



**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO**

**COMITÊ DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
TECNOLÓGICA - CPPGIT**

6ª REUNIÃO ORDINÁRIA DE 2026

Data: 4 de agosto de 2026 (terça-feira).

Horário: 8h30min.

Local: Videoconferência.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

CONVOCAÇÃO

A Pró-Reitora Adjunta de Pesquisa e Pós-Graduação da Universidade Federal Rural do Semi-Árido convoca todos os membros do Comitê de Pesquisa, Pós-Graduação e Inovação Tecnológica (CPPGIT) a se fazerem presentes à 6ª Reunião Ordinária, com data, local e horário abaixo determinados, para cumprir a seguinte pauta:

1. Apreciação e deliberação sobre a criação dos Programas Gerais de Componente Curricular – PGCCs das disciplinas: “Apicultura Avançada” e “Comportamento em Abelhas: Ênfase em Africanizadas e Sem Ferrão” do Programa de Pós-Graduação em Ciência Animal.
2. Apreciação e deliberação sobre o projeto de pesquisa "PI00001-2026 - Percepção de risco ambiental e atitudes frente ao rejeito salino da dessalinização no semiárido brasileiro", proposto pelo professor visitante Heitor de Oliveira Braga.
3. Apreciação e deliberação sobre a solicitação do diploma do egresso Augusto César de Sousa do Programa de Pós-Graduação em Engenharia Elétrica.
4. Apreciação e deliberação sobre a pauta da 6ª Reunião Ordinária do Consepe.
Segue link para acessar os documentos da reunião do Consepe:
<https://conselhos.ufersa.edu.br/convocacoes-pastas-e-atas-consepe-2026/>
5. Outras ocorrências.

Data: 4 de agosto de 2026 (terça-feira).

Horário: 8h30min.

Local: Videoconferência.

Mossoró-RN, 28 de julho de 2026.

Alexsandra Fernandes Pereira
Pró-Reitora Adjunta de Pesquisa e Pós-Graduação



Universidade Federal Rural do Semi-Árido (UFERSA)
Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG)
6ª Reunião Ordinária de 2026 do CPPGIT

1º PONTO

Apreciação e deliberação sobre a criação dos Programas Gerais de Componente Curricular – PGCCs das disciplinas: “Apicultura Avançada” e “Comportamento em Abelhas: Ênfase em Africanizadas e Sem Ferrão” do Programa de Pós-Graduação em Ciência Animal.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

PROGRAMA GERAL DE DISCIPLINA					
IDENTIFICAÇÃO					
CURSO(S)			DEPARTAMENTO/CENTRO		
Programa de Pós-Graduação em Ciência Animal			Centro de Ciências Agrárias		
NOME DO COMPONENTE CURRICULAR					
Apicultura Avançada					
CÓDIGO		SITUAÇÃO		POSIÇÃO NA INTEGRALIZAÇÃO	
PPA0234		Obrigatória ()		Semestral ()	
		Optativa ()		Anual (x)	
DOCENTE(S)					
Kátia Peres Gramacho					
CARGA HORÁRIA				CARGA HORÁRIA TOTAL	
TEÓRICA	PRÁTICA	TEÓRICO- PRÁTICA		TOTAL	NÚMERO de CRÉDITOS (15h/aula = 1 Crédito)
27H	33H	60		4	60H
OBJETIVOS DA DISCIPLINA					
Capacitar pós-graduandos da UFERSA e outras IES para assumirem cursos de apicultura em nível de graduação e extensão, com base em fundamentos técnico-científicos da biologia, morfologia, fisiologia das abelhas; flora, polinização e sanidade apícola; domínio de técnicas de manejo avançadas aplicadas à exploração apícola com alta produtividade e qualidade; composição, legislação e processamento dos diferentes produtos das abelhas; seleção de linhagens mais produtivas e resistentes às diferentes parasitoses das abelhas, bem como produção de rainhas. Situar a organização apícola no Brasil					
EMENTA					
Abelhas no mundo e no Brasil. Abelhas não <i>Apis</i> no Brasil e sua importância. Morfologia externa e interna das abelhas. Fisiologia do indivíduo e social das abelhas. Biologia e Desenvolvimento das abelhas. Atividades e comportamentos das abelhas. Flora Apícola. Polinização. Sanidade Apícola. Manejo para alta produtividade e qualidade dos produtos apícolas. Composição e uso dos produtos apícolas. Legislações brasileiras aplicadas à apicultura e meliponicultura. Organização do Associativismo, Cooperativismo, Cadeia Apícola e Empreendedorismo Apícola no Brasil.					



