID Controls by Fuzzy Control Methods. **Fuzzy Sets and Systems**, 70(2-3), 171-182, 1995. [https://doi.org/10.1016/0165-0114\(94\)00215-S](https://doi.org/10.1016/0165-0114(94)00215-S).

MONTGOMERY, D.C. & RUNGER, G.C. **Applied Statistics and Probability for Engineers**. 3 ed. Arizona, EUA: John Wiley & Sons, 2002. 706 p.

MÓR, F.N., CORTES, O.A.C.C. & MARCON, C.A.M. **Aborgadem Paralela da Evolução Diferencial em GPU**. Gramado, RS: XV Escola Regional de Alto Desempenho, 2015. 11 p.

MURCH, A., COX, D. & CUNNINGHAM, K. **Software Considerations for Subscale Flight Testing of Experimental Control Laws**, 2009. 10.2514/6.2009-2054.

OLIVEIRA, D. D. **Evaluation of Flight Simulators Qualification Requirements for Full Stall Training Tasks Impacts on Safety and Industry**. 2018. 109 f. UFU, Uberlândia, 2018.

PEREIRA, B.L., SOUZA, M.R.S.B., ROCHA, R.R. & TAVARES, J.J.P.Z.S. Two-Wheeled Vehicle Control Using Neural Network and Differential Evolution. **Postgraduate Symposium in Mechanical Engineering**, 2016.

PRECUP, R.E., TOMESCU, M. & PREITL, S. Fuzzy Logic Control System Stability Analysis Based on Lyapunov's Direct Method. **Int. J. of Computers**, 2009. IV. 415-426.

RABOACA, DUMITRESCU & MANTA. Aircraft Trajectory Tracking Using Radar Equipment with Fuzzy Logic Algorithm. **Mathematics**. 8. 207. 2020. 10.3390/math8020207.

RADHAKRISHNAN, C. & SWARUP, A. Performance Comparison for Fuzzy based Aircraft Pitch using Various Control Methods, **2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)**, 2020, pp. 428-433, doi: 10.1109/ICIRCA48905.2020.9183199.

RIBEIRO, L.R. & OLIVEIRA, N.M.F. UAV autopilot controllers test platform using Matlab/Simulink and X-Plane, **IEEE Frontiers in Education Conference (FIE)**, 2010, pp. S2H-1-S2H-6, doi: 10.1109/FIE.2010.5673378.

ROSKAM, J. **Airplane Flight Dynamics and Automatic Flight Controls**, University of Kansas, v. 1, 2001. 618p.

SANT'ANA, V.T. **Aerodynamic Modeling Using Neuro-Fuzzy for a Scaled Model of Cessna 182**, Monografia, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2019.

SCOTT, C. & GONZALEZ, O. **On the Development of a Fuzzy Logic Model-less Aircraft Controller**. 2020. 10.2514/6.2020-0765.

SELVACHANDRAN, G., QUEK, S., LAN, L., SON, L., GIANG, N., ABDEL-BASSET, M., & ALBUQUERQUE, V. (2019). A New Design of Mamdani Complex Fuzzy Inference System for Multi-attribute Decision Making Problems. **IEEE Transactions on Fuzzy Systems**, 2019. PP. 10.1109/TFUZZ.2019.2961350.

SHLEEG, A & ELLABIB, I. Comparison of Mamdani and Sugeno fuzzy interference systems for the breast cancer risk. **International Journal of Computer, Information, Systems and Control Engineering**, 7, 2013.

SIDDIQUE, N., & ADELI, H. **Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing**. John Wiley & Sons, Inc, 2013.

SILVA, G. V. M. **Controlo Não Linear**, Lisboa v.1, 2006. 227p.

SON, L.H., VIET & P., HAI, P. Picture inference system: a new fuzzy inference system on picture fuzzy set. **Applied Intelligence**, 46(3), 652-669, 2017.

STALLINGS, W. **Arquitetura de organização de computadores**. Prentice Hall, 5ª ed., 2003.

STORN, R. & PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. **Journal of Global Optimization**, Netherlands, v. 11, p. 341-359, 1997.

TAKAGI, T. & SUGENO, M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. **IEEE Transactions on Systems, Man, and Cybernetics**, Japan, v. 15, n. 1, p. 116-132, jan./fev. 1985.

