

RENATA BERNARDES

**SISTEMAS NEURO-FUZZY APLICADOS NO CONTROLE
DE UMA AERONAVE DE ASA FIXA**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA

2019

RENATA BERNARDES

SISTEMAS NEURO-FUZZY APLICADOS NO CONTROLE DE UMA
AERONAVE DE ASA FIXA

Monografia apresentada
ao Curso de Graduação em
Engenharia Mecatrônica da
Universidade Federal de
Uberlândia, com parte dos
requisitos para a obtenção do
título de **BACHAREL EM
ENGENHARIA
MECATRÔNICA.**

Orientador: Prof. Dr. Roberto
Mendes Finzi Neto

Uberlândia – MG

2019



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
FACULDADE DE ENGENHARIA MECÂNICA
COORDENAÇÃO DO CURSO DE GRADUAÇÃO EM
ENGENHARIA MECATRÔNICA



ATA DE DEFESA DE PROJETO DE FIM DE CURSO II
NÚMERO DE ORDEM: 220/2019

Às quatorze horas do décimo nono dia do mês de fevereiro de dois mil e dezenove, no Bloco 1M, Sala de Reuniões da FEMEC, no Campus Santa Mônica, reuniu-se a Banca Examinadora composta pelo professor Roberto Mendes Finzi Neto, da Faculdade de Engenharia Mecânica, da Universidade Federal de Uberlândia, e os Engenheiros Bruno Luiz Pereira, e Felipe Machini Malachias Marques, alunos de doutorado da Pós-Graduação da FEMEC-UFU, sob a presidência do primeiro, para desenvolver o processo de avaliação do **Projeto de Fim de Curso II, intitulado "SISTEMAS NEURO-FUZZY APLICADOS NO CONTROLE DE UMA AERONAVE DE ASA FIXA"**, apresentado pela aluna **RENATA BERNARDES – MATRÍCULA 11411EMT012** em atendimento aos requisitos determinados pelo Projeto Pedagógico do Curso de Graduação em Engenharia Mecatrônica. Após discorrer sobre seu trabalho, o candidato foi arguido pelos membros da Banca, diante das comunidades universitária e externa. Em seguida, a defesa do trabalho foi avaliada em seção privada pelos membros da Banca que, ao encerrar o processo, consideraram aprovada. As demandas complementares observadas pelos examinadores deverão ser satisfeitas no prazo máximo de 30 (trinta) dias corridos a contar da data da defesa, para dar validade a esta aprovação. Para constar, lavrou-se a Ata que vai assinada pelo presidente e demais membros da Banca.

Assinaturas:

Prof. Dr. Roberto Mendes Finzi Neto / (Orientador)

Eng. MSc. Bruno Luiz Pereira

Eng. MSc. Felipe Machini Malachias Marques

19 de fevereiro de 2019

Agradecimentos

À minha mãe Adelzira Santos Costa Bernardes e ao meu pai Jairo Bernardes da Silva Júnior por todo apoio, paciência, compreensão durante o curso e por acreditarem tanto na minha capacidade.

Ao meu namorado Heitor Menezes Souza Cunha, pelo carinho, apoio, incentivo e paciência durante os momentos difíceis, pela oportunidade de amadurecimento e por me fazer enxergar sempre as situações com mais otimismo.

A todos os meus amigos e familiares que direta ou indiretamente contribuíram para minha formação, com palavras de conforto, carinho e conselhos.

Ao Prof. Dr. Leonardo Sanches e Prof. Dr. Roberto Mendes Finzi Neto pela oportunidade de poder trabalhar em um projeto de tamanha contribuição para a pesquisa nacional e internacional.

Ao Mestre e Doutorando Bruno Luiz Pereira pelo incentivo, motivação e orientação no desenvolvimento deste trabalho, bem como à oportunidade de aprender e contribuir para uma tese de doutorado de grande caráter inovador.

A cada membro do Laboratório de Aeronaves Autônomas da UFU pela motivação e disposição em ajudar para obtenção do melhor resultado acerca deste trabalho.

E, finalmente, à Universidade Federal de Uberlândia e à Faculdade de Engenharia Mecânica pela oportunidade de participar do curso de Engenharia Mecatrônica e por toda a infraestrutura oferecida.

Resumo

A perda de controle da aeronave durante o vôo é a principal causa de morte em acidentes aéreos. Devido a isso torna-se de suma importância o desenvolvimento de um sistema de controle eficiente, que muitas das vezes não consegue ser obtido através das técnicas clássicas. Neste trabalho é realizado o controle de uma aeronave de asa fixa utilizando Sistemas *Neuro-fuzzy*, com parâmetros obtidos através da Evolução Diferencial e do Recozimento Simulado. Utiliza-se o *software* MatLab® para a obtenção do controlador e para a avaliação do desempenho e estabilidade do sistema. Os resultados obtidos indicam que o controlador é capaz de compensar eventuais perturbações no sistema. Além disso, o Recozimento Simulado requer um tempo menor de execução para obter a solução desejada, em contrapartida, este método apresenta maior desvio padrão dos resultados, o que sugere maior aleatoriedade e imprecisão que o método da Evolução Diferencial.

Palavras-chave: controle inteligente, Sistemas Neuro-fuzzy, Evolução Diferencial, Recozimento Simulado, desempenho, estabilidade.

Abstract

The loss of the aircraft's control during the flight is the main cause of deaths in plane crashes. For this reason, it is of great importance to develop an efficient controller, that many times cannot be obtained through the basic techniques. In this project it is performed the control of a fixed-wing airplane using neuro-fuzzy systems, with parameters obtained through Differential Evolution and from Simulated Annealing. It is used MatLab® *software* to obtain the controller and to evaluate the system's performance and stability. The obtained results indicate that the controller is capable of compensating eventual system disturbances. Furthermore, the Simulated Annealing shows a shorter time of execution to find the optimum controller, by contrast, this method shows a higher standard deviation of the results, which suggests a higher randomness and imprecision than the Differential Evolution method.

Keywords: intelligent control, Neuro-fuzzy systems, Differential Evolution, simulated annealing, performance, stability.

Lista de Figuras

1	Percentual de acidentes por tipo de ocorrência nos últimos 10 anos (Cenipa,2017).	12
2	Definição dos eixos, velocidades, forças, momentos e ângulos de Euler da aeronave	17
3	Representação de uma RNA (Tafner, 1998)	19
4	Representação de um neurônio e suas respectivas analogias ao neurônio biológico (Tafner, 1998).	20
5	Esquema de um neurônio i da camada I (Silva, 2006).	20
6	Representação da arquitetura de um sistema Baseado em Regras <i>Fuzzy</i> (Jafelice, 2003).	22
7	Formas usuais utilizada para função de pertinência (Pacheco, 2015)	22
8	Ilustração do método de Mamdani (Jafelice, 2003).	23
9	Ilustração do método de Takagi-Sugeno (Jafelice, 2003)	24
10	Representação da arquitetura básica de um ANFIS (Fonte: www.maxwell.vrac.puc-rio.br)	25
11	Representação de um sistema generalizado ANFIS (Pereira, 2016).	25
12	Representação da evolução diferencial, adaptada de Mór et al. (2015).	26
13	Máximos e mínimos de uma função qualquer (Fonte: www.matematica.pt)	28
14	Algoritmo básico do Reconhecimento Simulado (Fonte: www2.peq.coppe.ufrj.br)	29
15	Sistema de controle Longitudinal	31
16	Sistema de controle Longitudinal – malha interna	31
17	Sistema de controle Látero-direcional	32
18	Sistema de controle Látero-direcional – malha interna	33
19	Entrada (Ângulo de Arfagem) e saída (Deflexão do Profundor) do sistema de controle	34
20	Erro RMS	35
21	Ângulo de Arfagem e Deflexão do Profundor utilizando Evolução Diferencial	47
22	Erro RMS da malha interna do Controle Longitudinal obtido utilizando Evolução Diferencial	48
23	Ângulo de Arfagem e Deflexão do Profundor utilizando Reconhecimento Simulado	48
24	Erro RMS da malha interna do Controle Longitudinal obtido utilizando Reconhecimento Simulado	49
25	Altura utilizando Evolução Diferencial	50
26	Deflexão do Profundor da malha externa utilizando Evolução Diferencial	51
27	Erro RMS da malha externa do Controle Longitudinal obtido utilizando Evolução Diferencial	51
28	Altura utilizando Reconhecimento Simulado	52
29	Deflexão do Profundor da malha externa utilizando Reconhecimento Simulado	52
30	Erro RMS da malha externa do Controle Longitudinal obtido utilizando Reconhecimento Simulado	53
31	Ângulo de Rolagem e Deflexão do Aileron utilizando Evolução Diferencial	54
32	Erro RMS da malha interna do Controle Látero-direcional obtido utilizando Evolução Diferencial	55
33	Ângulo de Rolagem e Deflexão do Aileron utilizando Reconhecimento Simulado	56
34	Erro RMS da malha interna do Controle Látero-direcional obtido utilizando Reconhecimento Simulado	57
35	Ângulo de Guinada utilizando Evolução diferencial	58
36	Deflexão do Aileron e Deflexão do Leme utilizando Evolução diferencial	59
37	Erro RMS da malha externa do Controle Látero-direcional obtido utilizando Evolução Diferencial	59

38	Ângulo de Guinada utilizando Recozimento Simulado	60
39	Deflexão do Aileron e Deflexão do Leme utilizando Recozimento Simulado	60
40	Erro RMS da malha externa do Controle Látero-direcional obtido utilizando Recozimento Simulado	61

Lista de Tabelas

1	Parâmetros utilizados na Evolução Diferencial	38
2	Parâmetros utilizados no Recozimento Simulado	40
3	Comparação entre Evolução Diferencial e Recozimento Simulado para malha interna do Controle Longitudinal	49
4	Comparação entre Evolução Diferencial e Recozimento Simulado para malha externa do Controle Longitudinal	53
5	Comparação entre Evolução Diferencial e Recozimento Simulado para malha interna do Controle Látero-direcional	57
6	Comparação entre Evolução Diferencial e Recozimento Simulado para malha externa do Controle Látero-direcional	61

Lista de Símbolos

- a_i^l - variável de saída do neurônio i da RNA
 b_i^l - bias associado ao neurônio i
 Cf - função custo
 $C_{i,k}$ - vetor de cruzamento da Evolução Diferencial
 CR - taxa de cruzamento
 dt - tempo de amostragem
 E^k - energia interna da solução k
 F - fator de perturbação
 f_{obj} - função objetivo
 h - altitude
 I_{CG} - momento de inércia do centro de gravidade da aeronave
 I_x - momento de inércia em relação ao eixo x
 I_y - momento de inércia em relação ao eixo y
 I_z - momento de inércia em relação ao eixo z
 I_{xz} - produto de inércia relativamente a x e z
 L - momento de rolagem da aeronave
 L_0 - valor nominal de momento de rolagem
 M - momento de arfagem da aeronave
 M_0 - valor nominal de momento de arfagem
 m - massa do sistema
 N - momento de guinada da aeronave
 N_0 - valor nominal de momento de guinada
 n_e - número de elementos dos vetores
 P - velocidade angular de rolagem da aeronave
 P_0 - valor nominal da velocidade angular de rolagem
 p - perturbação da velocidade angular de rolagem
 p_{RM} - probabilidade em se aceitar um ponto com maior energia interna
 Q - velocidade angular de arfagem da aeronave
 Q_0 - valor nominal da velocidade angular de arfagem
 q - perturbação da velocidade angular de arfagem
 R - velocidade angular de guinada da aeronave
 R_0 - valor nominal da velocidade angular de guinada
 r - perturbação da velocidade angular de guinada
 $R_{x,\Phi}$ - matriz de transformação do sistema do corpo para o sistema inercial na direção x
 $R_{x,\Phi}^T$ - matriz de transformação do sistema inercial para o sistema do corpo na direção x
 $R_{y,\Theta}$ - matriz de transformação do sistema do corpo para o sistema inercial na direção y
 $R_{y,\Theta}^T$ - matriz de transformação do sistema inercial para o sistema do corpo na direção y
 $R_{z,\Psi}$ - matriz de transformação do sistema do corpo para o sistema inercial na direção z
 T - constante de tempo T_a - tempo de acomodação do sistema
 T^k - temperatura no instante k
 U - velocidade longitudinal da aeronave
 U_0 - valor nominal de velocidade longitudinal
 u - perturbação da velocidade longitudinal
 V - Velocidade lateral da aeronave
 V_0 - valor nominal de velocidade lateral
 $V_{i,k}$ - vetor de mutação da Evolução Diferencial

v - perturbação da velocidade lateral
 W - velocidade vertical da aeronave
 W_0 - valor nominal de velocidade vertical
 w - perturbação da velocidade vertical
 w_{ij}^l - peso sináptico associado ao neurônio i
 X - força longitudinal
 X_0 - valor nominal de força longitudinal
 X_E - posição x do eixo inercial da terra
 $X_{i,k}$ - vetor de inicialização da Evolução Diferencial
 x_k - valor de x (variável qualquer) no instante k
 x_{ref} - valor de x (variável qualquer) de referência
 Y - força transversal
 Y_0 - valor nominal de força transversal
 Y_E - posição y do eixo inercial da terra
 y_r - parâmetro de controle
 y_s - saída do controlador
 Z - força vertical
 Z_0 - valor nominal de força vertical
 Z_E - posição z do eixo inercial da terra
 α - fator utilizado no decaimento da temperatura no Recozimento Simulado
 δ_{ail} - deflexão do aileron da aeronave
 δL - perturbação do momento de rolagem
 δ_{lem} - deflexão do leme da aeronave
 δM - perturbação do momento de arfagem
 δN - perturbação do momento de guinada
 δ_{pr} - deflexão do profundor da aeronave
 δX - perturbação da força longitudinal
 δY - perturbação da força transversal
 δZ - perturbação da força vertical
 Θ - ângulo de arfagem da aeronave
 Θ_0 - valor nominal do ângulo de arfagem
 θ - perturbação do ângulo de arfagem
 σ - função de ativação da rede neural
 $\vec{\tau}_1$ - forças do sistema
 $\vec{\tau}_2$ - momentos do sistema
 Φ - ângulo de rolagem da aeronave
 Φ_0 - valor nominal do ângulo de rolagem
 ϕ - perturbação do ângulo de rolagem
 Ψ - ângulo de guinada da aeronave
 Ψ_0 - valor nominal do ângulo de guinada
 ψ - perturbação do ângulo de guinada

Glossário

ANFIS – Sistema de Inferência Neuro-*Fuzzy* Adaptativo (*Adaptative neuro-fuzzy inference system*)

ANN – Redes Neurais Artificiais (*Artificial Neural Network*)

CENIPA - Centro de Investigação e Prevenção de Acidentes Aeronáuticos

FIS – Sistema de Inferência *Fuzzy* (*Fuzzy Inference System*)

IA – Inteligência Artificial

PID - Proporcional-Integral-Derivativo

RNA – Rede Neural Artificial

SBRF – Sistema Baseado em Regras *Fuzzy*

VANTS – Veículos aéreos não tripulados

Conteúdo

1	Introdução	12
2	Objetivos	14
3	Justificativa	15
4	Fundamentação Teórica	16
4.1	Mecânica do Voo	16
4.1.1	Equações de movimento da aeronave	16
4.1.2	Teoria das Perturbações	18
4.2	Redes Neurais	19
4.2.1	Conceitos Gerais	19
4.2.2	Treinamento da rede neural	21
4.3	Teoria de conjuntos <i>Fuzzy</i>	21
4.4	Sistema Neuro- <i>Fuzzy</i>	24
4.5	Evolução diferencial	26
4.6	Recozimento simulado	27
5	Metodologia	31
5.1	Controle Longitudinal	31
5.2	Controle Látero-direcional	32
5.3	Ajuste de parâmetros de controlador	33
5.4	Metodologia de Análise	34
6	Desenvolvimento	36
6.1	model.m	36
6.2	convc2d.m	36
6.3	Ev_dif.m	36
6.4	Rec_sim.m	38
6.5	Disturb.m	40
6.6	Setpoint.m	41
6.7	ControlNF.m	41
6.8	RespModelo.m	43
6.9	Refdet.m	43
6.10	Calcadapt.m	44
6.11	main.m	45
7	Resultados e Discussões	46
7.1	Controle Longitudinal	46
7.1.1	Controle Longitudinal – Malha Interna	46
7.1.1.1	Evolução Diferencial	46
7.1.1.2	Recozimento Simulado	48
7.1.2	Controle Longitudinal – Malha Externa	49
7.1.2.1	Evolução Diferencial	50
7.1.2.2	Recozimento Simulado	52
7.2	Controle Látero-direcional	53
7.2.1	Controle Látero-direcional – Malha Interna	53

7.2.1.1	Evolução Diferencial	54
7.2.1.2	Recozimento Simulado	55
7.2.2	Controle Látero-direcional - Malha Externa	57
7.2.2.1	Evolução Diferencial	58
7.2.2.2	Recozimento Simulado	59
8	Conclusão e trabalhos futuros	62
9	Referências Bibliográficas	63

1 Introdução

Nos países que possuem grande extensão territorial, como é o caso do Brasil, se faz necessário uma estrutura de transporte que permite a alta mobilidade geográfica da população e uma facilidade na distribuição de mercadorias, de forma a se vencer o relevo sem muita dificuldade. Observa-se que o desenvolvimento econômico do país nas últimas décadas tem como consequência um crescimento considerável da aviação brasileira. De acordo com estudos realizados pelo CENIPA (Cenipa, 2017) (Centro de Investigação e Prevenção de Acidentes Aeronáuticos), nos últimos 10 anos (2008-2017) houve 1187 acidentes aéreos registrados, com média de 119 acidentes por ano. O maior percentual de acidentes, representando 53% conforme Fig.1, tem como principais causas:

- Falha do motor em voo;
- Perda de controle no solo;
- Perda de controle em voo;

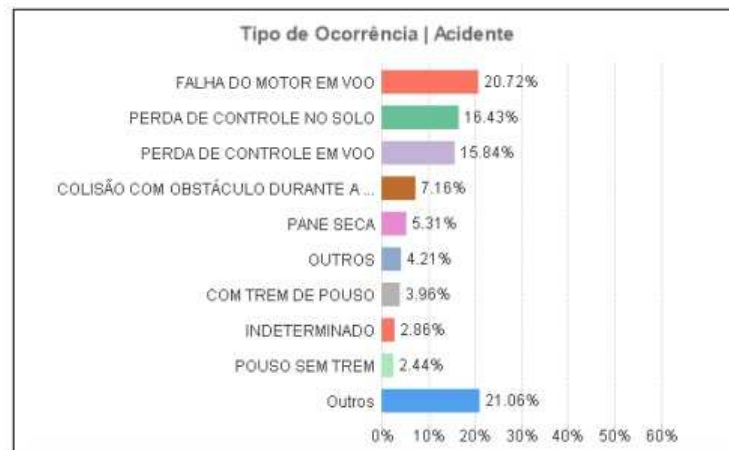


Figura 1: Percentual de acidentes por tipo de ocorrência nos últimos 10 anos (Cenipa,2017).