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

CON
TEÚ
DO
DO
PRO
GRA
MÁ
TIC
O

UNIDADE I		T	P	T-P
I	Fundamentos e construção de projetos	2	0	2
	Histórico e sistemática da apicultura e panorama mundial	2	0	2
	Noções de Revisão sistemática e cienciometria	2	2	4
	Bioclimatologia em abelhas	2	0	2
	Construção de projetos e pesquisas e protocolos	2	0	2
	Flora apícola e calendário apícola	1	3	4
UNIDADE II		T	P	T-P
II	Divisão de colônias por diferentes métodos	1	3	4
	Produção de rainhas	1	3	4
	Sanidade apícola: um só saúde	2	0	2
	Nutrição e alimentação na entressafra	2	2	4
	Produtos apícolas: composição, colheita e aproveitamento	2	2	4
	Legislação aplicada à apicultura e meliponicultura	2	0	2
UNIDADE III		T	P	T-P
III	Pesquisa bibliográfica e trabalhos avaliativos	0	4	4
	Melhoramento genético	2	0	2
	Associativismo, cooperativismo e empreendedorismo apícola	2	0	2
	Coleta de dados de campo	0	4	4
	Atividade de integração prática: dia de campo ou visita técnica	0	4	4
	Produção de protocolos práticos	2	2	4
	Apresentação de projetos	0	4	4

MÉTODOS		
TÉCNICAS	RECURSOS DIDATICOS	INSTRUMENTOS DE AVALIAÇÃO
- Exposição dialogada com uso de recursos audiovisuais - Seminários temáticos com base em artigos científicos - Aulas práticas de campo e laboratório (produção	- Projetor multimídia, computador e internet; - Equipamentos e materiais apícolas (colmeias, caixas de rainhas, seringas de transferência etc.); - Laboratórios da UFERSA e a Fazenda Experimental Rafael Fernandes, apiários e	- A avaliação será composta por atividades com pesos específicos, totalizando 10,0 pontos. Os critérios consideram o desempenho individual e coletivo nas





MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

- de rainhas, coleta e análise de pólen, sanidade dentre outras)
- Revisão sistemática e levantamento da literatura cinzenta
- Oficinas de escrita científica e de construção de projetos e protocolos
- Discussões orientadas por perguntas-chave
- Estudo de caso com apicultura e meliponicultura
- Pesquisadores convidados nacionais e internacionais para fazerem aulas em temas específicos
- meliponários;
- Plataformas digitais para pesquisa científica (CAPES, Web of Science, Google Scholar)
- Fichas analíticas padronizadas e protocolos de prática;
- Textos científicos atualizados fornecidos pelos docentes
- Microscópios e lâminas para análises laboratoriais
- tarefas práticas, teóricas e científicas.
- Elaboração de projeto de iniciação científica completo com orçamento (alinhado à dissertação ou tese) e apresentação oral;
- Relatório técnico da aulas práticas
- Projeto de qualificação de mestrado (quando aplicável)
- Revisão sistemática e levantamento da literatura cinzenta (para doutorandos)
- Fichas analíticas de leitura de artigos e protocolos de práticas
- Participação nas discussões e atividades práticas

REFERÊNCIAS BIBLIOGRÁFICAS (ABNT 2000)