THUMS, G. D., TORRES, L. A. B. & PALHARES, R. M. Metodologia de Sintonia PID Multi-Malha para Veículos Aéreos Não Tripulados: Dinâmica Longitudinal. **Congresso Brasileiro de Automática**, p. 8, 2012.

ULUS, Ş. & ESKI, İ. Neural network and fuzzy logic-based hybrid attitude controller designs of a fixed-wing UAV. **Neural Computing and Applications**, p.1-23, 2021, 10.1007/s00521-020-05629-5.

VALLE, M. E. **O Método de Inferência de Takagi-Sugeno para Sistemas Baseados em Regras Fuzzy**. Unicamp, 2015, 43 p.

WANG, H., SUN, W. & LIU, P. X. Adaptive Intelligent Control of Nonaffine Nonlinear Time-Delay Systems with Dynamic Uncertainties, **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, vol. 47, no. 7, pp. 1474-1485, 2017, doi: 10.1109/TSMC.2016.2627048.

WANG, Y., SHI, Y., CAI, M. & XU, W. Predictive control of air-fuel ratio in aircraft engine on fuel-powered unmanned aerial vehicle using fuzzy-RBF neural network, **Journal of the Franklin Institute**, Volume 357, Issue 13, 2020, p. 8342-8363, <https://doi.org/10.1016/j.jfranklin.2020.03.016>.

WU, N., ZHANG, Y. & ZHOU, K. Detection, estimation, and accommodation of loss of control effectiveness. International Journal of Adaptive Control and Signal Processing. **INT J ADAPT CONTROL SIGNAL**. 775-795. 2000. 10.1002/1099-1115(200011)14:73.3.CO;2-W.

XU, B., ZHANG, Q. & PAN, Y. Neural network based dynamic surface control of hypersonic flight dynamics using small-gain theorem. **Neurocomputing**, p. 10, 2016.

XU, B. & SUN, F. Composite Intelligent Learning Control of Strict-Feedback Systems With Disturbance, **IEEE Transactions on Cybernetics**, vol. 48, no. 2, pp. 730-741, 2018, doi: 10.1109/TCYB.2017.2655053.

ZABIDIN, Y.A.A., PAIRAN, M.F. & SHAMSUDIN, S.S. Dynamic Modelling and Control for Quadcopter UAV with LabVIEW and X-Plane Flight Simulator. **Journal of Complex Flow**, 2020.

ZADEH, L. A. Fuzzy Sets. **Information and Control**, Berkeley, v. 8, p. 338-353, mar. 1965.

ZHANG, P. & SHEN, Q. A Novel Framework of Fuzzy Rule Interpolation for Takagi-Sugeno-Kang Inference Systems. **IEEE International Conference on Fuzzy Systems**. 1-6. 2019. 10.1109/FUZZ-IEEE.2019.8858833.

ZHAO, Z., WANG, X., ZHANG, C., LIU, Z. & Yang, J., Neural Network Based Boundary Control of a Vibrating String System with Input Deadzone, **Neurocomputing**, V. 275, 2018, <https://doi.org/10.1016/j.neucom.2017.09.050>.

ZHOU, Q., LI, H., WU, C., WANG, L. & AHN, C.K. Adaptive Fuzzy Control of Nonlinear Systems with Unmodeled Dynamics and Input Saturation Using Small-Gain Approach, **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, vol. 47, no. 8, pp. 1979-1989, 2017, doi: 10.1109/TSMC.2016.2586108.

ZHOU, Q., WU, C. & SHI, P. Observer-based adaptive fuzzy tracking control of nonlinear systems with time delay and input saturation. **Fuzzy Sets and Systems**. 2017. 316. 10.1016/j.fss.2016.11.002.

APÊNDICE

A) Filtro de Kalman

O filtro de Kalman foi desenvolvido no ano de 1960 e até hoje é o método estocástico mais utilizado na filtragem de sinais. Ele se baseia em um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático é minimizado. Através da observação da variável denominada variável de observação, outra variável (não observável) denominada variável de estado pode ser estimada eficientemente.

Para o estudo do filtro de Kalman, utiliza-se a referência (Brown e Hwang, 2012), e alguns conceitos referentes ao tema são apresentados a seguir.