Além disso, registros mostram que o número de vítimas fatais é considerável. Dessa forma, é interessante que se desenvolva um controle de aeronaves que tenha como objetivo diminuir a ocorrência dos acidentes, garantindo assim, uma maior segurança para quem utiliza esse meio de transporte.

Devido a sua simplicidade e robustez, o controlador PID (Proporcional Integral Derivativo) é o principal método de controle utilizado hoje no mundo. Porém, quando dispomos de processos não lineares, que possuem variantes no tempo, estes controladores podem não responder de uma forma satisfatória (Gianelli,2013). Existem então outras técnicas de controle capazes de se adaptar ao sistema complexo e fornecer uma resposta otimizada, uma delas é utilizando a Inteligência Artificial (IA) para a obtenção do controlador e sua estrutura matemática.

A Inteligência artificial, sendo um ramo de pesquisa da ciência da computação, tem como objetivo construir algoritmos capazes de simular a inteligência dos seres vivos e aplicá-la no processamento de informações. Estes algoritmos devem, a partir da combinação de grandes quantidades de dados, de forma rápida e interativa, aprender com informações ou padrões apresentados nos dados e tomar decisões. Algumas tecnologias englobadas dentro do ramo da IA são:

- **Rede neural:** estrutura composta de unidades interconectadas, denominadas neurônios. Estas unidades processam os dados de entrada e retransmitem às unidades seguintes, até se obter a resposta desejada.
- **Machine learning:** processo de automatização da construção de modelos analíticos que utiliza da técnica de redes neurais aliada à estatística, pesquisa de operações e física para que sejam encontrados pontos de interesse nos dados.
- **Visão computacional:** utiliza do reconhecimento de padrões em imagens para identificar e interpretar o que existe ao redor.

Hoje existem alguns avanços na área, como a criação de uma tecnologia de aprendizagem desenvolvida por cientistas na Universidade de Berkley, Califórnia, que permite aos robôs que os mesmos imaginem o futuro de suas ações, possibilitando a tomada de decisões a partir do futuro previsto (O FUTURO DAS COISAS, 2017).

Neste projeto, a IA é de grande importância para criar um controlador que tem como principal característica a adaptabilidade, obtendo os parâmetros de forma eficaz e rápida. As técnicas utilizadas para a criação do controlador são apresentadas abaixo:

- **Sistema Neuro-*fuzzy*:** sistema de hibridização das técnicas de modelagem de Redes Neurais e Sistemas de Inferência *Fuzzy*.
- **Evolução Diferencial:** método robusto de minimização de funções, utilizado para obtenção dos parâmetros do Sistema Neuro-*fuzzy*.
- **Recozimento Simulado:** método de minimização de funções originado da analogia do recozimento de metais, também utilizado para obtenção dos parâmetros do Sistema *Neuro-fuzzy*.

O projeto a seguir contará com: objetivos, justificativa, fundamentação teórica, onde será realizada uma breve revisão bibliográfica acerca do tema, metodologia, desenvolvimento, resultados e discussões, conclusão e trabalhos futuros e referências bibliográficas.

2 Objetivos

Esta monografia está inserida em um projeto desenvolvido por alunos, professores, mestrandos e doutorandos da Universidade Federal de Uberlândia que tem como objetivo principal o desenvolvimento de um controlador adaptativo inteligente aplicado a uma aeronave Cessna 172 em escala reduzida, possibilitando sua total autonomia durante os vôos, que é garantida pela estabilidade e pelo desempenho do controlador.

Como objetivos específicos, tem-se a utilização de técnicas *neuro-fuzzy* frequentemente utilizada na inteligência artificial e, a obtenção dos parâmetros deverá ser feita utilizando dois métodos diferentes (Evolução Diferencial e Recozimento Simulado), os quais serão comparados ao fim deste trabalho. Também será realizada a avaliação do sistema quando submetido à perturbações em suas variáveis de estado.

3 Justificativa

De acordo com dados apresentados pelo CENIPA (Centro de Investigação e Prevenção de Acidentes Aeronáuticos, observados na Fig.1), uma das principais causas para os acidentes aéreos é a perda de controle da aeronave durante o voo. A aplicação de algoritmos de controle mais eficientes, visando melhorias em quesitos como performance e estabilidade, ajudaria a diminuir tais estatísticas, tornando aviões comerciais e VANTS mais seguros. Além disso, as vantagens do uso de controle inteligente se dão pela capacidade de adaptação do mesmo, oferecendo robustez às mudanças de cenários, como por exemplo, instabilidades climáticas, chuvas e vento forte.

É importante que, na graduação, tenha-se a aplicação na prática das disciplinas apresentadas de forma teórica e didática durante o curso. O projeto em questão engloba desde disciplinas mais básicas até as mais complexas, como Programação Aplicada à Engenharia, Controle de Sistemas Lineares, Controle Digital de Sistemas e Inteligência Computacional (oferecida como optativa), cumprindo a proposta de um Projeto de Fim de Curso.

Finalmente, vislumbrando o desenvolvimento da pesquisa nessa área, esse projeto está inserido na tese de doutorado do aluno Bruno Luiz Pereira, que vem com uma proposta inovadora e gerando uma grande e importante contribuição para o meio científico.

4 Fundamentação Teórica

4.1 Mecânica do Vôo

Esse item tem como objetivo fornecer uma breve introdução à modelagem dinâmica de uma aeronave. Para um estudo mais aprofundado pode-se utilizar como referência (Fossen, 2011) e (Roskam, 1997).

De acordo com Machini (2016) 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. Sendo assim, 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 capaz de transladar e rotacionar junto com a mesma B_{cs} . 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: arfagem (Θ), rolagem (Φ) e guinada (Ψ).

A definição das velocidades, posição ou ângulos e forças ou momentos podem ser visualizados a partir das Eq. (4.1), (4.2) e (4.3).

$$\vec{v} = \begin{pmatrix} U \\ V \\ W \\ P \\ Q \\ R \end{pmatrix} = \begin{pmatrix} \text{velocidade longitudinal} \\ \text{velocidade lateral} \\ \text{velocidade vertical} \\ \text{velocidade angular de rolagem} \\ \text{velocidade angular de arfagem} \\ \text{velocidade angular de guinada} \end{pmatrix} \quad (4.1)$$

$$\vec{\eta} = \begin{pmatrix} X_E \\ Y_E \\ Z_E, h \\ \Phi \\ \Theta \\ \Psi \end{pmatrix} = \begin{pmatrix} \text{posição x do eixo inercial da Terra} \\ \text{posição y do eixo inercial da Terra} \\ \text{posição z do eixo inercial da Terra, altitude} \\ \text{ângulo de rolagem} \\ \text{ângulo de arfagem} \\ \text{ângulo de guinada} \end{pmatrix} \quad (4.2)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{pmatrix} = \begin{pmatrix} \text{força longitudinal} \\ \text{força lateral} \\ \text{força vertical} \\ \text{momento rolagem} \\ \text{momento de arfagem} \\ \text{momento de guinada} \end{pmatrix} \quad (4.3)$$

A partir da Fig. 2 é possível visualizar os eixos, velocidades, forças, momentos e ângulos de Euler definidos anteriormente.

4.1.1 Equações de movimento da aeronave

A equação cinemática para translação e rotação de um corpo rígido fixo pode ser expressa da forma:

$$\begin{pmatrix} \dot{X}_E \\ \dot{Y}_E \\ \dot{Z}_E \end{pmatrix} = R_{z,\Psi} R_{y,\Theta} R_{x,\Phi} \begin{pmatrix} U \\ V \\ W \end{pmatrix} \quad (4.4)$$

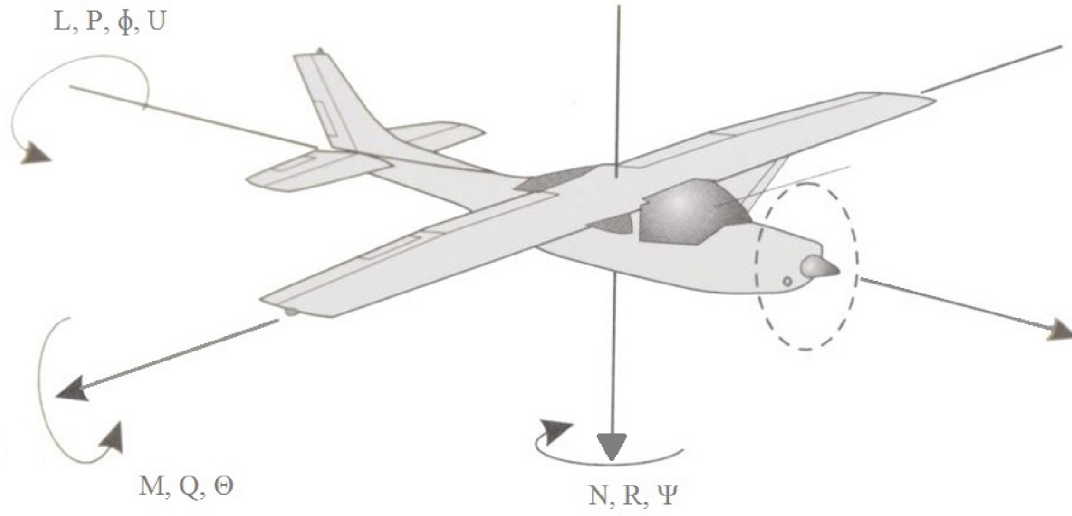


Figura 2: Definição dos eixos, velocidades, forças, momentos e ângulos de Euler da aeronave

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, 1997).

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

$$\begin{pmatrix} P \\ Q \\ R \end{pmatrix} = \begin{pmatrix} \dot{\Phi} \\ 0 \\ 0 \end{pmatrix} + R_{x,\Phi} \begin{pmatrix} 0 \\ \dot{\Theta} \\ 0 \end{pmatrix} + R_{x,\Phi} R_{y,\Theta} \begin{pmatrix} 0 \\ 0 \\ \dot{\Psi} \end{pmatrix} \quad (4.5)$$

Aplicando a segunda lei de Newton para um corpo rígido com 6 graus de liberdade tem-se que

$$m(\vec{v}_1 + \vec{v}_2 \times \vec{v}_1) = \vec{\tau}_1 \quad (4.6)$$

$$I_{CG}(\vec{v}_2 + \vec{v}_2 \times I_{CG} \vec{v}_2) = \vec{\tau}_2 \quad (4.7)$$

onde $\vec{v}_1 = [U, V, W]^T$, $\vec{v}_2 = [P, Q, R]^T$, $\vec{\tau}_1 = [Z, Y, Z]^T$ e $\vec{\tau}_2 = [L, M, N]^T$.

A partir do desenvolvimento (Fossen, 2011), tem-se

$$\begin{pmatrix} m(\dot{U} + QW - RV + g \sin(\Theta)) \\ m(\dot{V} + UR - WP - g \cos(\Theta) \sin(\Phi)) \\ m(\dot{W} + VP - QU - g \cos(\Theta) \cos(\Phi)) \\ I_x \dot{P} - I_{xz}(\dot{R} + PQ) + (I_z - I_y)QR \\ I_y \dot{Q} + I_{xz}(P^2 - R^2) + (I_x - I_z)PR \\ I_z \dot{R} - I_{xz}\dot{P} + (I_y - I_x)PQ + I_{xz}QR \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{pmatrix} \quad (4.8)$$

Por fim, as equações dinâmicas que representam o movimento da aeronave são dadas pelas Eq. (4.5) e (4.8), de forma que caracteriza um sistema possível determinado com nove variáveis e nove equações.

Essas equações são integradas ao longo do tempo a fim de obter a variação temporal das variáveis de estado.

4.1.2 Teoria das Perturbações

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

$$\vec{r} = \vec{r}_0 + \vec{\delta r} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{pmatrix} + \begin{pmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta L \\ \delta M \\ \delta N \end{pmatrix} \quad (4.9)$$

$$\vec{v} = \vec{v}_0 + \vec{\delta v} = \begin{pmatrix} U_0 \\ V_0 \\ W_0 \\ P_0 \\ Q_0 \\ R_0 \end{pmatrix} + \begin{pmatrix} u \\ v \\ w \\ l \\ m \\ n \end{pmatrix} \quad (4.10)$$

$$\begin{pmatrix} \Theta \\ \Phi \\ \Psi \end{pmatrix} = \begin{pmatrix} \Theta_0 \\ \Phi_0 \\ \Psi_0 \end{pmatrix} + \begin{pmatrix} \theta \\ \phi \\ \psi \end{pmatrix} \quad (4.11)$$

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

$$\begin{pmatrix} m(\dot{u} + Q_0 w + W_0 q - R_0 v - V_0 r + g \cos(\Theta_0)\theta) \\ m(\dot{v} + U_0 r + R_0 u - W_0 p - P_0 w - g \cos(\Theta_0) \cos(\Phi_0)\phi + g \sin(\Theta_0) \sin(\Phi_0)\theta) \\ m(\dot{w} + V_0 p + P_0 v - U_0 q - Q_0 u + g \cos(\Theta_0) \sin(\Phi_0)\phi + g \sin(\Theta_0) \cos(\Phi_0)\theta) \\ I_x \dot{p} - I_{xz} \dot{r} + (I_z - I_y)(Q_0 r + R_0 q) - I_{xz}(P_0 q + Q_0 p) \\ I_y \dot{q} + (I_x - I_z)(P_0 r + R_0 p) - 2I_{xz}(R_0 r + P_0 p) \\ I_z \dot{r} - I_{xz} \dot{p} + (I_y - I_x)(P_0 q + Q_0 p) + I_{xz}(Q_0 r + R_0 q) \end{pmatrix} = \begin{pmatrix} \delta X \\ \delta Y \\ \delta Z \\ \delta L \\ \delta M \\ \delta N \end{pmatrix} \quad (4.12)$$

Na Eq. (4.12), as variáveis que possuem sub-índice “0” são denominadas variáveis de equilíbrio ou de trimagem, as quais são obtidas assumindo o equilíbrio de forças e momentos nas Eq. (4.6) e (4.7).

$$\begin{pmatrix} m(Q_0 W_0 - R_0 V_0 + g \sin(\Theta_0)) \\ m(U_0 R_0 - P_0 W_0 - g \cos(\Theta_0) \sin(\Phi_0)) \\ m(P_0 V_0 - Q_0 U_0 - g \cos(\Theta_0) \cos(\Phi_0)) \\ (I_z - I_y)Q_0 R_0 - P_0 Q_0 I_{xz} \\ (P_0^2 - R_0^2)I_{xz} + (I_x - I_z)P_0 R_0 \\ (I_y - I_x)P_0 Q_0 + Q_0 R_0 I_{xz} \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ L_0 \\ M_0 \\ N_0 \end{pmatrix} \quad (4.13)$$

que serão as principais equações utilizadas para a definição da modelagem dinâmica da aeronave.

As variáveis de controle são definidas segundo a Eq. (4.14)

$$\vec{\delta\xi} = \begin{pmatrix} \delta pr \\ \delta ail \\ \delta lem \end{pmatrix} \quad (4.14)$$

onde δpr é a deflexão do profundor, δail a deflexão do aileron e δlem a deflexão do aileron da aeronave.

As variáveis de estado se relacionam com as variáveis de controle segundo (Fossen, 2011).

4.2 Redes Neurais

4.2.1 Conceitos Gerais

Uma Rede Neural Artificial (RNA) é uma técnica computacional que apresenta um modelo matemático inspirado na estrutura neural de seres vivos e na capacidade de aprendizado pela experiência. É constituída por um número finito de estruturas individualizadas (chamadas de neurônios), organizadas em camadas (Silva, 2006).

O esquema da rede pode ser visualizado na figura 3 apresentada abaixo.

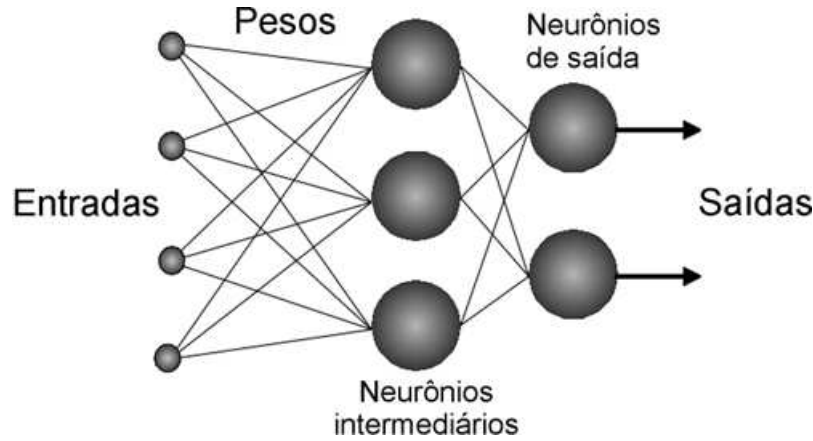


Figura 3: Representação de uma RNA (Tafner, 1998)

O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico (Tafner, 1998). Fazendo uma analogia, os dentritos são análogos às entradas, as sinapses têm representação pelos pesos, os estímulos são captados e processados pela função de soma e o disparo do neurônio biológico é representado pela função de transferência ou função de ativação. As Fig. 4 e 5 mostram representações gráficas do funcionamento de um neurônio.

A saída a_i^l de cada neurônio é dada pela Eq. (4.15). σ é a função de ativação da rede neural e tem saída geralmente com intervalo entre $[-1,1]$, sendo responsável por manter a característica de não linearidade à estrutura matemática. Os parâmetros w_{ij}^l e b_i^l são, respectivamente, o valor do peso sináptico e do bias associados ao neurônio i , esses parâmetros são obtidos durante o treinamento da rede (Pereira, 2017).

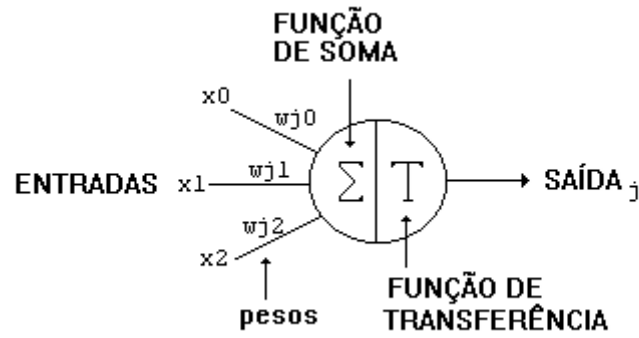


Figura 4: Representação de um neurônio e suas respectivas analogias ao neurônio biológico (Tafner, 1998).