BIBLIOGRAFIA BÁSICA

- CRANE, E. The World History of Beekeeping and Honey Hunting. Duckworth. 2000.
- GOODMAN, L. Form and Function in the Honey Bee. IBRA. 2003.
- GRAHAM, J.M. (Ed). The hive and the honeybee. Ext. Rev. Hamilton, Illinois: Dadant & Sons Inc., 1992.
- HEPBURN, H. R.; RADLOFF, S. E. Honeybees in Africa. 1. Springer. 1998.
- LAIDLAW JR., H.H. Criação Contemporânea de Rainhas(trad.). Edit. La Salle, Canoas-RS. 1998.
- LANGSTHOTH, L.L. A Practical Treatise on the Hive and Honey-Bee. 1860. Third Ed. Revised, and Illustrated with Seventy-seven Engraving. Northern Bee Books. 2014
- MICHENER, C.D. The Social Behavior of the Bees. Harvard University Press. 1974.
- MORITZ, R.F.A.; SOUTHWICK, E.E. Bees as Superorganisms. An Evolutionary Reality. Springer-Verlag. 1992.
- PAGE, JR., R. The Spirit of the Hive: The Mechanism of Social Evolution. Harvard University Press. 2013.
- RINDERER, R.E. Bee Genetics and Breeding. Academic Press, Inc. 1986.
- SEELEY, T.D. The Wisdom of the Hive. The Social Physiology of





Honey Bee Colonies. Harvard University Press.

- SEELEY, T. D. Ecologia da Abelha: Um estudo de adaptação na vida social. Paixão Editores. 2006.
- SEELEY, T. D. Honey Bee Democracy. Princeton University Press. 2010.
- SPIVAK, M.; FLETCHER, D.J.C.; BREED, M.C. The African Honey Bee. Westview Studies in Insect Biology. 1. Westview Press. 1991.
- TAUTZ, J. O Fenômeno das Abelhas. Artmed. 2010.
- Von FRISCH, K. The dance Language and Orientation of Bees. Transl. By L. E. Chadwick with a new foreword by T.D. Seeley. Harvard University Press. 1993
- WISNTON M. K. Biologia de Abelhas (trad). Magister, PA. 2003

BIBLIOGRAFIA COMPLEMENTAR

- Anais dos Encontros sobre Abelhas de Ribeirão Preto-
<https://rge.fmrp.usp.br/encontro/anais.php>
- FONSECA, V.L.I.; CANHOS, D.A.L; ALVES, D.A.; SARAIVA, A.M.(Org.) POLINIZADORES NO BRASIL: Contribuição e Perspectivas para a Biodiversidade, Uso Sustentável, Conservação e Serviços Ambientais. EDUSP, 2012.
- FONSECA, V.L.I.; SARAIVA, A.M.; DE JONG, D.(Ed). Bees as Pollinators in Brazil. Editora Holos, 2006.
- FREITAS, B.M.; PINHEIRO, J.N. Polinizadores e Pesticidas: Princípios e Manejo para os Agroecossistemas Brasileiros. Brasília: MMA, 2012.
<https://www.semabelhasemalimento.com.br/wp-content/uploads/2015/02/polinizadores-e-pesticidas-final-pdf>
- LANDIM, C.C. Abelhas: morfologia e função de sistemas. Editora UNESP, 2009.
- SILVEIRA, F.A.; MELO, A.R.; ALMEIDA, E.A.B. Abelhas Brasileiras: sistemática e identificação. Belo Horizonte, 2002.
https://www.researchgate.net/publication/280112185_Abelhas_Brasileiras_Sistemtica_e_Identificacao

PERIÓDICOS:

- **Journal of Apicultural Research:**
<https://www.ibrabee.org.uk/journal-of-apicultural-research>
- **Apidologie:** <https://link.springer.com/journal/13592>
- **Journal of Economic Entomology:**
<https://academic.oup.com/jee>
- **Science:** <https://www.science.org/journal/science>
- **Nature:** <https://www.nature.com/>
- **PLOS ONE:** <https://journals.plos.org/plosone/>