- **Variável de Estado (x):**

- Variável relacionada à dinâmica de um sistema, baseada em um modelo matemático. Relacionada também à predição de dados futuros.
- O valor de x na iteração posterior é o valor na iteração atual com uma correção ϕ_k e corrompido por um sinal w_k de natureza aleatória (sequência branca) e com variância Q , Eq. (A.1):

$$x_{k+1} = \phi_k x_k + w_k. \quad (\text{A.1})$$

- **Variável de Observação (z):**

- Variável relacionada à aquisição de dados de um sensor;
- O valor obtido pelo sensor no tempo k é o valor da variável de estado com uma correção H_k e somado com ruído branco v_k com variância R , Eq. (A.2):

$$z_k = H_k x_k + v_k. \quad (\text{A.2})$$

- **Propagação:**

- É a predição do valor das variáveis em iterações seguintes baseada nos valores atuais destas e no modelo matemático do sistema.

- **Atualização:**

- É a correção da predição utilizando os novos dados coletados pelos sensores.

- **Variância do Erro (P):**

- O erro de estimação do filtro de Kalman em uma iteração k é definido como sendo a diferença entre o valor da variável de estado obtido pelo modelo (x_k) e o valor estimado pelo filtro (\hat{x}_k).
- Como o valor do erro flutua em torno de zero durante a execução do filtro (e portanto, tem média nula), a variância do erro é equivalente ao seu erro quadrático. Dessa forma, para se obter as equações do filtro de Kalman, é calculada a derivada de P em relação a K (ganho do filtro) e igualada a zero, de modo a minimizar de forma ótima o valor do erro quadrático.

No tópico a seguir são indicadas as matrizes e vetores que expandem os conceitos acima para problemas em que há mais de uma variável de estado. Além disso, são apresentadas as equações do filtro de Kalman.

A.1 Expansão do Método para Problemas com Mais de Uma Variável de Estado e Equações do Filtro de Kalman

Seguem as matrizes e vetores utilizados nas equações do filtro de Kalman:

- x_k ($n \times 1$): Vetor de estados do processo no tempo t_k ;
- ϕ_k ($n \times n$): Matriz de transição de estados, relacionando x_k com x_{k+1} ;
- w_k ($n \times 1$): Vetor cujos elementos são sequências brancas e com matriz de covariância conhecida; Resposta em t_{k+1} devido à presença de ruído branco na entrada do sistema no intervalo (t_k, t_{k+1}) ;
- z_k ($m \times 1$): Vetor de observações no tempo t_k ;
- H_k ($m \times n$): Matriz que relaciona x_k com z_k ;
- v_k ($m \times 1$): Vetor cujos elementos são sequências brancas e com matriz de covariância conhecida, sendo que cada elemento está relacionado a uma das variáveis de observação. Para obter as equações do filtro, adota-se que a correlação entre v_k e w_k é nula;
- P_k ($n \times n$): Matriz de covariância do erro;
- Q_k ($n \times n$): Matriz de covariância de w_k , $Q_k = E[w_k, w_k^T]$;
- R_k ($m \times m$): Matriz de covariância de v_k , $R_k = E[v_k, v_k^T]$;

- K_k ($n \times m$): Ganho do filtro de Kalman;
- I ($n \times n$): Matriz Identidade.

A seguir são apresentadas as equações do filtro de Kalman. Os valores a priori (-) são aqueles que ainda não passaram pelas equações de atualização de sua iteração correspondente, já os valores a posteriori (+) são aqueles que já passaram por esse processo:

- **Equações de Atualização:**

- Ganho do Filtro de Kalman:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}. \quad (\text{A.3})$$

- Atualização do Estado:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-]. \quad (\text{A.4})$$

- Atualização da Matriz de Covariância do Erro:

$$P_k^+ = (I - K_k H_k) P_k^-. \quad (\text{A.5})$$

- **Equações de Propagação:**

- Propagação do Estado:

$$\hat{x}_{k+1}^- = \phi_k \hat{x}_k^+. \quad (\text{A.6})$$

- Propagação da Matriz de Covariância do Erro:

$$P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k. \quad (\text{A.7})$$

Após terem sido apresentadas as equações do filtro de Kalman, no próximo tópico é abordado como se determinar os parâmetros deste.