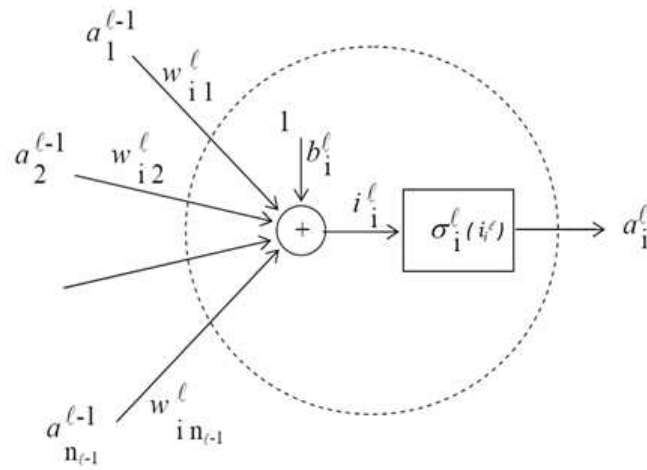


Figura 5: Esquema de um neurônio i da camada l (Silva, 2006).

$$\begin{aligned}
i_i^l &= \sum_{j=1}^{n_{i-1}} w_{ij}^l a_j^{l-1} + b_i^l \\
a_i^l &= \sigma_i^l(i_i^l)
\end{aligned} \tag{4.15}$$

4.2.2 Treinamento da rede neural

Depois de estabelecidos os números de camadas e de neurônios em cada camada, deve-se realizar o treinamento da rede neural e assim determinar valores para os parâmetros w_{ij}^l e b_i^l .

O treinamento da rede neural é executado através de métodos de otimização, sejam eles inspirados nos métodos da máxima descida e de métrica variável, ou até mesmo métodos heurísticos, baseando-se na minimização da função custo Cf apresentada na Eq. (4.16).

$$Cf = \sum_{k=1}^{Nd} [y(k) - \hat{y}(k)]^2 \tag{4.16}$$

Sendo Nd o número de elementos da sequência, $y(k)$ é o conjunto de dados usados para o treinamento da rede e $\hat{y}(k)$ são os valores obtidos na saída da rede neural. Com a minimização da função custo, espera-se que ao final do treinamento da rede, obtenha-se valores de saída $\hat{y}(k) \approx y(k)$, $k = 1, 2, \dots, Nd$.

Esse treinamento pode ser realizado de duas formas: de forma *off-line* (também chamado de treinamento em grupo) e de forma *on-line* (ou tempo real).

O treinamento *off-line* é utilizado para sistemas invariantes no tempo, de onde se consegue extrair todos os dados necessários ou para se obter valores iniciais dos parâmetros da rede. Nesse caso, o valor de Nd deve ser na casa de centenas permitindo uma grande variedade de informações do processo. O número de iterações deve ser definido de modo a otimizar o processo de busca e obter o valor mínimo de Cf . Após realizado o treinamento *off-line*, é necessário testar a rede para outros valores de entrada/saída para garantir que a mesma tenha boa capacidade de generalização. O treinamento *on-line* é utilizado para sistemas variantes no tempo. O número de iterações utilizadas durante o treinamento por cada passo do controle deverá ser no mínimo na casa de centenas (Pereira, 2017), isso garantirá a rápida adaptação da rede ao sistema. Em sistemas não-lineares, o treinamento *on-line* pode causar o esquecimento da rede acerca das características globais, fazendo com que esta funcione apenas para pontos com características locais

4.3 Teoria de conjuntos *Fuzzy*

Para o estudo dos principais conceitos da área da Teoria dos Conjuntos *Fuzzy* é utilizado como referência (Pereira, 2017).

Conjunto *Fuzzy*: na teoria clássica de conjuntos, um elemento pertence ou não a um dado conjunto. Zadeh (Zadeh, 1965) propôs que alguns elementos poderiam ser mais pertinentes a um conjunto do que outros, existindo um fator de pertinência do elemento variando no intervalo de $[0, 1]$, sendo que 0 representa a total exclusão do elemento àquele conjunto e 1 representa a total pertinência do elemento. Um subconjunto F do conjunto universo U é então definido em termos de uma função de pertinência u que relaciona cada elemento x de U a um dado número $u(x)$, o grau de pertinência de x a F .

Sistema baseado em Regras *Fuzzy*: Sistemas baseados em Regras *Fuzzy* (SBRF) contém quatro componentes: um processador de entrada, que realiza a fuzzificação dos dados de entrada,

uma base de regras, composta por um conjunto de regras nebulosas, uma máquina de inferência *fuzzy* e um processador de saída, que faz a defuzzificação dos dados e apresenta um número real como saída (Pereira, 2017). A base de regras define como os conjuntos *fuzzy* se relacionam a partir de um conjunto de regras do tipo “Se ... então...”. Depois de estabelecida essa base de regras, o SBRF pode ser visto como um mapeamento entre entrada e saída da forma $y = f(x)$, $x \in R^n$ e $y \in R^m$. A figura da arquitetura é apresentada na Fig. 6.



Figura 6: Representação da arquitetura de um sistema Baseado em Regras *Fuzzy* (Jafelice, 2003).

Os componentes do SBRF são descritos a seguir:

Processador de Entrada: esse componente realiza a fuzzificação dos dados de entrada, ou seja, a tradução das entradas do sistema em conjuntos *fuzzy* em seus respectivos domínios. São construídas as funções de pertinência para cada entrada. A Fig. 7 apresenta algumas funções de pertinência utilizadas. Foram utilizadas funções de pertinência Gaussianas para o desenvolvimento deste trabalho.

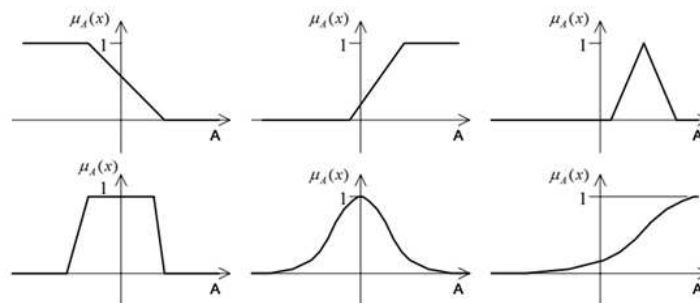


Figura 7: Formas usuais utilizada para função de pertinência (Pacheco, 2015)

Base de Regras: considerado o núcleo do SBRF, composto por uma série de proposições *fuzzy* na forma “Se...então...”.

Máquina de inferência *Fuzzy*: nesse componente do SBRF, cada proposição *fuzzy* criada na base de regras é traduzida matematicamente por meio de métodos de inferência *Fuzzy*. Os métodos de Mamdani e Takagi-Sugeno são apresentados a seguir.

Método de Mamdani: Agrega cada uma das regras da base de regras em operadores lógicos de OU (modelado pelo operador máximo) e E (modelado pelo operador mínimo).

A seguir é apresentado um exemplo que traduz melhor o funcionamento do método (Pereira, 2016):

- Regra 1: Se (x é A_1 e y é B_1) então (z é C_1)
- Regra 2: Se (x é A_2 e y é B_2) então (z é C_2)

A Fig. 8 ilustra a saída z do sistema de inferência Mamdani gerada a partir das entradas x e y e das regras de composição Max-min.

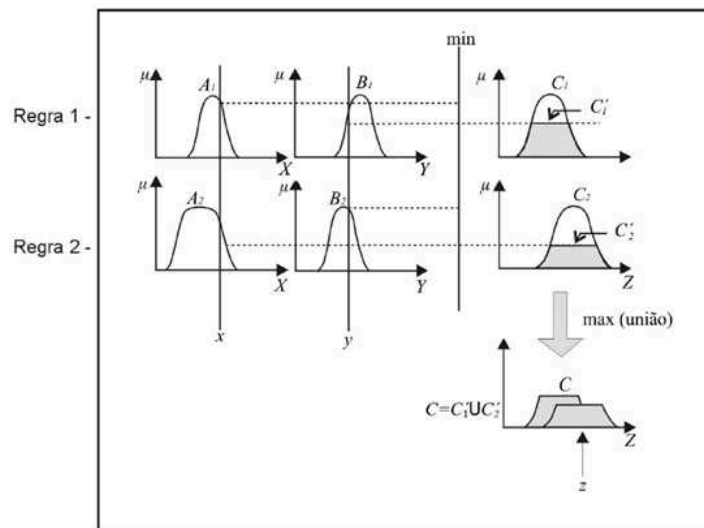


Figura 8: Ilustração do método de Mamdani (Jafelice, 2003).

Método de Takagi-Sugeno: esse método realiza os passos do método de Mamdani, porém, apresenta uma modelagem diferenciada dos consequentes do SBRF, sendo estes, funções do tipo $f_i = f(x_1, x_2, \dots, x_n)$, em que x_1, x_2, \dots, x_n são as entradas do SBRF.

A Fig. 9 ilustra a saída z do sistema de inferência Takagi-Sugeno gerada a partir das entradas x e y .

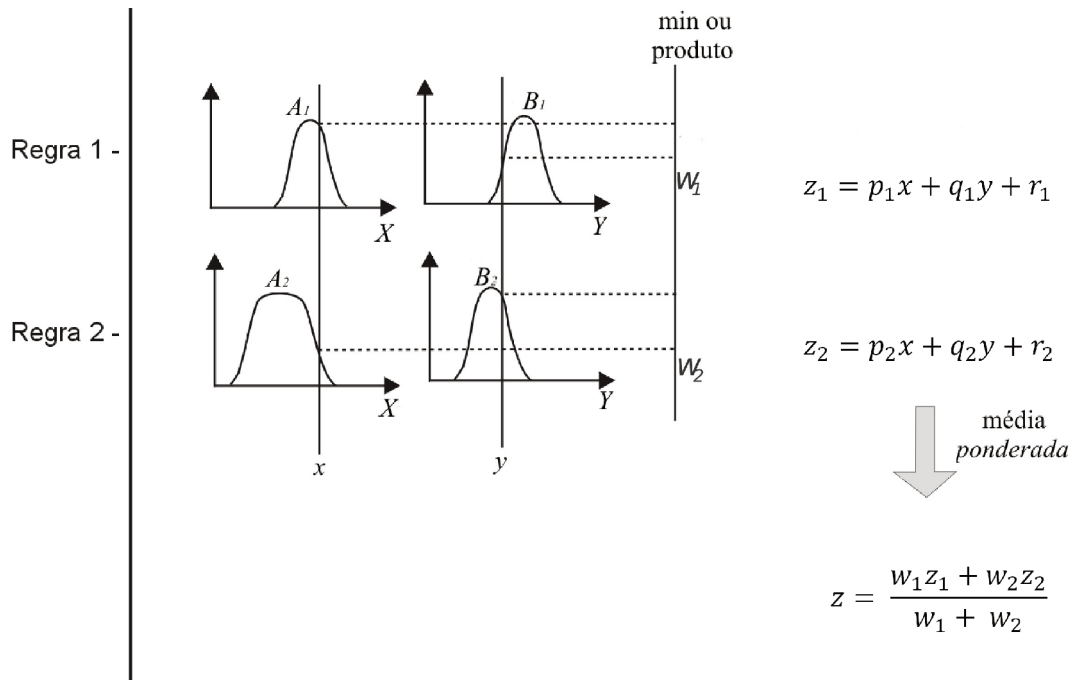


Figura 9: Ilustração do método de Takagi-Sugeno (Jafelice, 2003)

Processador de Saída: o processador de saída realiza a defuzzificação do sistema, isto é, realiza a transformação da saída do sistema *fuzzy* em um número real. Deve-se escolher um método de defuzzificação que seja mais adequado ao problema. São definidos 2 métodos.

Média Ponderada: método de defuzzificação mais utilizado para o método de inferência de Takagi-Sugeno. A saída z é calculada a partir da Eq. (4.17) a seguir. Para o desenvolvimento do trabalho, utilizou-se a média ponderada.

$$z = \frac{\sum_{i=1}^k C_i(u) f_i}{\sum_{i=1}^k C_i(u)} \quad (4.17)$$

Centro de gravidade: forma de defuzzificação mais utilizada para o método de inferência Mamdani. É semelhante à média ponderada, porém os pesos são os valores de $C(u)$ que indicam o grau de compatibilidade do valor u com o conceito modelado pelo conjunto *fuzzy* C , onde R é a região de integração.

$$\frac{\int_R uC(u)du}{\int_R C(u)du} \quad (4.18)$$

4.4 Sistema Neuro-Fuzzy

O sistema Neuro-fuzzy adaptativo (*Adaptative neuro- inference system* - ANFIS) é um sistema constituído a partir da hibridização das técnicas de modelagem de Redes Neurais Artificiais (*Artificial Neural Network* - ANN) e Sistema de Inferência Fuzzy (*Fuzzy Inference System* - FIS). O ANFIS apresenta a estrutura do FIS (com o processo de fuzzificação, a base de regras, a máquina de inferência *fuzzy* e a defuzzificação) porém, com a mesma disposição da estrutura da ANN e sua

capacidade de aprendizado. A partir da Fig. 10 é possível observar a arquitetura básica de um ANFIS e suas camadas.

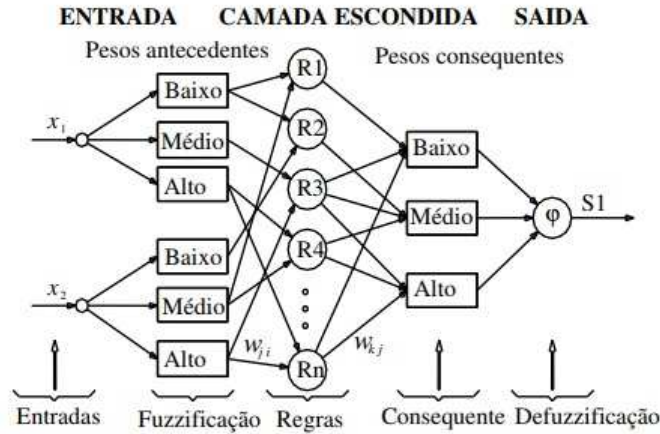


Figura 10: Representação da arquitetura básica de um ANFIS (Fonte: www.maxwell.vrac.puc-rio.br)

Na Fig. 11, é mostrada a composição e a definição de cada camada do ANFIS utilizando o método de inferência *fuzzy* de Takagi-Sugeno (Takagi e Sugeno, 1985; Chen, Lin e Lin, 2008; Pereira, 2016):

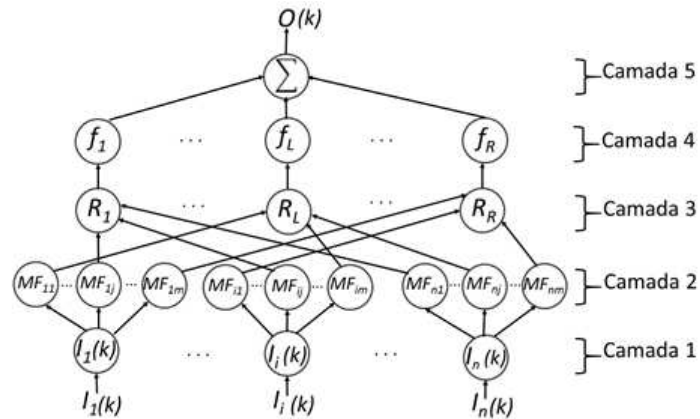


Figura 11: Representação de um sistema generalizado ANFIS (Pereira, 2016).

- **Camada 1:** A camada 1 representa as entradas do sistema, cada nó recebe uma entrada $I_i(k)$, de forma que a saída seja definida pela Eq. 4.19.

$$u_i^{(1)} = I_i(k) \quad (4.19)$$

- **Camada 2:** Na camada 2 ocorre a fuzzificação dos dados de entrada do sistema. São construídas as funções de pertinência (MF) para cada entrada. A saída é definida pela Eq. 4.20.

$$u_i^{(2)} = f_i(k) \quad (4.20)$$

Onde $f(k)$ é a função de pertinência escolhida para o problema (ex: função Gaussiana, função triangular, ...).

- **Camada 3:** Na camada 3 ocorre a criação da base de regras do ANFIS com uma coleção de proposições do tipo Se... Então..., para toda regra R_L , os operadores lógicos E e OU são modelados matematicamente como produto e máximo respectivamente. Cada saída $u_i^{(3)}$ da camada 3 depende da sua respectiva base de regras.

- **Camada 4:** Na camada 4 ocorre a definição do conseqüente de cada nó, que é definido como sendo uma função $f_L : R^n \rightarrow R$ tal que $f_L = f(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k)$ tal que $w_{1L}, \dots, w_{jL}, \dots, w_{oL}$ são pesos determinados durante o treinamento do ANFIS (Pereira, 2016). A saída da camada 4 é dada pela Eq. (4.21):

$$u_i^{(4)} = u_i^{(3)} f(I_1, \dots, I_i, \dots, I_n, w_{1L}, \dots, w_{jL}, \dots, w_{oL}, k) \quad (4.21)$$

- **Camada 5:** Na camada 5 ocorre a defuzzificação e a conseqüente determinação da saída do ANFIS, utilizando a média ponderada.

$$O(k) = \frac{\sum_{L=1}^R u_i^{(4)(k)}}{\sum_{L=1}^R u_i^{(3)(k)}} \quad (4.22)$$

4.5 Evolução diferencial

A evolução diferencial é um método robusto de minimização de funções (que podem ser não lineares e não diferenciáveis) e de fácil implementação, capaz de chegar ao valor ótimo global de forma mais rápida e fácil que os métodos heurísticos. Requer poucas variáveis de controle e apresenta boa adaptabilidade à computação paralela (Storn e Price, 1996). O método da evolução diferencial busca encontrar o valor de mínimo global de uma função $f_{obj}(X)$, onde X é um vetor de elementos que representa cada uma das variáveis independentes da função. As etapas e operadores que constituem o algoritmo estão descritos a seguir (Han, Lin e Chang, 2013) e representados na Fig. 12.



Figura 12: Representação da evolução diferencial, adaptada de Mór et al. (2015).

Na etapa de inicialização do algoritmo são definidos os parâmetros de execução do método e os limites de busca de cada variável. São inicializados também os vetores (ou indivíduos) $X_{i,0}$ seguindo uma forma de distribuição uniforme dentro do intervalo $[0, 1]$, onde cada um dos indivíduos representa uma possível solução para o problema de otimização e cujos elementos, são os valores normalizados das variáveis do processo. Em seguida é calculada a função custo Cf utilizada como critério de minimização para obtenção dos parâmetros dos controladores do processo.