APROVAÇÃO PELO COLEGIADO



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

DATA	COORDENADOR DO PPGCA
APROVAÇÃO PELO CONSEPE	
DATA	COORDENADOR DO PPGCA



UFERSA | 20 ANOS



PROGRAMA GERAL DE DISCIPLINA				
IDENTIFICAÇÃO				
CURSO(S)		DEPARTAMENTO/CENTRO		
Programa de Pós-Graduação em Ciência Animal		Centro de Ciências Agrárias		
NOME DO COMPONENTE CURRICULAR				
Comportamento em Abelhas: Ênfase em Africanizadas e Sem Ferrão				
CÓDIGO	SITUAÇÃO		POSIÇÃO NA INTEGRALIZAÇÃO	
	Obrigatória ()		Semestral ()	
	Optativa ()		Anual (x)	
CARGA HORÁRIA (60h)				CARGA HORÁRIA TOTAL NÚMERO de CRÉDITOS (15h/aula = 1 Crédito)
TEÓRICA	PRÁTICA	TEÓRICO- PRÁTICA	TOTAL	
30	30	60	60	4
OBJETIVOS DA DISCIPLINA				
Capacitar os alunos a compreender, analisar e aplicar os principais aspectos comportamentais das abelhas sociais (africanizadas e sem ferrão), relacionando-os com fatores biológicos, genéticos, ecológicos, ambientais e produtivos. Desenvolver competências para a observação, registro e análise do comportamento animal utilizando métodos científicos padronizados, incluindo protocolos do COLOSS BEEBOOK e internacionais. Integrar ciência básica e aplicada para propor estratégias de manejo, melhoramento genético, conservação e promoção da saúde das colônias, considerando estressores bióticos e abióticos no semiárido. Estimular habilidades práticas em campo e laboratório, incluindo construção de etogramas, análises de termorregulação, comportamento forrageiro e técnicas de treinamento, culminando na produção de um artigo científico curto com dados experimentais coletados durante a disciplina.				
EMENTA				
Estudo do comportamento de <i>Apis mellifera</i> africanizadas e abelhas sem ferrão, principalmente as que ocorrem em nossa região, integrando biologia, genética, ecologia, sanidade e produção animal. Aborda forrageamento, defensividade, enxameação, reprodução, termorregulação, higiene, comunicação e organização social, incluindo estressores bióticos e abióticos e fenômenos como Colony Collapse Disorder (CCD). Ênfase em técnicas de observação, registro e análise comportamental, com metodologias padronizadas baseadas no COLOSS BEEBOOK e protocolos internacionais. Aplicações práticas incluem manejo, produção de rainhas, melhoramento genético e conservação de colônias, destacando a relevância ecológica e produtiva no semiárido. Combina teoria e prática em campo e laboratório, preparando os alunos para interpretar e aplicar dados comportamentais em programas de manejo, conservação e sanidade.				

CONTEÚDO PROGRAMÁTICO





UNIDADE I – Biologia, Genética e Comportamento Básico (20h)		T	P	T-P
1	Introdução à biologia e genética de abelhas africanizadas (<i>Apis mellifera</i> L.) e abelhas sem ferrão	5	0	0
2.	Preparação de colmeia de observação, captura e marcação de abelhas (protocolos de von Frisch)	0	5	0
3.	Construção de entogramas e registro sistemático do repertório comportamental	0	5	0
4.	Discussão sobre implicações ecológicas e produtivas	5	0	0
	<i>Subtotal</i>	10	10	20
UNIDADE II: Comportamentos Específicos e Aplicações Práticas 25h		T	P	T-P
1.	Comportamento forrageiro e polinização: análise de frequência e duração	3	3	0
2.	Comportamento defensivo e enxameatório	3	3	0
3.	Produção e introdução de rainhas de <i>Apis mellifera</i> (Africanizadas)	3	3	0
4.	Comportamento reprodutivo: acasalamento natural e inseminação instrumental	3	3	0
5.	Comportamento higiênico e sua relação com sanidade e estressores (Varroa, Nosema, vírus)	3	2	0
	<i>Subtotal</i>	15	20	0
UNIDADE III : Aprendizagem, Memória e Tecnologias Aplicadas: 15h		T	P	T-P
1.	Ensaios de aprendizagem e memória: condicionamento clássico e diferencial	3	1	4
2.	Observação da dança das abelhas (“waggle dance”) e análise de precisão	1	3	4
3.	Uso de tecnologias de monitoramento (RFID, rastreamento de voo, sensores de temperatura)	4	0	4
4.	Stressores abióticos no semiárido: temperatura, umidade, disponibilidade floral e CCD	2	1	3
		10	06	15