A.2 Determinação dos Parâmetros do Filtro de kalman

Durante o projeto do filtro de Kalman, é desejável ter a representação do processo escrita na forma vetorial. A partir do processo em estudo, é possível reescrevê-lo da forma:

$$\dot{x} = Fx + Gu, \quad (\text{A.8})$$

em que x é o vetor de estados do processo, \dot{x} é sua derivada e u é o vetor de perturbação do sistema, que devido a critérios matemáticos do filtro de Kalman, corresponde a uma entrada de ruído branco unitário.

Alguns dos parâmetros do filtro são determinados de forma mais direta, como o valor da matriz H_k , que relaciona as variáveis de observação e os elementos correspondentes do vetor de estados x ; e os valores dos elementos da matriz R_k , que quando é assumido que os

dados provenientes de cada um dos sensores utilizados não são correlacionados, são determinados pelo cálculo da variância dos dados obtidos por cada um destes.

Por outro lado, o cálculo das matrizes ϕ_k e Q_k depende do modelo do sistema, e tornam-se de difícil obtenção quando os elementos das matrizes F e G não são constantes.

A seguir são apresentados o método analítico e métodos numéricos (relativamente simples e eficazes, mas restritos ao caso de os elementos de F e G serem constantes) para se obter esses parâmetros:

A.2.1 Método Analítico

- **Obtenção de ϕ_k :**

É possível obter o valor de ϕ_k através da equação (A.9), sendo \mathcal{L}^{-1} o operador relativo à inversa da transformada de Laplace, e Δt a taxa de amostragem do sistema:

$$\phi_k = \{\mathcal{L}^{-1}[(sI - F)^{-1}]\}_{t=\Delta t}. \quad (\text{A.9})$$

- **Obtenção de Q_k :**

Baseando-se no modelo matemático do sistema – Eq. (A.8), a Eq. (A.1) também pode ser escrita na forma:

$$x(t_{k+1}) = \phi(t_{k+1}, t_k)x(t_k) + \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \tau)G(\tau)u(\tau)d\tau. \quad (\text{A.10})$$

Desse modo, conforme pode ser verificado abaixo, é possível obter a matriz Q_k :

$$Q_k = E[w_k, w_k^T],$$

$$Q_k = E \left\{ \left[\int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \xi)G(\xi)u(\xi)d\xi \right] \left[\int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \eta)G(\eta)u(\eta)d\eta \right]^T \right\},$$

$$Q_k = \int_{t_k}^{t_{k+1}} \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \xi)G(\xi)E[u(\xi)u^T(\eta)]G^T(\eta)\phi^T(t_{k+1}, \eta)d\xi d\eta. \quad (\text{A.11})$$

A.2.2 Método Numérico

- **Obtenção de ϕ_k :**

Para se obter o valor de ϕ_k , pode-se partir da Eq. (A.8) e impor $u = 0$ a fim de que a equação descreva a dinâmica natural do sistema. Assim, temos:

$$\dot{x} = Fx. \quad (\text{A.12})$$

Verifica-se que a resolução desta equação pode ser escrita da forma:

$$\frac{x(t_2)}{x(t_1)} = \phi(t_2, t_1) = e^{F(t_2 - t_1)}. \quad (\text{A.13})$$

A partir da Eq. (A.13), pode-se concluir que:

$$\phi_k = \frac{x_{k+1}}{x_k} = e^{F\Delta t}. \quad (\text{A.14})$$

A exponencial matricial que aparece na equação anterior pode ser resolvida pelo comando do MATLAB $\text{expm}(F\Delta t)$, que a expande em série – Eq. (A.15) – e a trunca.

$$e^{F\Delta t} = I + F\Delta t + \frac{(F\Delta t)^2}{2!} + \dots. \quad (\text{A.15})$$

- **Obtenção de Q_k :**

- Primeiramente, forme uma matriz $2n \times 2n$ que se chama A (n é o número de elementos do vetor de estados x).

$$A = \begin{bmatrix} -F & GG^T \\ 0 & F^T \end{bmatrix} \Delta t. \quad (\text{A.16})$$

- Faça e^A e chame de B .

$$B = \begin{bmatrix} \dots & \phi_k^{-1} Q_k \\ 0 & \phi_k^T \end{bmatrix}. \quad (\text{A.17})$$

- Faça a transposição da parte inferior direita para verificar o valor de ϕ_k .
- Finalmente, Q_k é obtido através da parte superior direita de B , como segue:

$$Q_k = \phi_k x \text{ (Parte superior direita de } B \text{)}. \quad (\text{A.18})$$