A etapa de mutação consiste na alteração dos vetores $X_{i,k}$ a partir da Eq. (4.23).

$$V_{i,k} = X_{i,k} + F(X_{j,k} - X_{l,k}) \quad (4.23)$$

em que $V_{i,k}$ é o vetor obtido a partir do vetor $X_{i,k}$ na iteração k , F é o fator de perturbação do vetor $X_{i,k}$ (geralmente contido no intervalo de $[0,2]$), $X_{j,k}$ e $X_{l,k}$ são outros vetores, escolhidos aleatoriamente e pertencentes à amostra (conjunto S – conjunto das soluções possíveis para o problema de otimização) na iteração k .

Na fase de cruzamento (ou recombinação), o objetivo é mesclar os vetores $X_{i,k}$ e $V_{i,k}$ garantindo maior diversidade na população. A tomada de decisão sobre a operação realizada nessa fase é mostrada na Eq. (4.24).

$$C_{e,i,k} = \begin{cases} V_{e,i,k} & \text{se } rand(0,1) \leq CR \text{ OU } e = randnum(1, n_e) \\ X_{e,i,k} & \text{se } rand(0,1) > CR \text{ OU } e \neq randnum(1, n_e) \end{cases} \quad (4.24)$$

Para cada elemento e do vetor $V_{i,k}$, utiliza-se um gerador de valores uniformes distribuídos no intervalo $[0,1]$, $rand(0,1)$ e compara-se com a taxa de Cruzamento CR também definida no intervalo $[0,1]$. O vetor $C_{i,k}$ é o vetor candidato a participar do conjunto S na iteração $k+1$. Para garantir que pelo menos um elemento do vetor $C_{i,k}$ candidato a participar seja pertencente ao vetor $V_{i,k}$ obtido na fase de mutação, adiciona-se a segunda condição na tomada de decisão da Eq. 4.24. $randnum(1, n_e)$ é um gerador de números inteiros uniformemente distribuídos no intervalo de $[1, n_e]$, sendo n_e o número de elementos dos vetores. Esse procedimento é repetido para todos os vetores do conjunto S .

Na última fase, a fase da seleção é feita uma análise dos vetores $C_{i,k}$ gerados na fase do cruzamento. O procedimento para a seleção é descrito na Eq. (4.25) e que deve ser realizado para todos os vetores do conjunto S .

$$X_{i,k+1} = \begin{cases} X_{e,i,k} & \text{se } Cf(X_{i,k}) < Cf(C_{i,k}) \\ C_{i,k} & \text{se outro caso} \end{cases} \quad (4.25)$$

Após a seleção, avança-se para a iteração $K + 1$, e todo o procedimento é repetido a partir da fase de mutação até a iteração k_e , quando o critério de parada pré-determinado for atendido.

4.6 Recozimento simulado

O método do Recozimento Simulado é originado da analogia do recozimento de metais, que é o processo de alteração das propriedades de um material metálico por aquecimento e resfriamento lento do mesmo.

O ponto de importância para um recozimento é saber o que acontece ao sistema em temperaturas baixas, e isso está diretamente relacionado com a energia interna do material. Experimentos que garantem baixos níveis de energia interna do material, necessitam de um recozimento cuidadoso onde existe uma redução lenta da temperatura desde o estado de fusão do material, resultando numa estrutura cristalina mais pura. O processo de encontrar um estado de energia baixo é simulado para se encontrar a solução otimizada do problema. O método do Recozimento Simulado faz uma busca local de implementação simples proposto por (METROPOLIS et al.,1953).

São feitas analogias matemáticas que auxiliarão na solução do problema. O estado termodinâmico do sistema corresponde à solução atual do problema, a equação de energia do sistema termodinâmico (a qual se deseja minimizar) corresponde à função custo ou função objetivo e o estado de referência será correspondente ao mínimo global da função.

Para se implementar a programação, deve-se escolher a temperatura inicial, o número de iterações do algoritmo com a mesma temperatura e a estratégia de redução de temperatura ao longo da execução do algoritmo. Tem-se que, a partir de um ponto no espaço da estrutura, calcula-se a energia do mesmo e a energia de um ponto próximo nesse mesmo espaço, denominado de ponto vizinho. Se a energia do ponto vizinho for menor, então esse ponto passa a ser o novo ponto de interesse e assim o algoritmo prossegue, calculando os próximos pontos vizinhos. Se a energia do ponto vizinho for maior que a energia do ponto de interesse, o ponto vizinho não é descartado automaticamente devido ao fato de um ponto de maior energia poder eventualmente sair de mínimos locais e conduzir a solução para o mínimo global.

A figura 13 abaixo mostra um exemplo da situação descrita. A simulação encontra os pontos x_1 e x_2 que se apresentam como mínimos locais, mas que, analisando a função, não são mínimos globais. Para se encontrar o mínimo global da função da Fig. 13, é necessário passar pelo ponto x_4 , que se apresenta como máximo local até chegar ao ponto próximo x_5 que é o mínimo global da função. Se não existisse a probabilidade de aceitação do ponto vizinho de maior energia que o ponto atual de interesse, o algoritmo ficaria preso no mínimo global e não seria obtida uma função custo ótima.

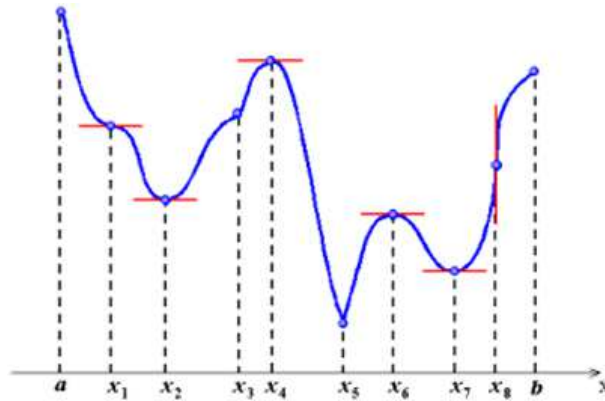


Figura 13: Máximos e mínimos de uma função qualquer (Fonte: www.matematica.pt)

Na inicialização do algoritmo, como a temperatura está alta, a probabilidade de se aceitar um ponto de maior energia é alta, varrendo o espaço de soluções de forma eficaz. À medida que a temperatura diminui, também diminuem as chances de transições para pontos de maior energia e o algoritmo executa uma varredura mais local.

A figura 14 representa um fluxograma com o algoritmo básico do método de otimização do Recozimento Simulado.

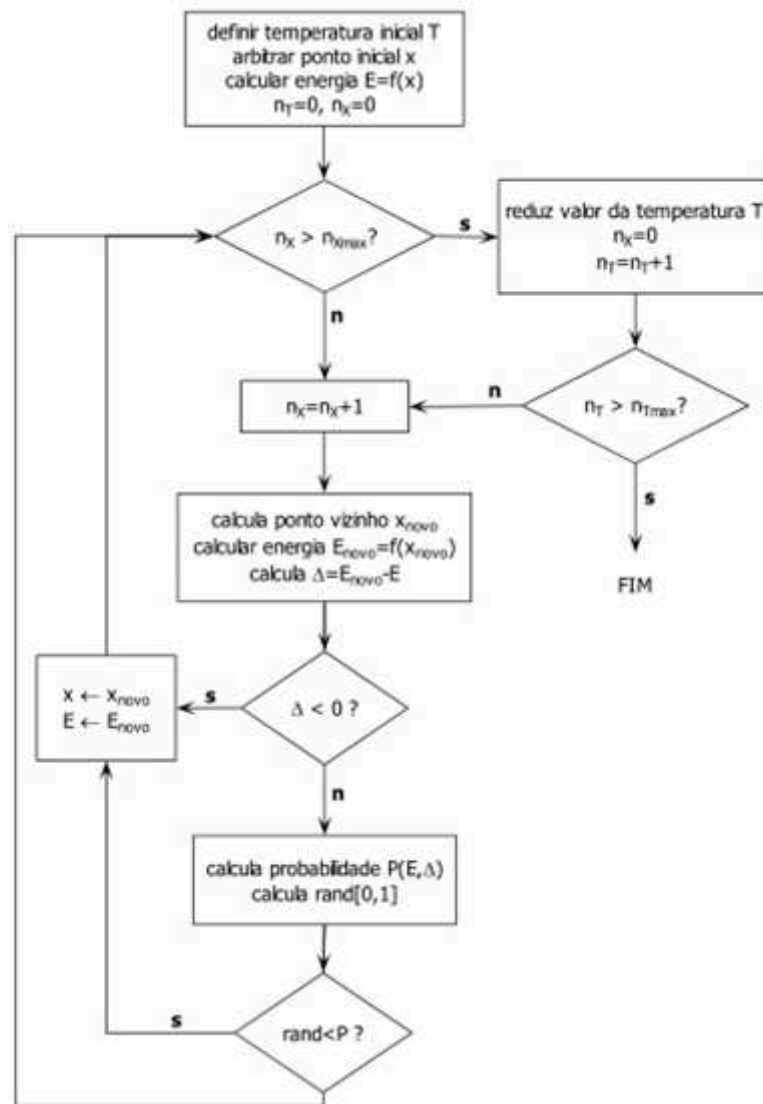


Figura 14: Algoritmo básico do Recozimento Simulado (Fonte: www2.peq.coppe.ufrj.br)

A redução da temperatura é usualmente feita linearmente a partir da Eq. (4.26):

$$T^{k+1} = \alpha T^k \quad (4.26)$$

onde α varia entre 0,7 e 0,95. Também podem ser utilizadas funções exponenciais de resfriamento.

O número de ciclos de resfriamento, ou número de temperaturas (n_{Tmax}) e o número de iterações para cada temperatura (n_{xmax}) devem ser suficientes em grandeza para atender o problema. Em geral, n_{Tmax} consiste no intervalo [10, 20] e n_{xmax} em [20, 100], podendo variar de acordo com o problema proposto.

Para se avaliar a energia da solução vizinha, é proposto que se faça a diferença entre a energia da solução atual e a energia da solução vizinha, de forma que:

$$\Delta = E^{k+1} - E^k \quad (4.27)$$

Se Δ for negativo, tem-se que a solução atual é mais energética que a solução vizinha proposta, então a solução vizinha passa a ser a solução atual de interesse. Se Δ for positivo, a solução vizinha é mais energética que a solução atual de interesse e a probabilidade de a mesma substituir a solução atual é definida pela Eq. (4.28).

$$p_{RM} = e^{\frac{-\Delta}{T}} \quad (4.28)$$

A partir do cálculo da probabilidade de aceitação da nova solução mais energética que a atual, é escolhido um número aleatório (randômico) r a partir de uma distribuição uniforme no intervalo $[0, 1]$. Se $r < p_{RM}$ então a transição é aceita, se não, o ponto é descartado e o algoritmo busca um novo ponto.

O algoritmo será executado por $n_{x_{max}} \times n_{T_{max}}$ vezes, apresentando o melhor resultado de otimização.

5 Metodologia

Em função da característica dinâmica da aeronave apresentada na seção 4.1, é possível desacoplar as equações do movimento em dois modelos distintos: longitudinal e látero-direcional. Sendo assim, são desenvolvidos dois sistemas de controle para o sistema aeronáutico: controle longitudinal (que inclui o controle da altitude e do ângulo de arfagem a partir da deflexão do profundor do Cessna 172) e controle látero-direcional (que inclui o controle do ângulo de guinada e do ângulo de rolagem a partir da deflexão do aileron e do leme).

5.1 Controle Longitudinal

Neste caso, o controle longitudinal é responsável por controlar a altitude (h) e o ângulo de arfagem (θ) da aeronave a partir da deflexão do profundor do Cessna 172. O controlador possui uma estrutura em cascata, separado em dois sistemas baseado em regras *fuzzy* para um melhor desempenho. O mesmo pode ser visualizado na Fig. 15.

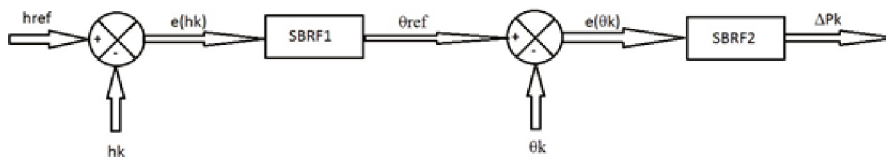


Figura 15: Sistema de controle Longitudinal

Primeiramente, são encontrados os parâmetros da malha interna (SBRF2) conforme Fig. 16.

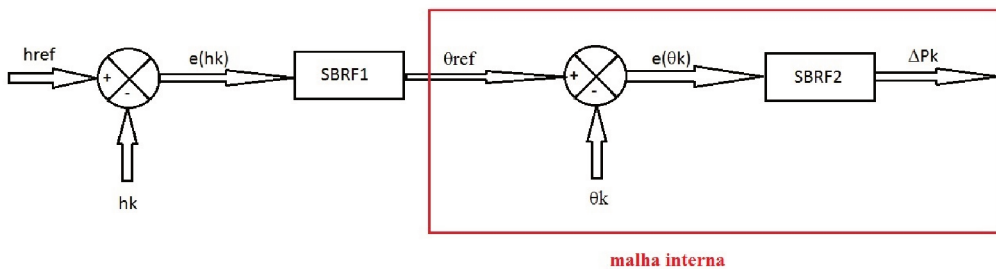


Figura 16: Sistema de controle Longitudinal malha interna

O ângulo de arfagem de referência da aeronave ou *setpoint* (θ_{ref}), é definido como parâmetro de entrada da malha interna que, combinado com o ângulo de rolagem no instante K coletado durante a simulação θ_k , gera o sinal de erro. O SBRF2 possui como entrada o erro gerado ($e(\theta_k)$) e a partir dele, são encontrados os parâmetros para o controlador da malha interna, tendo como saída a deflexão do profundor (ΔP_k).

Após ser encontrada a configuração do controlador para a malha interna, os parâmetros são fixados, iniciando a análise do controle longitudinal completo apresentado na Fig. 15. A entrada da malha externa é a altitude de referência da aeronave (h_{ref}). O erro de altitude, $e(hk)$, é encontrado a partir da diferença entre a altitude de referência e a altitude coletada no instante k (hk), e é definido como sendo a entrada do SBRF1. Os parâmetros do controlador da malha externa são encontrados e é realizada a avaliação do desempenho do sistema.

5.2 Controle Látero-direcional

O controle látero-direcional realiza a adequação do ângulo de guinada (Ψ) e de rolagem (Φ) da aeronave a partir da deflexão de dois atuadores da sua estrutura: o aileron e o leme. O controlador possui uma estrutura em cascata com três sistemas baseados em regras *fuzzy*, onde a malha interna realiza o controle da variável de rolagem e a malha externa realiza o controle completo do ângulo de guinada. O sistema pode ser visualizado conforme Fig. 17.

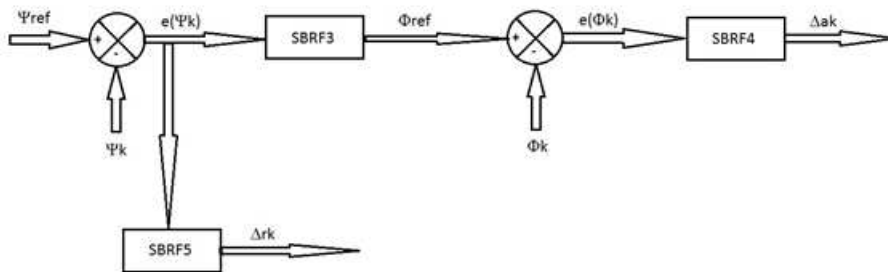


Figura 17: Sistema de controle Látero-direcional

Analogamente ao sistema de controle longitudinal, são calculados os parâmetros da malha interna para assim serem obtidos os parâmetros da malha externa. A malha interna pode ser visualizada conforme Fig. 18.

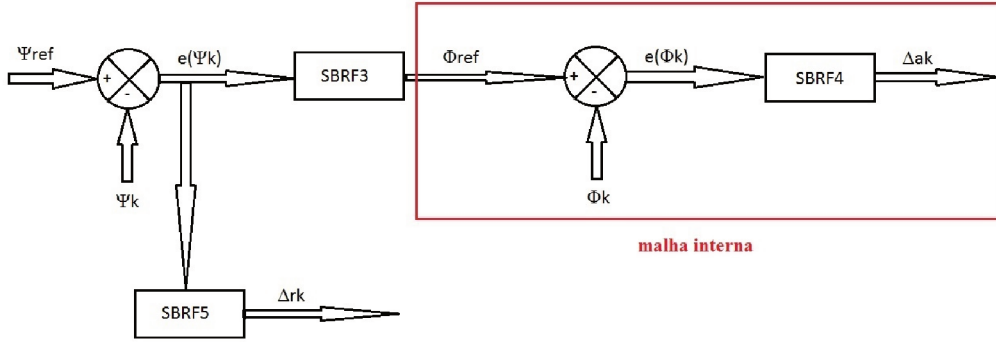


Figura 18: Sistema de controle Látero-direcional malha interna

A malha interna recebe como parâmetro de entrada o ângulo de rolagem de referência (Φ_{ref}) e é gerado um erro desse ângulo ($e(\Phi_k)$) quando combinado ao Φ da aeronave no instante k (Φ_k). O erro é aplicado ao SBRF4, os parâmetros do controlador são encontrados e é gerada como saída a deflexão do aileron (Δak).

São fixados os parâmetros do SBRF4 e malha externa recebe como entrada o ângulo de guinada de referência (Ψ_{ref}). O erro de guinada $e(\Psi_k)$ é definido como entrada de SBRF3 e SBRF5. O SBRF5 produz como saída a deflexão do leme (Δrk). Encontrados os parâmetros dos dois controladores externos, é feita a análise final do sistema apresentado na Fig. 17.

5.3 Ajuste de parâmetros de controlador

Os parâmetros dos controladores são encontrados utilizando os métodos de otimização: Evolução Diferencial e Recozimento Simulado. Como observado nos itens 4.5 e 4.6, para a aplicação desses métodos são necessárias funções que servirão como critério de otimização a fim de obter-se um resultado ótimo. A função utilizada em ambos os casos é descrita na Eq. 5.1, definida como erro RMS (*Root Mean Square*) - E_{RMS} .

$$E_{RMS} = \frac{\sum \sqrt{(x_{ref,k} - x_k)^2}}{n} \quad (5.1)$$

Onde $x_{ref,k}$ é o valor de referência da variável x na iteração k , x_k é o valor da variável na iteração k e n é o número de dados disponíveis para treinamento.

Para a obtenção do controlador ótimo nas duas situações, são necessários alguns ajustes nos parâmetros utilizados em cada método.

Na Evolução Diferencial, os resultados são dependentes do número máximo de gerações e do número de indivíduos gerados, que devem ser suficientes em grandeza para ser possível encontrar o mínimo global da função de interesse. Além disso, a probabilidade de cruzamento (definida no intervalo de $[0, 1]$) e o fator de perturbação (definido no intervalo de $[0, 2]$) devem ser ajustados de acordo com cada sistema e problema a ser tratado.