MÉTODOS		
TÉCNICAS	RECURSOS DIDATICOS	INSTRUMENTOS DE AVALIAÇÃO
• Observação e análise comportamental de abelhas • Registro sistemático e análise de dados • Treinamento e manejo de colônias • Aplicação de metodologias padronizadas do <i>COLOSS BEEBOOK</i> • Aulas expositivas e exposição dialogada com recursos audiovisuais	• Aulas expositivas • Seminários com artigos científicos • Práticas em campo e laboratório • Aulas remotas por meio de plataformas de videoconferência (ex.: Google Meet) • Palestras e colaborações	• Seminários com artigos científicos: 40% • Projeto de pesquisa: 40% • Relatório de aulas práticas: 20% • Participação e frequência: 20%



• Seminários temáticos baseados em artigos científicos • Oficinas de escrita científica, elaboração de projetos e protocolos • Revisão sistemática da literatura e levantamento da literatura cinzenta • Discussões orientadas por perguntas-chave • Estudos de caso em apicultura e meliponicultura • Práticas de campo e laboratório: produção de rainhas, coleta e análise de pólen, sanidade, entre outras • Estudos dirigidos e análises guiadas • Participação de pesquisadores convidados, nacionais e internacionais, em temas específicos	• com profissionais convidados de outras instituições e do exterior • Projetor multimídia, computador e internet; • Equipamentos e materiais apícolas (colmeias, caixas de rainhas, seringas de transferência etc.); • Laboratórios da UFERSA e a Fazenda Experimental Rafael Fernandes, apiários e meliponários; • Plataformas digitais para pesquisa científica (CAPES, Web of Science, Google Scholar) • Fichas analíticas padronizadas e protocolos de prática; • Textos científicos atualizados fornecidos pelos docentes • Microscópios e lâminas para análises laboratoriais	
--	---	--



REFERÊNCIAS BIBLIOGRÁFICAS (ABNT 2000)

BIBLIOGRAFIA BÁSICA

- DIETEMANN, V.; ELLIS, J. D.; NEUMANN, P. (eds.). *The COLOSS BEEBOOK: Standard Methods for *Apis mellifera* Research*. **Journal of Apicultural Research**, 2013–2019. Disponível em: <https://coloss.org/beebook/>.
- OLIVEIRA-JÚNIOR, J. M. B.; SANTOS, L. (org.). **Ecologia e Comportamento de Abelhas**. Ponta Grossa: Atena Editora, 2021.
- PAGE JR., R. E. *The Spirit of the Hive: The Mechanisms of Social Evolution*. Cambridge: Harvard University Press, 2013.
- SEELEY, T. D. **Honeybee Ecology: A Study of Adaptation in Social Life**. Princeton: Princeton University Press, 1985.
- SEELEY, T. D. *The Lives of Bees: The Untold Story of the Honey Bee in the Wild*. Princeton: Princeton University Press, 2019.
- TAUTZ, J. *O Fenômeno das Abelhas*. Ribeirão Preto: FUNPEC, 2010.
- WINSTON, M. L. **A Biologia da Abelha**. Tradução de Silvana C. dos Santos. Jaboticabal: FUNEP, 2003.
- XAVIER, A. J. et al. **Jandairas: passado, presente e futuro**. Mossoró: EDUFERSA, 2021.

BIBLIOGRAFIA COMPLEMENTAR

- FONSECA, V. L. I. et al. (org.). **Polinizadores no Brasil: contribuição e perspectivas para a biodiversidade, uso sustentável, conservação e serviços ambientais**. São Paulo: EDUSP, 2012. 488 p.
- FONSECA, V. L. I.; SARAIVA, A. M.; JONG, D. DE. (eds.). **Bees as pollinators in Brazil**. Ribeirão Preto: Holos, 2006. 112 p.
- FREITAS, B. M.; PINHEIRO, J. N. **Polinizadores e pesticidas: princípios e manejo para os agroecossistemas brasileiros**. Brasília: MMA, 2012. 112 p.
- LANDIM, C. C. **Abelhas: morfologia e função de sistemas**. São Paulo: UNESP, 2009. 408 p.
- PAGE JR., R. E. **The spirit of the hive: the mechanisms of social evolution**. Cambridge: Harvard University Press, 2013.
- SILVEIRA, F. A.; MELO, A. R.; ALMEIDA, E. A. B. **Abelhas brasileiras: sistemática e identificação**. Belo Horizonte, 2002. 253 p.
- SPIVAK, M.; FLETCHER, D. J. C.; BREED, M. D. (eds.). **The African honey bee**. Boulder: Westview Press, 1991.

Artigos Científicos para discussões, seminários e elaborações de projetos envolvendo temas da disciplina como, Danças e sinais vibracionais, Reconhecimento visual e padrões sociais, Análise de redes sociais de abelhas, outros serão passados na disciplina:

BITTERMAN, M. E. (1988). *Classical Conditioning of Proboscis Extension in Honeybees (*Apis mellifera*)*. *Journal of Comparative Psychology*, 102(4), 358–365.
<https://psycnet.apa.org/doi/10.1037/0735-7036.102.4.358>



- BRAUN, B., & BICKER, G.** (1983). *Habituation of an Appetitive Reflex in the Honeybee*. *Journal of Neurophysiology*. <https://journals.physiology.org/doi/10.1152/jn.1983.49.4.1117>
- CHEN, J., RODRIGUEZ RINCON, J. O., DEGRANDI-HOFFMAN, G., FEWELL, J., HARRISON, J., & KANG, Y.** (2023). *Impacts of seasonality and parasitism on honey bee population dynamics*. *arXiv*. <https://arxiv.org/abs/2306.13300>
- COUVILLON, M. J.** (2021). *Encyclopedia of Social Insects*. Springer, Cham.
- GHAROONI-FARD, G., BYERS, M., DESHMUKH, V., BRADLEY, E., MAYO, C., TOPAZ, C., & PELEG, O.** (2023). *A Computational Topology-based Spatiotemporal Analysis Technique for Honeybee Aggregation*. *arXiv*. <https://arxiv.org/abs/2307.09720>
- HEJAZI, B., KÜCHLER, C., BAGHERI, G., & BODENSCHATZ, E.** (2022). *Flight of a honeybee in turbulent wind*. *arXiv*. <https://arxiv.org/abs/2209.05977>
- KIRCHNER, W. H.** (1993). *Vibrational signals in the tremble dance of the honeybee, Apis mellifera*. *Behavioral Ecology and Sociobiology*, 33(3), 169–172.
<https://link.springer.com/article/10.1007/BF00172151>
- MEDINA-FLORES, C. A. et al.** Effect of hygienic behavior on resistance to chalkbrood disease (*Ascosphaera apis*) in Africanized bee colonies (*Apis mellifera*). **Revista Mexicana de Ciencias Pecuarias**, v. 13, n. 1, p. 225-239, 2022
- LIANG, A.** (2024). *Developing an AI-based Integrated System for Bee Health Evaluation*. *arXiv*. <https://arxiv.org/abs/2401.09988>
- RATNIEKS, F. L. W.** (2001). *Are you being served? Supermarkets and bee hives*. *The Beekeepers Quarterly*, 67, 26–27.
- SANTOS, R. G. et al.** Comparative study of the performance of africanized bees managed in thermal stress and thermal comfort in a semiarid region. **Acta Veterinaria Brasilica**, Mossoró, v. 17, n. 4, p. 1-13, 2023. DOI: 10.21708/avb.2023.17.4.12083
- SLEDEVIČ, T., SERACKIS, A., MATUZEVIČIUS, D., PLONIS, D., & VDOVIK, G.** (2025). *Visual recognition of honeybee behavior patterns at the hive entrance*. *PLOS ONE*, 20(2), e0318401.
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0318401>
- SPIVAK, M.; DANKA, R. G.** Perspectives on hygienic behavior in *Apis mellifera* and other social insects. **Apidologie**, v. 52, p. 1-16, 2021. DOI: <https://doi.org/10.1007/s13592-020-00784-z>