Analogamente à Evolução Diferencial, no Recozimento Simulado os resultados são dependentes das escolhas dos parâmetros, que são o número de ciclos de resfriamento, o número de iterações para

cada temperatura, a temperatura inicial, o desvio padrão da distribuição normal e o α (utilizado para definição do decaimento da temperatura) também têm grande influência na execução do método

5.4 Metodologia de Análise

Ao final da execução da simulação no ambiente criado no *software* MatLab® , são mostrados os resultados através de gráficos de entrada, saída do controlador (deflexão do atuador da aeronave) e evolução do erro RMS gerado. A fim de ilustrar o desempenho do sistema, será utilizado como exemplo a malha interna do Controle Longitudinal.

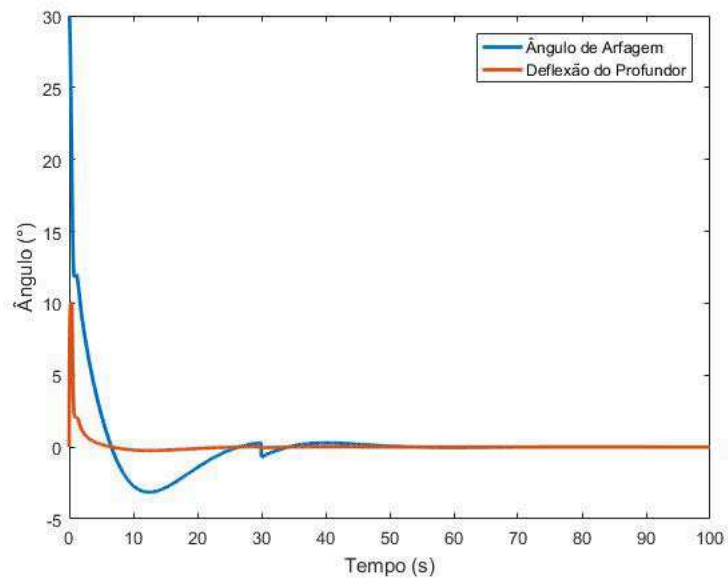


Figura 19: Entrada (Ângulo de Arfagem) e saída (Deflexão do Profundor) do sistema de controle

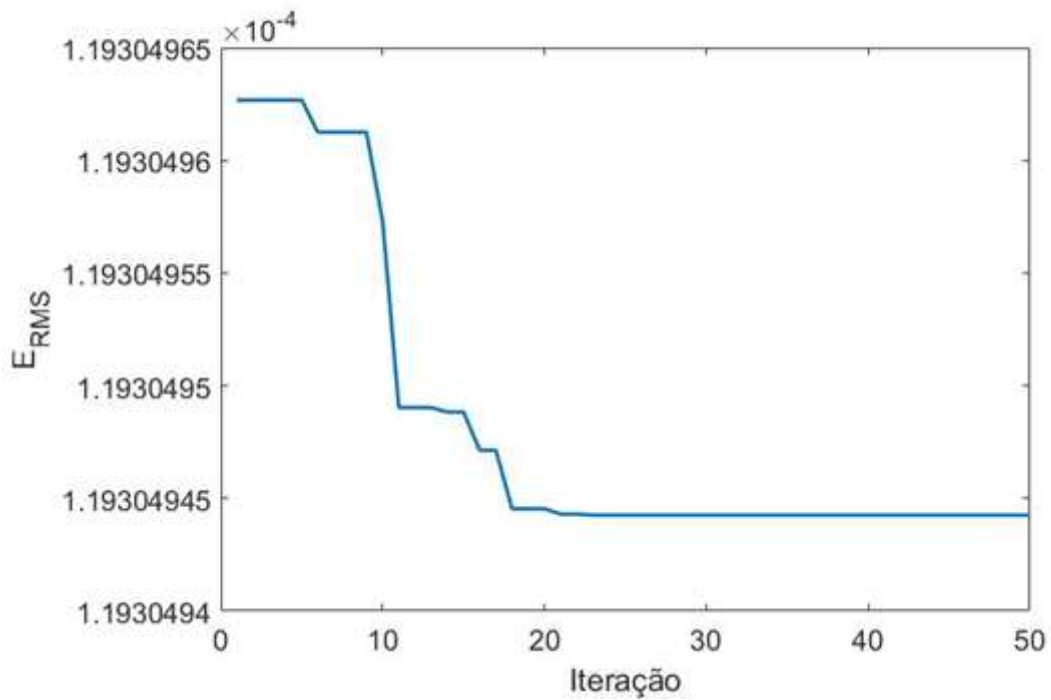


Figura 20: Erro RMS

O diagnóstico de desempenho do controlador é feito a partir do gráfico mostrado na Fig. 19. Uma vez que o ângulo de arfagem possui perturbações durante o tempo de simulação, o profundor deve ser capaz de eliminar as mesmas e garantir uma estabilidade à entrada do sistema, mantendo-a próxima ao valor desejado. Paralelamente, é necessário que o Erro RMS apresentado na Fig. 20 seja mínimo ao final das iterações. Ainda, o tempo de execução dos dois métodos deve ser coletado e analisado para comparação.

6 Desenvolvimento

A simulação desenvolvida no *software* MatLab® consiste de um programa principal *main.m* que interage com programas secundários criados para o cálculo e a aplicação do controlador. A seguir será descrito cada programa de forma a contribuir com o entendimento do projeto

6.1 model.m

Possui o modelo implementado da planta do Cessna 172 em escala real, contendo inicialmente a definição de parâmetros de trimagem, inércia, aerodinâmicos e propulsivos, longitudinais e laterais. A obtenção das matrizes dinâmicas se dá segundo as Eq. (4.12) e (4.14) descrita no item 4.1.1.

6.2 convc2d.m

Realiza a discretização da planta definida no programa *model.m*, levando em consideração a utilização de um segurador de ordem 0, conforme fragmento de código 1.

```
function [G,H]= convc2d(A,B,dt)

G=expm(A*dt) ;
nm=length(A) ;
H=(A\ (expm(A*dt) -eye(nm) ) ) *B;

end
```

Fragmento de código 1: Programa convc2d.m

onde A é a matriz dinâmica do sistema, B é a matriz de entradas do sistema, que representam o modelo de espaço de estados, G e H correspondem, respectivamente, às matrizes A e B discretizadas e dt é o tempo de amostragem do sistema de controle.

6.3 Ev_dif.m

Programa que faz o cálculo dos parâmetros do controlador a partir da minimização da função de interesse utilizando o método da Evolução Diferencial.

São inicializados os parâmetros e vetores a serem utilizados em todo o método, como número máximo de gerações (*nmaxgeracoes*), número de vetores (*nind*), número de variáveis de cada vetor (*nvar*), probabilidade de cruzamento (*pcruz*), fator de perturbação (F) e demais vetores e matrizes para armazenamento durante o processo (X, V, C, fX, fC, var).

Para garantir que o algoritmo encontre os parâmetros do controlador de forma ótima e evitar que exista uma perda em processamento deve-se estabelecer um limite de busca, definidos conforme fragmento de código 2.

```

function var=Ev_Dif

nmaxgeracoes=30;
nind=30;
nvar=1;

pcruz=0.95;%Probabilidade de cruzamento
F=0.4;

X=zeros(nind,nvar); %indivíduos
V=X;
C=X;
fX=zeros(nind,1); %valor da fobj para cada individuo
fC=fX;
var=zeros(nvar,1);

%Definição dos limites da busca das variáveis

for i=1:nvar
    linf(i)=0;
    lsup(i)=5;
end

```

Fragmento de código 2: programa *Ev_dif.m* – definição de parâmetros, vetores e limites de busca

A população inicial é gerada e armazenada na matriz X a partir da função aleatória do MatLab® *rand* que gera números compreendidos no intervalo $[0,1]$. A etapa de Avaliação transforma o domínio de $[0,1]$ para o domínio definido entre os limites pré-estipulados para cada uma das variáveis otimizadas no processo, ao final é calculada a aptidão para cada um dos vetores originais utilizando a função objetivo. Esse processo pode ser visualizado no fragmento de código 3.

```

%POPULACAO INICIAL
for i=1:nind
    for j=1:nvar
        X(i,j)=rand;
    end
end

%AVALIAÇÃO DOS individuos
for i=1:nind
    for j=1:nvar
        var(j)=(lsup(j)-linf(j))*X(i,j)+linf(j);
    end
    fX(i)= calcadapt(var);
end

```

Fragmento de código 3: programa *Ev_dif.m* – população inicial e avaliação dos indivíduos

É realizada a etapa da mutação descrita no item 4.5. As variáveis $mut1$ e $mut2$ são utilizadas para a escolha aleatória dos vetores $X_{j,k}$ e $X_{l,k}$ descritos na Eq. (4.23). Para garantir que todos os elementos dos vetores estejam no intervalo entre $[0, 1]$ é criada uma condição descrita no fragmento de código 4.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPERADOR MUTAÇÃO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mut1=randperm(nind);
mut2=randperm(nind);
for i=1:nind
    for j=1:nvar
        V(i,j)=X(i,j)+F*(X(mut1(i),j)-X(mut2(i),j));
        if V(i,j)<0
            V(i,j)=0;
        end
        if V(i,j)>1
            V(i,j)=1;
        end
    end
end
end

```

Fragmento de código 4: programa *Ev_dif.m* – mutação

As etapas de Cruzamento e Seleção são executadas conforme Eq. (4.24) e (4.25) descritas no item 4.4. O algoritmo possui três critérios de parada, sendo eles erro relativo, erro absoluto e número máximo de iterações.

Os parâmetros iniciais utilizados na Evolução Diferencial são apresentados na tabela 1.

Tabela 1: Parâmetros utilizados na Evolução Diferencial

Tipo de Controle		Controle Longitudinal		Controle Látero-direcional	
	Malha	Malha Interna	Malha Externa	Malha Interna	Malha Externa
Parâmetros	Número máximo de iterações	30	50	30	20
	Número de vetores para cada geração	30	50	30	$1,173 \cdot 10^{-3}$
	Número de vetores de cada vetor	1	1	1	2
	Probabilidade de Cruzamento	0,95	0,95	0,95	0,95
	Fator de Perturbação	0,4	0,4	0,4	0,4
	Limite Inferior de Busca	0	0	0	0
	Limite Superior de Busca	5	30	2	2

6.4 Rec_sim.m

Realiza o cálculo da minimização da função de interesse a partir do método do Recozimento Simulado.

São inicializados os parâmetros e vetores necessários para a aplicação computacional como temperatura inicial (T), número de ciclos de resfriamento (nT_max), número de iterações para cada temperatura (nX_max), decaimento da temperatura (α) e outros. É estabelecido um limite de busca de variáveis que garante ao método o melhor desempenho e menor perda durante o processo, análogo ao processo da Evolução Diferencial. O fragmento de código 5 é referente a esta etapa.

```
function var=Rec_Sim

T = 10; %Definição da temperatura inicial
nT_max = 100; %Numero de ciclos de resfriamento / ~nmaxgeracoes
nX_max = 5; %Numero de iterações para cada T / ~nind
sigma = 0.1; % Desvio padrão da distribuição normal
m = 1; %Numero de elementos do vetor X / ~nvar
X_novo = zeros(m,1); %Vetor vizinho candidato a ser o novo X
E_novo = zeros(m,1); % E = f(X) // Função Custo do vetor vizinho
delta = 0; %Diferença entre Enovo e E
Prob = 0; %função probabilidade
w = 0; %numero aleatorio
alfa = 0.9; % Parametro para modificar a temperatura (decaimento de T)
E_v = []; %vetor de melhores E

%Definição dos limites da busca das variáveis
for i=1:m
    linf(i)=0;
    lsup(i)=5;
end
```

Fragmento de código 5: programa *Rec_sim.m* – definição de parâmetros e vetores e limites de busca

O vetor inicial X é arbitrado utilizando a função *rand* do software MatLab® que gera valores seguindo uma distribuição uniforme no intervalo de $[0,1]$. A avaliação de X é feita de forma semelhante à apresentada no item 6.3. A etapa da inicialização pode ser visualizada conforme fragmento de código 6.

```

%Inicialização
X = rand([m 1]); % Arbitrar X inicial
nT = 0; % numero do ciclo de resfriamento
nX = 0; % numero da iteração dentro de T
% var = X;
% E = calcadapt(var); % E = f(X) // Função Custo

%AVALIAÇÃO DOS indivíduos
for j=1:m
    var(j)=(lsup(j)-linf(j))*X(j)+linf(j);
end
E = calcadapt(var); % E = f(X) // Função Custo

```

Fragmento de código 6: programa *Rec_sim.m* - inicialização e avaliação dos indivíduos

O algoritmo começa sua execução segundo o fluxograma da Fig.14 do item 4.6 e possui como critério de parada a temperatura, que deve ser menor que a temperatura máxima estabelecida durante a inicialização das variáveis.

Os parâmetros utilizados no Recozimento Simulado podem ser visualizados na tabela 2.

Tabela 2: Parâmetros utilizados no Recozimento Simulado

Tipo de Controle		Controle Longitudinal		Controle Látero-direcional	
	Malha	Malha Interna	Malha Externa	Malha Interna	Malha Externa
Parâmetros	Temperatura Inicial	10	10	10	10
	Ciclos de Resfriamento	100	50	100	60
	Iterações para cada temperatura	5	5	5	5
	Desvio Padrão da Distribuição Normal	0,1	0,06	0,1	0,1
	Número de elementos do Vetor	1	1	1	1
	Alfa (para decaimento da temperatura)	0,9	0,9	0,9	0,9

6.5 Disturb.m

Cria as perturbações do sistema, que são as entradas coletadas no instante K , conforme fragmento de código 7.

```
function [perturbacao]= disturb(t)

nt=length(t);
perturbacao=zeros(1,nt);

%%%%%%%%%% impulso unitário %%%%%%%%%%
perturbacao(1)=30*pi/180;

perturbacao(floor(0.3*nt))=-1*pi/180;

end
```

Fragmento de código 7: programa Disturb.m

onde t é o tempo de simulação e $perturbacao$ é o vetor de perturbações criado.

6.6 Setpoint.m

Cria o vetor de entradas de referência do sistema, conforme fragmento de código 8.

```
function [set]= setpoint(t)

nt=length(t);
set=zeros(1,nt);

end
```

Fragmento de código 8: programa Setpoint.m

onde t é o tempo de simulação e set é o vetor de entradas de referência criado.

6.7 ControlNF.m

Representa o Sistema Neuro-*fuzzy* que faz o cálculo do controlador de acordo com suas entradas e Base de Regras. A função recebe como parâmetro de entrada a variável *prop*, que é o vetor de parâmetros do controlador gerados pelo método de minimização de funções escolhido, e o erro da variável que deve ser controlada. No exemplo, trata-se da malha interna do controle longitudinal, que recebe como entrada o erro de θ . O erro de θ passa pelo processo de normalização de modo que a ordem de grandeza de cada uma das variáveis não alterem o processo de otimização. Esse processo pode ser visualizado conforma fragmento de código 9.

```

function output=control_NF(erro_theta,delta_prof_k,prop)

%normalização das entradas do sistema
erro_theta = erro_theta/(60*pi/180);

%prop é a matriz linha que vem do método de otimização
%matriz de alfas utilizado na camada 4
prop=prop';
prop=[prop(1);prop(2);prop(3)];

% O sistema possui 2 entradas com 3 subconjuntos cada uma
input=1; %número de entradas
sub_NF=3; %numero de subconjuntos fuzzy
u1=zeros(input,1); % entradas na camada 1 (1 entradas)
u2=zeros(input,sub_NF); % entradas na camada 2 (1 entradas, 3
conjuntos fuzzy cada)
m=[-1 0 1]; %-1 0 1];% média associada a funcao de pert gaussiana
s=ones(input,sub_NF)*0.4; %desvio padrao das normais associadas a
função de pert gaussiana
s2=s.^2;

```

Fragmento de código 9: programa control_NF.m – definição de variáveis e normalização

onde $erro_theta$ é o $e(\theta k)$, $delta_prof_k$ corresponde a ΔPk , $prop$ são os parâmetros encontrados no método de otimização e $output$ é o sinal de saída do controlador.

São definidas variáveis nomeadas como “ ux ” onde “ x ” é o índice da camada. Na camada 1 ocorre a definição das variáveis de entrada do Sistema Neuro-fuzzy. É realizada a tradução de $u1$ para o conjunto *fuzzy* em seus respectivos domínios conforme apresentado no fragmento de código 10. São utilizadas nesse projeto funções de pertinência Gaussiana.

```

%De acordo com o esquema do projeto:
rules = 3; %Número de regras Fuzzy (Critérios "E" "OU") (camada 3)
u3=zeros(rules,1); % Regras Fuzzy (camada 3)
u4=zeros(rules,1); % Função (camada 4)
output = 0; %Saída do sistema

%%%CAMADA 1: entrada de dados
    u1(1)=erro_theta;
    %u1(2)=delta_prof_k;

%%%CAMADA 2: obtenção dos graus de pertinência
for i=1:input
    for j=1:sub_NF
        u2(i,j)=exp(-(u1(i)-m(i,j))^2/s2(i,j));
    end
end

```

Fragmento de código 10: programa control_NF.m – entrada de dados e fuzzificação

Na camada 3 ocorre a definição da base de regras utilizada no controlador. Na camada 4 ocorre a modelagem dos conseqüentes de saída segundo o método de Takagi-Sugeno apresentado no item 4.3. A defuzzificação do sistema ocorre com a aplicação da média ponderada, definida na Eq. (4.17).

6.8 RespModelo.m

A partir do estado do sistema no instante k e do controle aplicado, calcula e fornece o estado do sistema no instante $k + 1$, conforme fragmento de código 11.

```
function [new_output,new_state]=RespModelo(old_state,u,C,D,G,H)

u=[ 0;u;0];
new_state=G*old_state+H*u;
new_output=C*new_state+D*u;

end
```

Fragmento de código 11: programa RespModelo.m

onde old_state corresponde ao estado do sistema na iteração k , u representa a resposta do controlador, C é a matriz que relaciona os estados às variáveis observáveis, D é a matriz que relaciona a saída do controlador às variáveis observáveis, new_state corresponde ao estado do sistema na iteração $k + 1$ e new_output corresponde à saída do sistema dinâmico na iteração $k + 1$.

6.9 Refdet.m

Garante que o processo de otimização busque o pólo determinado para o sistema, conforme fragmento de código 12.