APROVAÇÃO PELO COLEGIADO

DATA	COORDENADOR DO PPGCA

APROVAÇÃO PELO CONSEPE

DATA	COORDENADOR DO PPGCA



CON
TEÚ
DO
PRO
GRA
MÁ
TIC
O

UNIDADE I		T	P	T-P
I	Fundamentos e construção de projetos	2	0	2
	Histórico e sistemática da apicultura e panorama mundial	2	0	2
	Noções de Revisão sistemática e cienciomentria	2	2	4
	Bioclimatologia em abelhas	2	0	2
	Construção de projetos e pesquisas e protocolos	2	0	2
	Flora apícola e calendário apícola	1	3	4
UNIDADE II		T	P	T-P
II	Divisão de colônias por diferentes métodos	1	3	4
	Produção de rainhas	1	3	4
	Sanidade apícola: um só saúde	2	0	2
	Nutrição e alimentação na entressafra	2	2	4
	Produtos apícolas: composição, colheita e aproveitamento	2	2	4
	Legislação aplicada à apicultura e meliponicultura	2	0	2
UNIDADE III		T	P	T-P
III	Pesquisa bibliográfica e trabalhos avaliativos	0	4	4
	Melhoramento genético	2	0	2
	Associativismo, cooperativismo e empreendedorismo apícola	2	0	2
	Coleta de dados de campo	0	4	4
	Atividade de integração prática: dia de campo ou visita técnica	0	4	4
	Produção de protocolos práticos	2	2	4
	Apresentação de projetos	0	4	4

MÉTODOS		
TÉCNICAS	RECURSOS DIDATICOS	INSTRUMENTOS DE AVALIAÇÃO
- Exposição dialogada com uso de recursos audiovisuais - Seminários temáticos com base em artigos	- Projetor multimídia, computador e internet; - Equipamentos e materiais apícolas (colmeias, caixas de rainhas, seringas de transferência	- A avaliação será composta por atividades com pesos específicos, totalizando 10,0 pontos. Os critérios consideram o desempenho



científicos - Aulas práticas de campo e laboratório (produção de rainhas, coleta e análise de pólen, sanidade dentre outras) - Revisão sistemática e levantamento da literatura cinzenta - Oficinas de escrita científica e de construção de projetos e protocolos - Discussões orientadas por perguntas-chave - Estudo de caso com apicultura e meliponicultura - Pesquisadores convidados nacionais e internacionais para fazerem aulas em temas específicos	etc.); - Laboratórios da UFERSA e a Fazenda Experimental Rafael Fernandes, apiários e meliponários; - Plataformas digitais para pesquisa científica (CAPES, Web of Science, Google Scholar) - Fichas analíticas padronizadas e protocolos de prática; - Textos científicos atualizados fornecidos pelos docentes - Microscópios e lâminas para análises laboratoriais	individual e coletivo nas tarefas práticas, teóricas e científicas. - Elaboração de projeto de iniciação científica completo com orçamento (alinhado à dissertação ou tese) e apresentação oral; - Relatório técnico da aulas práticas - Projeto de qualificação de mestrado (quando aplicável) - Revisão sistemática e levantamento da literatura cinzenta (para doutorandos) - Fichas analíticas de leitura de artigos e protocolos de práticas - Participação nas discussões e atividades práticas
---	--	--

REFERÊNCIAS BIBLIOGRÁFICAS (ABNT 2000)

BIBLIOGRAFIA BÁSICA

- CRANE, E. The World History of Beekeeping and Honey Hunting. Duckworth. 2000.
- GOODMAN, L. Form and Function in the Honey Bee. IBRA. 2003.
- GRAHAM, J.M. (Ed). The hive and the honeybee. Ext. Rev. Hamilton, Illinois: Dadant & Sons Inc., 1992.
- HEPBURN, H. R.; RADLOFF, S. E. Honeybees in Africa. 1. Springer. 1998.
- LAIDLAW JR., H.H. Criação Contemporânea de Rainhas(trad.). Edit. La Salle, Canoas-RS. 1998.
- LANGSTHOTH, L.L. A Practical Treatise on the Hive and Honey-Bee. 1860. Third Ed. Revised, and Illustrated with Seventy-seven Engraving. Northern Bee Books. 2014
- MICHENER, C.D. The Social Behavior of the Bees. Harvard University Press. 1974.
- MORITZ, R.F.A.; SOUTHWICK, E.E. Bees as Superorganisms. An Evolutionary Reality. Springer-Verlag. 1992.
- PAGE, JR., R. The Spirit of the Hive: The Mechanism of Social Evolution. Harvard University Press. 2013.
- RINDERER, R.E. Bee Genetics and Breeding. Academic Press, Inc. 1986.
- SEELEY, T.D. The Wisdom of the Hive. The Social Physiology of



Honey Bee Colonies. Harvard University Press.