```
function [ref]=refdet(set,signal,dt)

nd=length(set);
fd=0.1; % fator de decaimento da resposta em direção ao setpoint.
        %ddelta/dt=-fd*delta      range: [0 a inf]
        %                          quanto maior, mais decai
        %                          se for neg: sistema instável
ref=signal;

for i=2:nd
    delta=(signal(i-1)-set(i-1));
    delta=delta*exp(-fd*dt);
    ref(i)=delta+set(i-1);
end

end
```

Fragmento de código 12: programa Refdet.m

onde set é o vetor de referência, $signal$ corresponde ao sinal de entrada da função e ref é o valor de referência na iteração k .

6.10 Calcadapt.m

Realiza o cálculo da função objetivo (E_{RMS}) do processo de minimização da possível solução, recebendo como parâmetro de entrada os valores encontrados pelo método de otimização. O algoritmo está associado a algumas funções descritas nos itens anteriores, como a criação de perturbações no sistema (Disturb.m) e do vetor da variável de referência do sistema (Setpoint.m). Determinadas as condições iniciais das variáveis, ocorre a chamada do código do sistema Neuro-fuzzy, que retorna a ação de controle. Devido à dinâmica dos atuadores, descrita pela Eq. 6.1, a ação de controle deve levar em conta o atraso imposto pelos mesmos.

$$G(s) = \frac{20}{20 + s} \quad (6.1)$$

Discretizando a equação (6.1), sendo Δt o tempo de amostragem do sistema de controle, tem-se

$$y_r(k) = \frac{20\Delta t y_s(k) + y_r(k+1)}{1 + 20\Delta t} \quad (6.2)$$

em que $y_s(k)$ é a ação de controle calculada pelo ANFIS e $y_r(k)$ é a ação de controle de fato percebida pela aeronave, levando em consideração o atraso dos atuadores.

Além disso, é necessário que sejam limitados o erro da variável de entrada e a ação de controle, para que a simulação reproduza uma cópia mais aproximada das condições do mundo real. Esse processo pode ser visualizado conforme fragmento de código 13.

```

for i=2:nt
    if erro_theta>20*pi/180
        erro_theta=20*pi/180;
    elseif erro_theta<-20*pi/180
        erro_theta=-20*pi/180;
    end
    de_s(i)=control_NF(erro_theta,de_s(i-1),prop);
    if de_s(i)>23*pi/180
        de_s(i)=23*pi/180;
    elseif de_s(i)<-23*pi/180
        de_s(i)=-23*pi/180;
    end

    de(i)=(20*T*de_s(i)+de(i-1))/(1+20*T);

```

Fragmento de código 13: programa calcadapt.m – ação de controle

A resposta do sistema é calculada através do programa RespModelo.m e encontrados os erros correspondentes ao novo estado do sistema. Através da função objetivo é realizada a avaliação do controle.

6.11 main.m

A função main.m realiza o cálculo da planta e sua discretização, através dos programas model.me convc2d.m. Os parâmetros do controlador são calculados através do programa de minimização de funções e calcadapt.m retorna os valores de interesse a serem analisados graficamente. É a função principal que realiza toda a coleta dos dados e demonstração através de gráficos que serão explanados no item seguinte.

7 Resultados e Discussões

7.1 Controle Longitudinal

Conforme o item 5.1, o controle longitudinal é dividido em duas partes. Na primeira, ocorre o processo de otimização dos parâmetros da malha interna do controlador, que posteriormente são fixados para a obtenção dos parâmetros da malha externa. A seguir serão apresentados os resultados obtidos para ambas as partes.

7.1.1 Controle Longitudinal – Malha Interna

Possui como parâmetro de entrada do SBRF2 o erro relativo ao ângulo de arfagem de referência da aeronave (θ_{ref}) e sua ação de controle é a deflexão do profundor (ΔPk). O valor do pólo para o sistema é previamente fixado em 0,1. O setpoint do θ é definido como zero em toda a simulação e existem perturbações nos instantes $t = 0s$ e $t = 30s$ que podem ser visualizadas na Fig. 21. A base de regras definida para o controlador ANFIS desse caso é:

$$\begin{aligned} e(\theta k) = \text{PEQUENO} &\rightarrow \Delta Pk = \text{PEQUENO}; \\ e(\theta k) = \text{MÉDIO} &\rightarrow \Delta Pk = \text{MÉDIO}; \\ e(\theta k) = \text{GRANDE} &\rightarrow \Delta Pk = \text{GRANDE}; \end{aligned}$$

O erro RMS apresentado nesse item está relacionado ao ângulo de arfagem do Cessna 172.

7.1.1.1 Evolução Diferencial :

As Figs. 21 e 22 mostram os resultados da simulação no *software* MatLab® relativos à aplicação do método Evolução Diferencial.

Nota-se que, no instante $t = 0s$ existe um pico no ângulo de arfagem proveniente da perturbação de 30 graus. Em conformidade com a base de regras estipulada, é possível verificar que a ação de controle minimiza o efeito da perturbação sobre o sistema. Observa-se que o sistema acomoda-se em torno do valor de referência em $t = 40s$, comprovando na prática a teoria, uma vez que:

$$T_a = 4T \quad (2\%), \quad T = \frac{1}{\text{pólo}} \quad (7.1)$$

onde T_a é o tempo de acomodação do sistema e T a constante de tempo. Para $t > 50s$, o sistema encontra-se em regime permanente.

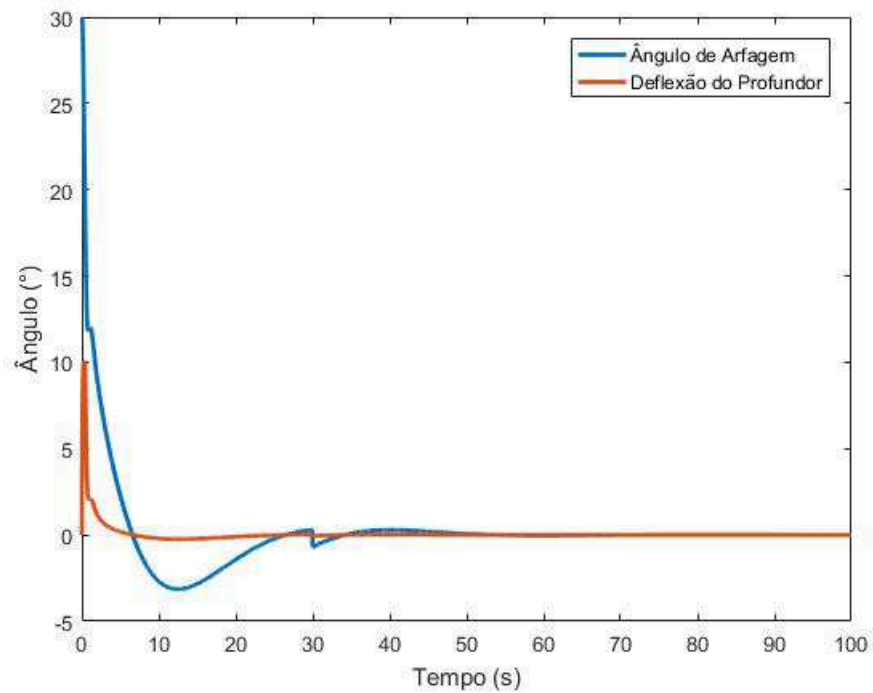


Figura 21: Ângulo de Arfagem e Deflexão do Profundor utilizando Evolução Diferencial

Na Fig. 22 é possível observar na curva o comportamento esperado na evolução do erro RMS ao longo da execução da Evolução Diferencial, decaindo até próximo da iteração $K = 20$ e se tornando invariante até $K = 30$. Porém, nota-se que os valores possuem pequena dimensão devido ao fator de decaimento do pólo desejado, definido anteriormente.

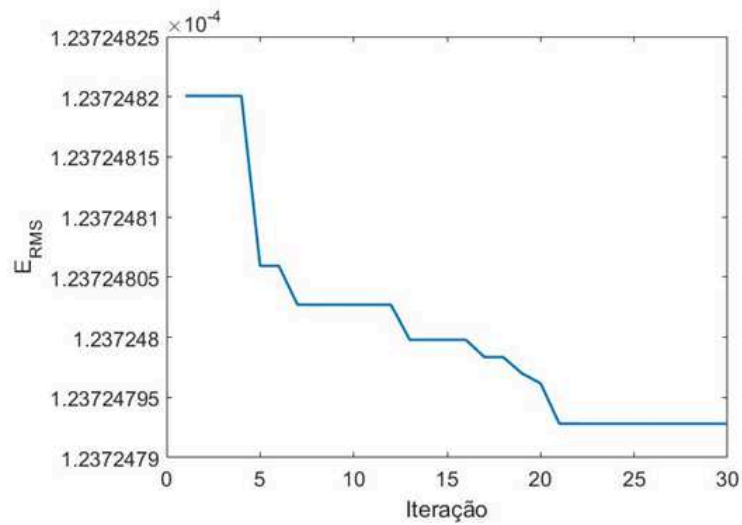


Figura 22: Erro RMS da malha interna do Controle Longitudinal obtido utilizando Evolução Diferencial

7.1.1.2 Recozimento Simulado :

A partir da Fig.23, percebe-se que a resposta do sistema ao controle é mais rápida que a obtida na Evolução Diferencial, uma vez que o ângulo de arfagem responde mais rapidamente à ação de deflexão do profundor. Verifica-se a comprovação da teoria, uma vez que em $t = 40s$ o sistema atinge a acomodação e em $t = 50s$ o mesmo encontra-se efetivamente em regime permanente.

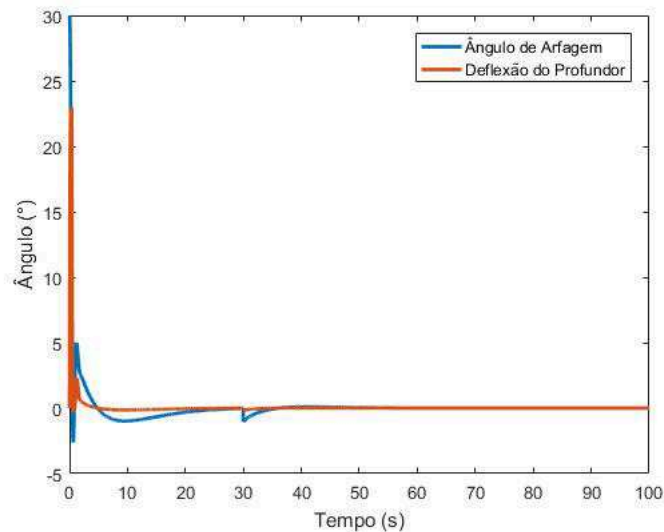


Figura 23: Ângulo de Arfagem e Deflexão do Profundor utilizando Recozimento Simulado

Analisando o erro RMS obtido ao longo da execução do Recozimento Simulado pode-se observar

o comportamento esperado para o gráfico mostrado na Fig. 24. A busca pela minimização da função objetivo permite que, para encontrar o mínimo global da função, exista na iteração $K + 1$ um ponto de maior energia, evitando a seleção de um ponto de mínimo local como um ponto de mínimo global. Devido a esse fato, pode-se observar pontos de pico no comportamento da curva ao longo de toda a simulação até atingir o resultado de mínimo da função. Os valores de erro encontrados para o Recozimento Simulado também não são de grande dimensão, pois acompanham o fator de decaimento do pólo desejado.

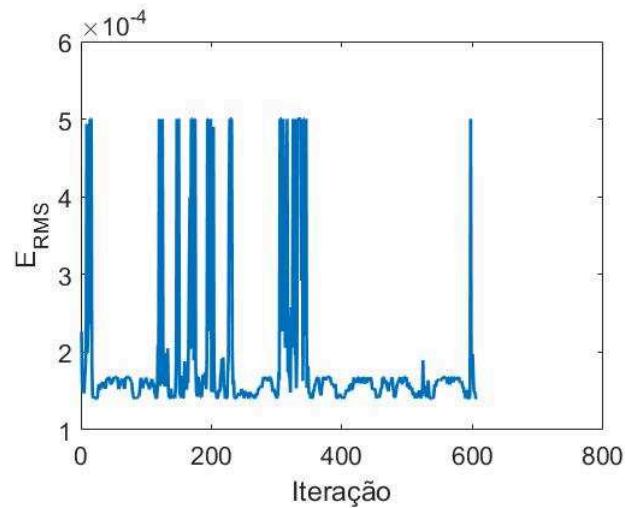


Figura 24: Erro RMS da malha interna do Controle Longitudinal obtido utilizando Recozimento Simulado

A tabela 3 mostra um quadro comparativo entre os dois métodos de minimização de funções analisando tempo de execução da simulação e erro RMS, sendo realizadas cinco simulações para cada método.

Tabela 3: Comparação entre Evolução Diferencial e Recozimento Simulado para malha interna do Controle Longitudinal

	Evolução Diferencial		Recozimento Simulado	
	Média	Desvio Padrão	Média	Desvio Padrão
Tempo de Execução	32,167	1,582	21,073	0,277
Erro RMS	$1,237 \cdot 10^{-4}$	0	$1,670 \cdot 10^{-4}$	$3,718 \cdot 10^{-7}$

O Recozimento Simulado possui uma média e desvio padrão de tempo de execução menores que quando comparado à Evolução Diferencial, porém a média e desvio padrão do erro RMS do primeiro método são menores que os do segundo.

7.1.2 Controle Longitudinal – Malha Externa

Apresenta como parâmetro de entrada para o SBRF1 o erro relativo à altitude do sistema e possui como saída a deflexão do profundor (ΔPk). Os parâmetros de *setpoint* de altitude foram definidos de forma que, para o intervalo de tempo $t = [0, 75]s$, a altitude seja de 50 metros e, para

$t = [75, 150]s$ ela decai novamente para 0 metros. O pólo escolhido para a malha externa é de 0.5. Nesse contexto não são criadas perturbações no sistema, apenas a variação do parâmetro de referência de altura. A base de regras definida para o controlador ANFIS é definida como:

$$e(hk) = \text{PEQUENO} \rightarrow \Delta Pk = \text{PEQUENO};$$

$$e(hk) = \text{MÉDIO} \rightarrow \Delta Pk = \text{MÉDIO};$$

$$e(hk) = \text{GRANDE} \rightarrow \Delta Pk = \text{GRANDE}$$

O erro RMS nesse item refere-se à altitude da aeronave.

7.1.2.1 Evolução Diferencial :

Observa-se na Fig. 25, que no intervalo de $t = [0, 25]s$ a altura se encontra em regime transitório, permanecendo oscilando em torno do ponto de referência de 50 metros. Próximo de $t = 8s$ o sistema atinge o tempo de acomodação, uma vez que $T = 2s$. No intervalo de $t = [25, 50]s$ o sistema se encontra estável. O mesmo comportamento ocorre no intervalo de $t = [0, 75]s$ em torno do ponto de referência de 0 metros. O sistema atinge o regime permanente em $t = 100s$.

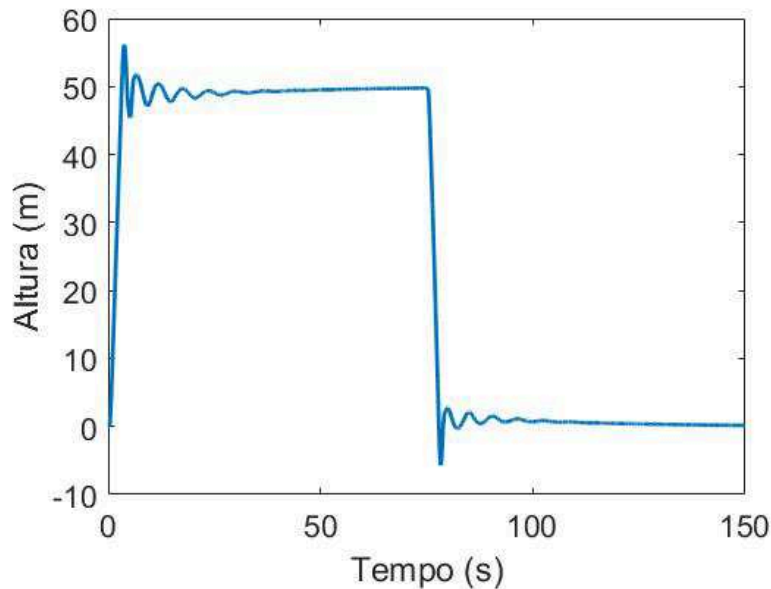


Figura 25: Altura utilizando Evolução Diferencial

Na Fig. 26, o controlador apresenta elevado ganho na presença de grandes perturbações, diminuindo sua magnitude de ação até atingir o regime permanente, quando permanece próximo de zero.

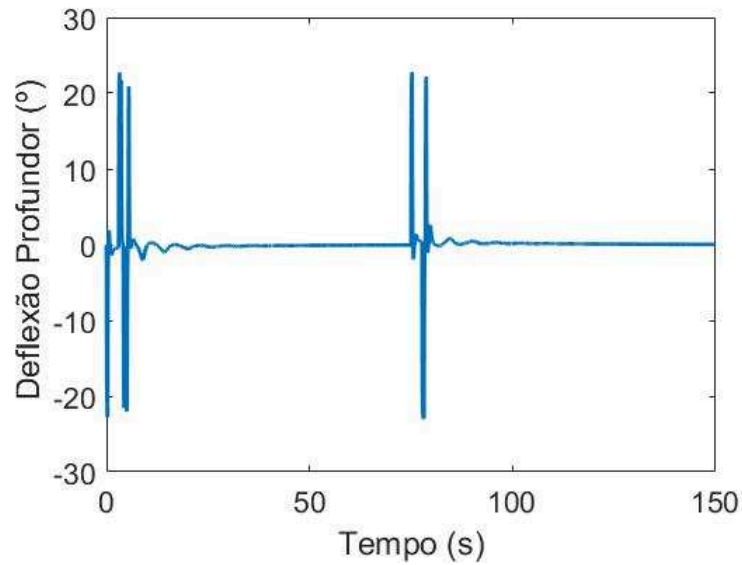


Figura 26: Deflexão do Profundor da malha externa utilizando Evolução Diferencial

A partir da Fig. 27, observa-se que a curva do erro RMS se comporta como o esperado, ocorrendo a minimização da função custo ao longo de toda simulação, se tornando constante a partir da iteração $K = 20$ e garantindo o bom resultado do método. Observa-se que para a malha externa, o erro possui valores maiores comparados aos erros obtidos na malha interna, comprovando a influência do fator de decaimento do pólo desejado, uma vez que o pólo do primeiro caso é 0,1 e o pólo analisado nesse item é de 0,5.

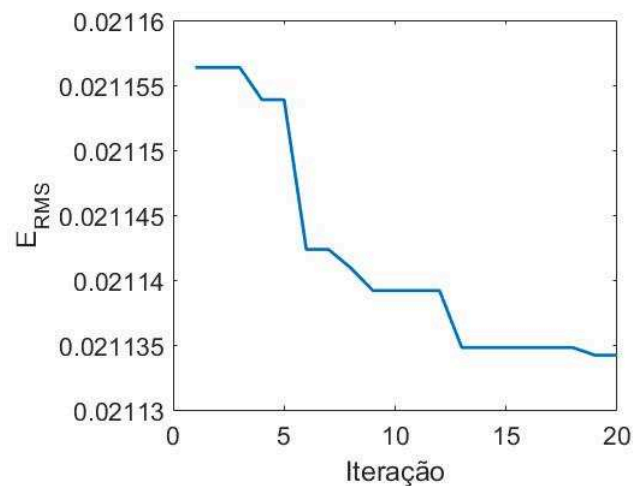


Figura 27: Erro RMS da malha externa do Controle Longitudinal obtido utilizando Evolução Diferencial

7.1.2.2 Rezoimento Simulado :

As curvas de altura e deflexão do profundor observadas nas Fig. 28 e 29 possuem o mesmo comportamento apresentado nas Fig. 28 e 29, obtidas através da Evolução Diferencial.

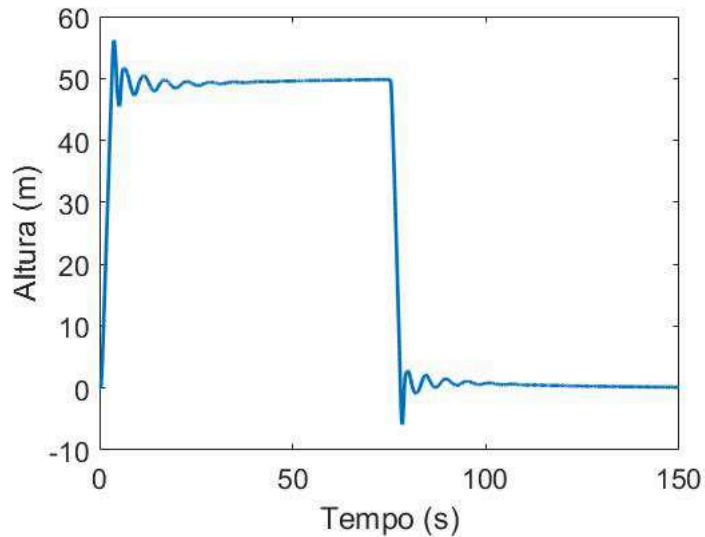


Figura 28: Altura utilizando Rezoimento Simulado

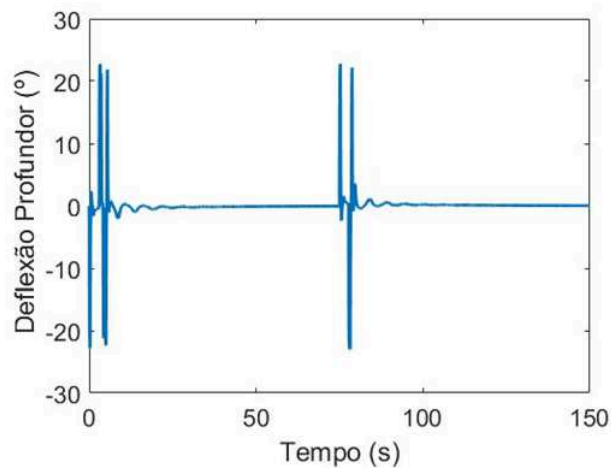


Figura 29: Deflexão do Profundor da malha externa utilizando Rezoimento Simulado

Em comparação ao método da Evolução Diferencial, a partir da Fig. 30, observa-se mudança no comportamento do erro RMS, que possui uma curva que apresenta picos provenientes da procura pelo mínimo global e permissão de pontos maiores de energia na iteração $K + 1$. Os valores apresentados possuem maior magnitude quando comparados à malha interna devido ao fator de decaimento do pólo desejado.

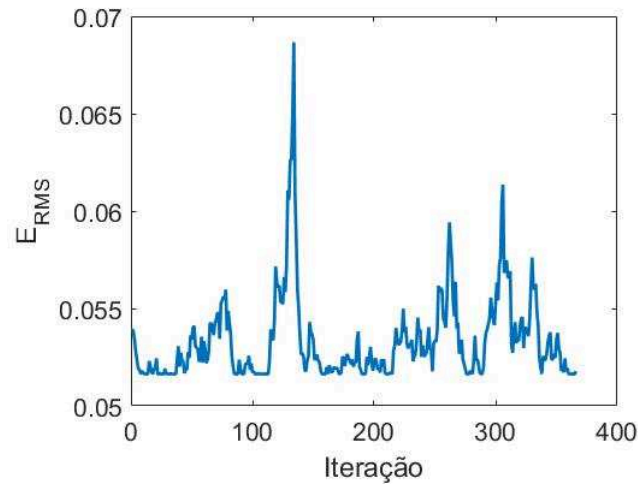


Figura 30: Erro RMS da malha externa do Controle Longitudinal obtido utilizando Recozimento Simulado

Na tabela 4, é possível aferir alguns dados importantes relativos à comparação entre a execução da Evolução Diferencial e do Recozimento Simulado para o cálculo do controle da malha externa. A média e desvio padrão do tempo de execução da simulação para a Evolução Diferencial são maiores comparados ao Recozimento simulado porém, o erro RMS do segundo método se apresenta ligeiramente maior e menos constante para as cinco execuções da simulação. Utilizou-se cinco simulações para a obtenção dos dados.

Tabela 4: Comparação entre Evolução Diferencial e Recozimento Simulado para malha externa do Controle Longitudinal

	Evolução Diferencial		Recozimento Simulado	
	Média	Desvio Padrão	Média	Desvio Padrão
Tempo de Execução	36,383	0,303	32,085	0,191
Erro RMS	0,0211	0	0,022	1,29 10^{-3}

7.2 Controle Látero-direcional

O controle látero-direcional da aeronave é dividido em malha externa e malha interna e seus resultados são apresentados de forma separada. Os parâmetros do controlador da malha interna são encontrados, fixados e posteriormente aplicados à malha externa

7.2.1 Controle Látero-direcional – Malha Interna

A malha interna contém como parâmetro de entrada do SBRF4 o erro relativo ao ângulo de rolagem($e(\Phi_k)$) e como saída do sistema, a deflexão do aileron (Δak). O *setpoint* $\Phi_{ref,k}$ é zero para todo o tempo de simulação, sendo causadas perturbações em dois instantes ($t=0$ s e $t=30$ s). O pólo desejado para esse sistema é 0,6. A base de regras desse sistema é definida como:

$e(\Phi_k) = \text{PEQUENO} \rightarrow \Delta_{ak} = \text{GRANDE};$

$e(\Phi_k) = \text{MÉDIO} \rightarrow \Delta_{ak} = \text{MÉDIO};$

$e(\Phi_k) = \text{GRANDE} \rightarrow \Delta_{ak} = \text{PEQUENO};$

Essa definição de base de regras influencia no comportamento da curva de saída do atuador, uma vez que a saída deve ser contrária à entrada do sistema.

7.2.1.1 Evolução Diferencial :

Na Fig. 31 observa-se um pico positivo na curva do ângulo de rolagem no instante $t = 0s$ causado pela perturbação aplicada ao sistema, a resposta do controlador possui ação contrária, em conformidade com a base de regras do SBRF4 e definida pelos limites físicos de deflexão do atuador. É possível perceber que no intervalo de $t = [50, 100]s$ o sistema encontra-se em regime permanente.

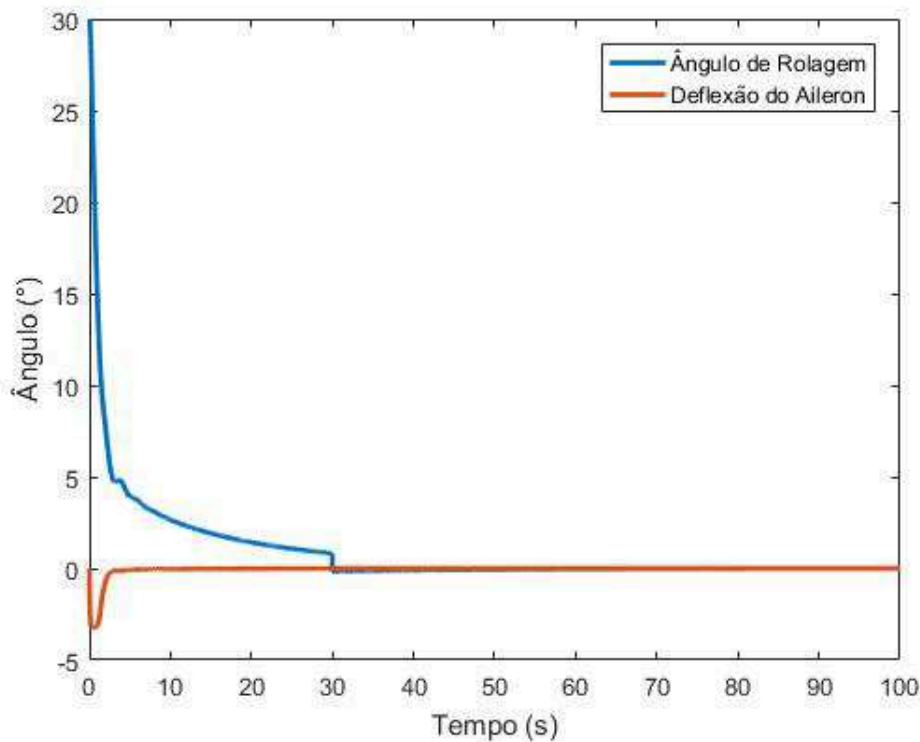


Figura 31: Ângulo de Rolagem e Deflexão do Aileron utilizando Evolução Diferencial

A partir da Fig. 32, observa-se que a curva do erro RMS se comporta de forma esperada, com o decaimento da função objetiva ao longo de toda a simulação, permanecendo quase constante próximo da iteração $K = 30$. A magnitude dos valores apresentados sofre grande influência do fator de decaimento do pólo desejado definido.

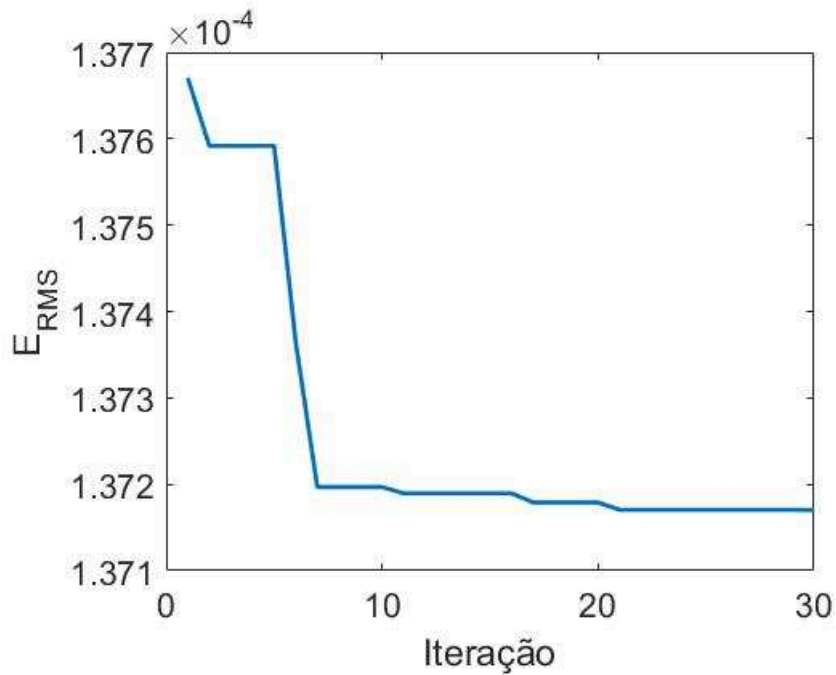


Figura 32: Erro RMS da malha interna do Controle Látero-direcional obtido utilizando Evolução Diferencial

7.2.1.2 Recozimento Simulado :

As curvas de ângulo de rolagem e deflexão do aileron, apresentadas na Fig. 33, apresentam variações quando comparados à Fig. 31, obtida a partir da Evolução Diferencial. O ângulo de rolagem possui uma curva de decaimento suave, sem ocorrência de overshoot para o primeiro método, no segundo método, oscila em torno do ponto de referência até atingir o regime permanente. Observa-se que, no instante próximo de $t = 7s$, é atingido o tempo de acomodação. O sistema atinge o regime permanente no intervalo $t = [10, 100]s$.

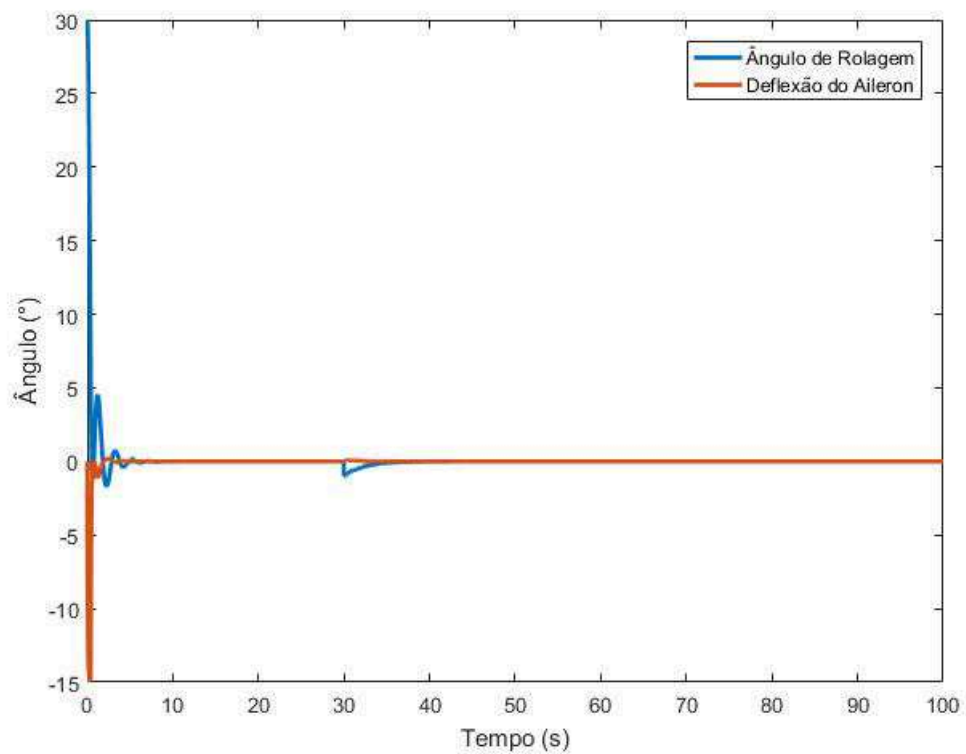


Figura 33: Ângulo de Rolagem e Deflexão do Aileron utilizando Recozimento Simulado

A partir da Fig. 34, pode-se observar que a curva do erro RMS possui a ocorrência de picos devido à busca por mínimos globais e se comporta de forma esperada para o método. O pólo desejado influencia nos valores obtidos.

Tabela 5: Comparação entre Evolução Diferencial e Recozimento Simulado para malha interna do Controle Látero-direcional

	Evolução Diferencial		Recozimento Simulado	
	Média	Desvio Padrão	Média	Desvio Padrão
Tempo de Execução	31,450	0,277	20,944	0,111
Erro RMS	$1,327 \cdot 10^{-4}$	0	$1,575 \cdot 10^{-4}$	$2,781 \cdot 10^{-6}$

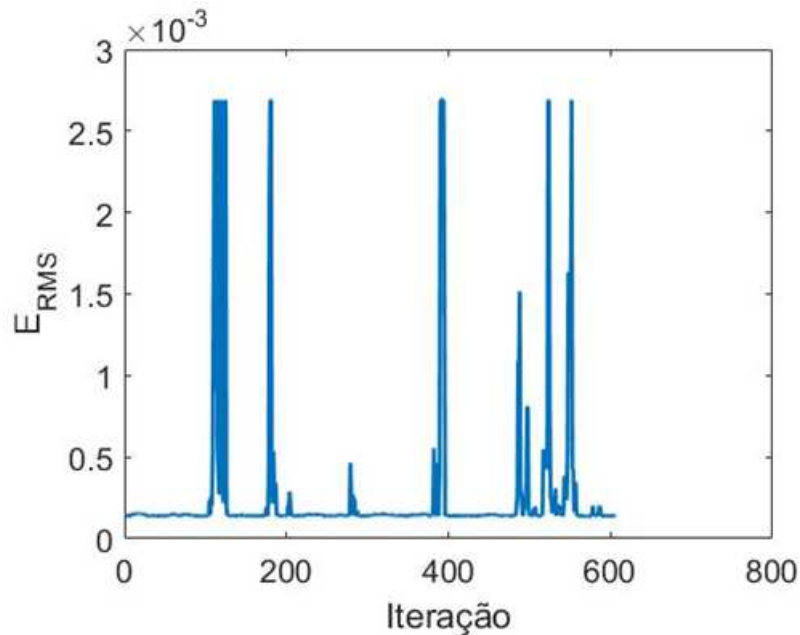


Figura 34: Erro RMS da malha interna do Controle Látero-direcional obtido utilizando Recozimento Simulado

Segundo a tabela 5 pode-se observar um menor tempo de execução da simulação com menor desvio padrão para o segundo método, em contrapartida, o erro RMS encontrado para o primeiro método se apresenta menor e constante, com desvio padrão 0. Foram realizadas cinco simulações para a obtenção dos dados apresentados.

7.2.2 Controle Látero-direcional - Malha Externa

A malha externa do controle látero-direcional tem como entrada para os SBRF correspondentes, o erro relativo ao ângulo de guinada ($e(\Psi_k)$) e produz duas saídas no sistema, a ação de controle por meio da deflexão de dois atuadores da aeronave: o aileron (Δa_k) e o leme (Δr_k). Os parâmetros do controlador da malha interna são calculados, fixados e aplicados ao SBRF3 para o cálculo do seu controlador, produzindo como saída a deflexão do aileron. Sua base de regras é definida como:

$$e(\Psi_k) = \text{PEQUENO} \rightarrow \Delta a_k = \text{PEQUENO};$$

$$e(\Psi_k) = \text{MÉDIO} \rightarrow \Delta a_k = \text{MÉDIO};$$

$$e(\Psi_k) = \text{GRANDE} \rightarrow \Delta a_k = \text{GRANDE};$$

O SBRF5 é independente da malha interna, sendo utilizado como parâmetros de entrada para o cálculo do seu controlador apenas o $e(\Psi_k)$, produzindo como saída Δr_k . Possui como base de regras:

$$e(\Psi_k) = \text{PEQUENO} \rightarrow \Delta r_k = \text{GRANDE};$$

$$e(\Psi_k) = \text{MÉDIO} \rightarrow \Delta r_k = \text{MÉDIO};$$

$$e(\Psi_k) = \text{GRANDE} \rightarrow \Delta r_k = \text{PEQUENO};$$

O *setpoint* de Ψ , $\Psi_{ref,k}$ é definido como 90 graus para o intervalo $t = [0, 50]s$ e depois retorna a 0 graus para $t = [50, 100]s$. Não são aplicadas perturbações para esse caso e o pólo desejado escolhido possui o valor de 0,5.

7.2.2.1 Evolução Diferencial :

A partir da Fig. 35 observa-se que o ângulo de guinada da aeronave possui uma curva suave com pequenas oscilações para se estabilizar no *setpoint* desejado. No instante de $t = 8s$ é atingido o tempo de acomodação. O sistema atinge o regime permanente próximo ao instante $t = 30s$, permanecendo estável no intervalo $t = [30, 50]s$ em torno do ponto de referência de 90 graus. Para $t = [50, 100]s$, a resposta do ângulo de guinada possui o mesmo comportamento, atingindo a estabilidade em torno de 0 graus.

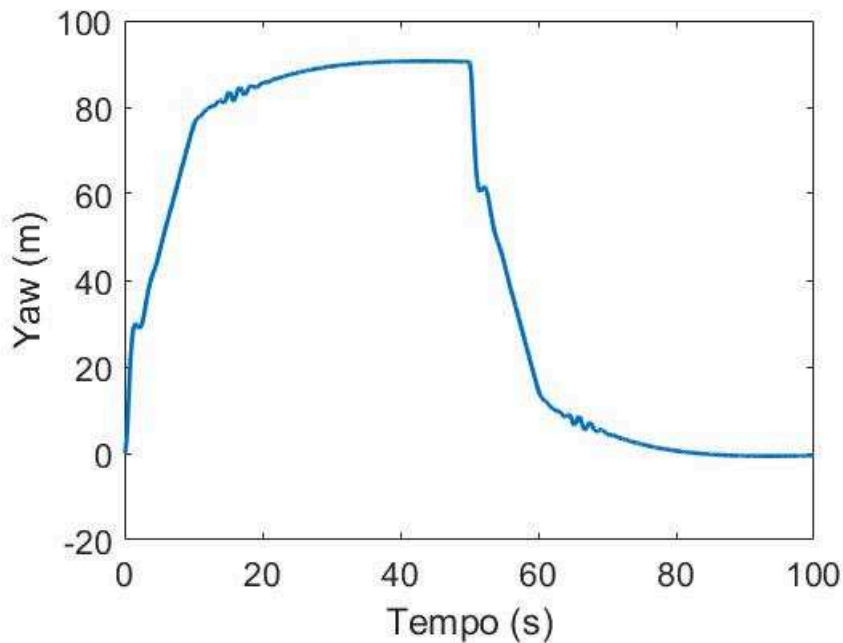


Figura 35: Ângulo de Guinada utilizando Evolução diferencial

Nota-se que as ações de controle fornecidas pela deflexão do aileron e do leme, que podem ser observadas na Fig. 36, apresentam ganho elevado nos instantes de mudança do ponto de referência e são quase nulas próximos do instante de estabilidade do sistema ($t = 30s$ e $t = 70s$)

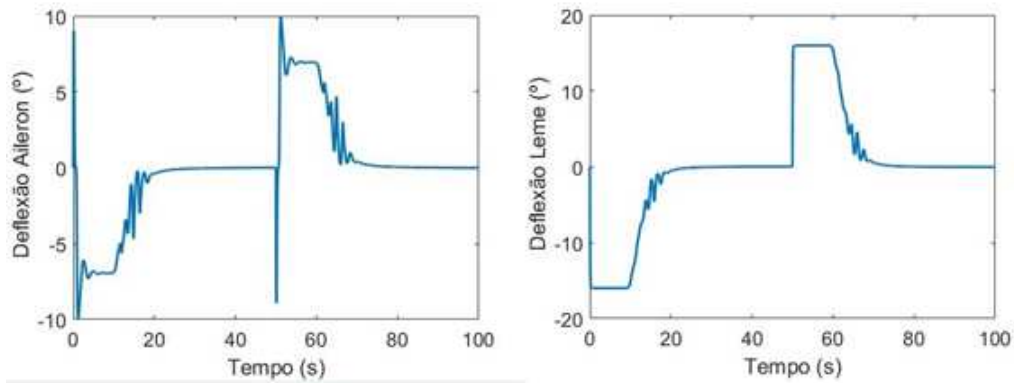


Figura 36: Deflexão do Aileron e Deflexão do Leme utilizando Evolução diferencial

Observa-se na 37 que o erro RMS possui decaimento ao longo de toda a simulação, confirmando a execução correta do método e se estabilizando próximo da iteração $K=18$. O valor do erro sofre grande influencia do fator de decaimento do pólo desejado.

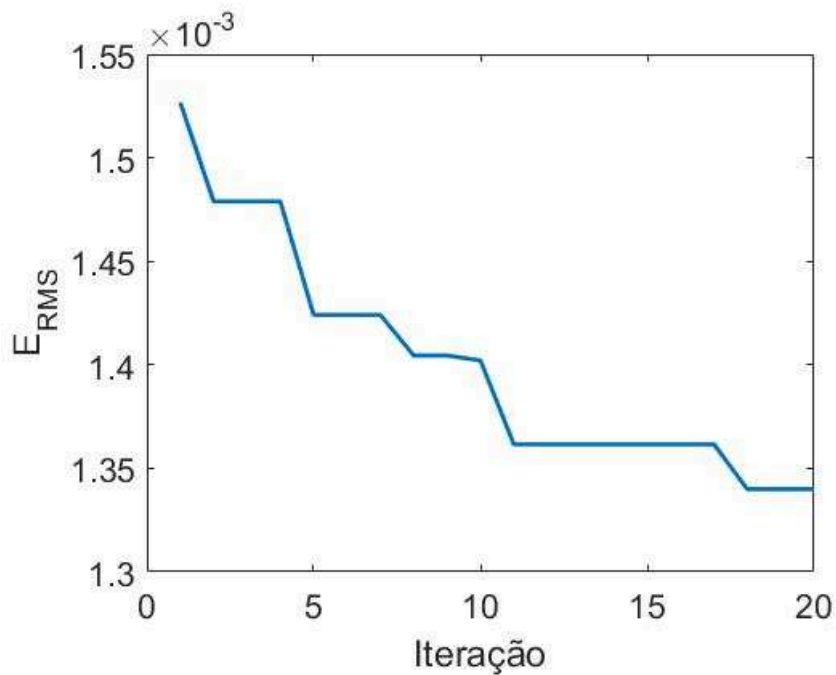


Figura 37: Erro RMS da malha externa do Controle Látero-direcional obtido utilizando Evolução Diferencial

7.2.2.2 Recozimento Simulado :

Observa-se comparando as Fig. 35 e 38, que a resposta do ângulo de guinada não se comporta da forma esperada, não atingindo a acomodação no instante $t=8s$, conforme teoria.

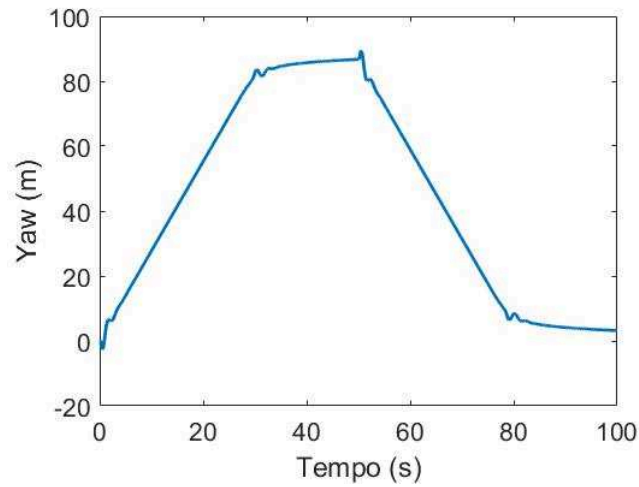


Figura 38: Ângulo de Guinada utilizando Recozimento Simulado

A partir da Fig. 39, é possível observar que a ação de controle por meio da deflexão do aileron ocorre de uma forma mais abrupta em determinados instantes e permanece próximo de 0 graus em um maior intervalo de tempo de simulação. A deflexão do leme ocorre de forma mais suave.

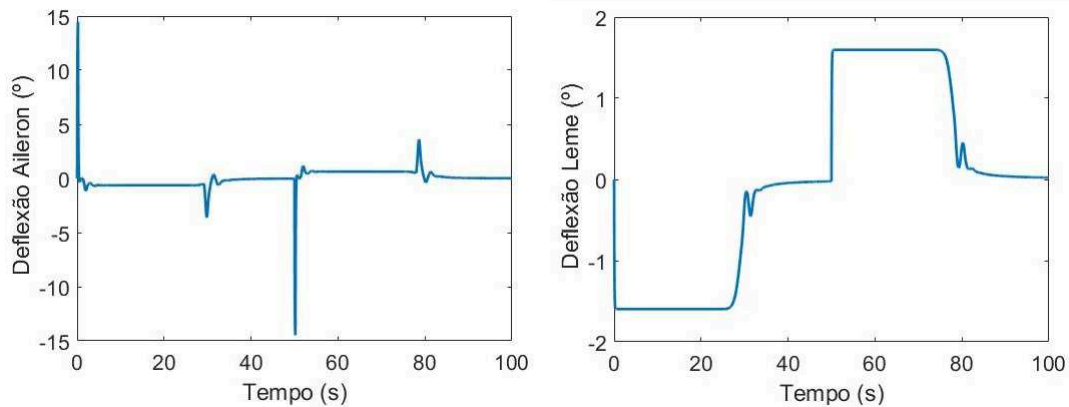


Figura 39: Deflexão do Aileron e Deflexão do Leme utilizando Recozimento Simulado

A evolução do erro RMS ocorre de forma esperada, apresentando picos de máximo globais e locais em função da busca pelo mínimo global e sofrendo influencia do fator de decaimento do pólo desejado, conforme Fig. 40.

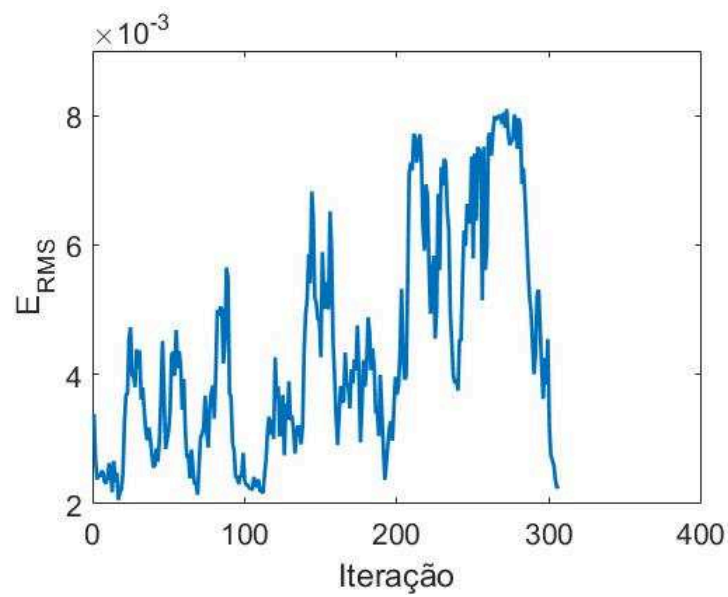


Figura 40: Erro RMS da malha externa do Controle Látero-direcional obtido utilizando Recozimento Simulado

A partir da tabela 6 observa-se que o primeiro método possui um tempo de execução e desvio padrão maiores que o segundo método, em contrapartida, o erro RMS obtido da minimização da função objetivo é menor e possui um desvio padrão bem menor se comparado ao do segundo método.

Tabela 6: Comparação entre Evolução Diferencial e Recozimento Simulado para malha externa do Controle Látero-direcional

	Evolução Diferencial		Recozimento Simulado	
	Média	Desvio Padrão	Média	Desvio Padrão
Tempo de Execução	33,602	0,450	24,435	0,305
Erro RMS	$1,32 \cdot 10^{-3}$	$4,472 \cdot 10^{-5}$	$3,32 \cdot 10^{-3}$	$1,173 \cdot 10^{-3}$

8 Conclusão e trabalhos futuros

O trabalho apresenta a aplicação de técnicas de controle inteligente na análise e controle do avião Cessna 172 em escala real, para isso faz-se necessária a realização da modelagem dinâmica da aeronave e do sistema de controle utilizado. São realizadas simulações envolvendo dois métodos diferentes de otimização, utilizados na obtenção dos parâmetros do controlador.

Observa-se que, para todas as análises de sistemas realizadas, o controlador agiu de forma efetiva, produzindo na entrada uma resposta evoluindo ao encontro do ponto de referência definido.

Ambos os métodos de otimização utilizados se mostraram capazes de encontrar os parâmetros para se obter um controlador otimizado, porém, demonstraram características que diferem entre si e devem ser analisadas conforme comportamento e necessidades do problema a ser estudado. A Evolução Diferencial é um método que se mostra assertivo no cálculo do controlador, produzindo menores erros e uma constância na apresentação dos resultados, com desvio padrão relacionado ao erro RMS igual a zero para diferentes simulações. Apesar disso, possui um tempo de execução maior, comprometendo a eficiência do controlador na hipótese de ser utilizado em aplicação *online*. O Recozimento Simulado apresenta maior dispersão no erro RMS, podendo influenciar na resposta do controlador a cada simulação realizada. Em contrapartida, a média e o desvio padrão do tempo de execução do método é inferior, viabilizando sua utilização em aplicação *online*.

Vislumbrando trabalhos futuros, pretende-se observar o comportamento dos dois métodos apresentados quando utilizadas outras funções objetivo, a fim de melhorar a resposta do sistema controlado. Ainda, o desenvolvimento do controle de forma *online* se faz pertinente, uma vez que confere a adaptabilidade do controlador a condições extremas que modificam o comportamento do sistema.

9 Referências Bibliográficas

BATISTA, Rafael Gomes; MENEGHETTI, Carlos Renato. Sistema comparativo de controle inteligente de nível aplicado às lógicas *fuzzy* e pid. Unisal, jun. 2018.

BRASIL ESCOLA. Inteligencia artificial. Disponível em: <<https://brasilecola.uol.com.br/informatica/inteligencia-artificial.htm>>. Acesso em: 10 jan. 2019.

CONTEUDO - USP. Redes neurais artificiais. Disponível em: <<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>>. Acesso em: 29 mai. 2018.

FOSSEN, Thor I. Mathematical models for control of aircraft and satellites: subtítulo do livro. 2 ed. Department of Engineering Cybernetics, NTNU, 2001. 31 p.

FTP - UNICAMP. Redes neurais artificiais. Disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf>. Acesso em: 29 mai. 2018.

Gianelli, C. A., “Estudo Comparativo entre Controle Adaptativo e Controle Fuzzy”, Monografia, Universidade de São Paulo, São Carlos, SP, Brasil, 2013.

GOMIDE, Fernando A. C.; GUDWIN, Ricardo R.; TANSCHKEIT, Ricardo. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. Proceedings of 6th international fuzzy systems association (ifsa) conference, São paulo, 1995.

HAN, Ming-Feng; LIN, Chin-Teng; CHANG, Jyh-Yeong. Differential Evolution With Local Information for Neuro-Fuzzy Systems Optimisation. Knowledge-Based Systems, Taiwan, v. 44, p. 78-89, fev. 2013.

HAYKIN, Simon. Redes neurais : Princípios e prática. 2 ed. Hamilton, Ontário, Canadá: Bookman, 2001.

KIRKPATRICK, S.; JR., C. D. Gelatt; VECCHI, M. P.. Optimization by simulated annealing. Science , [S.L.], v. 220, n. 4598, p. 671-680, mai. 1983.

MACHINI, F. M. M., “Modelagem, Simulação e Controle de uma Aeronave Multirrotora”, Monografia, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2016.

MAXWELL - PUC - RIO. Sistemas neuro-fuzzy. Disponível em: <https://www.maxwell.vrac.puc-rio.br/32823/32823_5.pdf>. Acesso em: 30 mai. 2018.

MTC - M . Capítulo 5 recozimento simulado (simulated annealing). Disponível em: <<http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19@80/2010/01.20.19.25/doc/cap5.pdf?metadatarpository=&mirror=sid.inpe.br/mtc-m19@80/2009/08.21.17.02.53>>. Acesso em: 25 mai. 2018.

O FUTURO DAS COISAS. Robôs agora podem visualizar o próprio futuro. Disponível em: <<https://ofuturodascoisas.com/robos-agora-podem-visualizar-o-proprio-futuro/>>. Acesso em: 10 jan. 2019.

PADILHA, Bruno Ricardo; ZAIONS, Douglas Roberto; SPULDARO, Everton. Projeto aerodinâmico, estabilidade e controle de um veículo aéreo não tripulado (vant) de asa fixa. Unoesc, jan. 2012. Disponível em: <<https://editora.unoesc.edu.br/index.php/acet/article/download/1515/pdf>>. Acesso em: 10 jan. 2019.

PEQ DR. Método de recozimento simulado (simulated annealing). Disponível em: <http://www2.peq.coppe.ufrj.br/pessoal/professores/arge/coq897/naturais/aulas_piloto/aula2.pdf>. Acesso em: 25 mai. 2018.

PEREIRA, B. L. “Aplicação de Sistemas Neuro-Fuzzy e Evolução Diferencial na Modelagem e Controle de Veículo de Duas Rodas”, Dissertação, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2017

ROSKAM, Jan; LAN, Chuan-tau Edward. Airplane aerodynamics and performance. 1 ed.

KANSAS: Design, Analysis and Research Corporation(DARcorporation), 1997. 699 p.

SAS. Inteligencia artificial. Disponível em: <https://www.sas.com/pt_br/insights/analytics/inteligencia-artificial.html>. Acesso em: 10 jan. 2019.

SISTEMA CENIPA. Aviões sumário estatístico. Disponível em: <http://sistema.cenipa.aer.mil.br/cenipa/paginas/arquivos/avioes_sumario_estatistico.pdf>. Acesso em: 10 jan. 2019.

STORN, Rainer; PRICE, Kenneth. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Netherlands, v. 11, p. 341-359, Novembro 1996.

TAFNER, Malcon Anderson. Redes neurais artificiais: aprendizado e plasticidade,. Cérebro&mentev. 2, n. 5, Março/Maio 1998.

USPNET- USP. Sistemas fuzzy lógica fuzzy e sistemas baseados em regras fuzzy. Disponível em: <http://each.uspnet.usp.br/sarajane/wp-content/uploads/2015/06/2015-sistemas_fuzzy.pdf>. Acesso em: 29 mai. 2018 .

Valavanis, K.P., 2007. Advances in Unmanned Aerial Vehicles. Vol. 33, Springer, ISBN 978-1-4020-6113-4.

ZADEH, L.a. Fuzzy sets*. Information and control, v. 8, n. 3, p. 338-353, Junho 1965.