- SEELEY, T. D. Ecologia da Abelha: Um estudo de adaptação na vida social. Paixão Editores. 2006.
- SEELEY, T. D. Honey Bee Democracy. Princeton University Press. 2010.
- SPIVAK, M.; FLETCHER, D.J.C.; BREED, M.C. The African Honey Bee. Westview Studies in Insect Biology. 1. Westview Press. 1991.
- TAUTZ, J. O Fenômeno das Abelhas. Artmed. 2010.
- Von FRISCH, K. The dance Language and Orientation of Bees. Transl. By L. E. Chadwick with a new foreword by T.D. Seeley. Harvard University Press. 1993
- WISNTON M. K. Biologia de Abelhas (trad). Magister, PA. 2003

BIBLIOGRAFIA COMPLEMENTAR

- Anais dos Encontros sobre Abelhas de Ribeirão Preto-
<https://rge.fmrp.usp.br/encontro/anais.php>
- FONSECA, V.L.I.; CANHOS, D.A.L; ALVES, D.A.; SARAIVA, A.M.(Org.) POLINIZADORES NO BRASIL: Contribuição e Perspectivas para a Biodiversidade, Uso Sustentável, Conservação e Serviços Ambientais. EDUSP, 2012.
- FONSECA, V.L.I.; SARAIVA, A.M.; DE JONG, D.(Ed). Bees as Pollinators in Brazil. Editora Holos, 2006.
- FREITAS, B.M.; PINHEIRO, J.N. Polinizadores e Pesticidas: Princípios e Manejo para os Agroecossistemas Brasileiros. Brasília: MMA, 2012.
<https://www.semabelhasemalimento.com.br/wp-content/uploads/2015/02/polinizadores-e-pesticidas-final-.pdf>
- LANDIM, C.C. Abelhas: morfologia e função de sistemas. Editora UNESP, 2009.
- SILVEIRA, F.A.; MELO, A.R.; ALMEIDA, E.A.B. Abelhas Brasileiras: sistemática e identificação. Belo Horizonte, 2002.
https://www.researchgate.net/publication/280112185_Abelhas_Brasileiras_Sistematica_e_Identificacao

PERIÓDICOS:

- **Journal of Apicultural Research:**
<https://www.ibrabee.org.uk/journal-of-apicultural-research>
- **Apidologie:** <https://link.springer.com/journal/13592>
- **Journal of Economic Entomology:**
<https://academic.oup.com/jee>
- **Science:** <https://www.science.org/journal/science>
- **Nature:** <https://www.nature.com/>
- **PLOS ONE:** <https://journals.plos.org/plosone/>

APROVAÇÃO PELO COLEGIADO



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

DATA	COORDENADOR DO PPGCA
APROVAÇÃO PELO CONSEPE	
DATA	COORDENADOR DO PPGCA



UFERSA | **20**
ANOS



Universidade Federal Rural do Semi-Árido (UFERSA)
Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG)
6ª Reunião Ordinária de 2026 do CPPGIT

2º PONTO

Apreciação e deliberação sobre o projeto de pesquisa "PI00001-2026 - Percepção de risco ambiental e atitudes frente ao rejeito salino da dessalinização no semiárido brasileiro", proposto pelo professor visitante Heitor de Oliveira Braga.



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – UFERSA
CENTRO DE CIÊNCIAS AGRÁRIAS
PROGRAMA DE DOUTORADO EM DESENVOLVIMENTO E MEIO AMBIENTE
ASSOCIAÇÃO PLENA EM REDE

PESQUISADOR RESPONSÁVEL:

PROF. DR. HEITOR DE OLIVEIRA BRAGA

NATUREZA DA PESQUISA:

PROJETO EXECUTADO POR PROFESSOR
VISITANTE DA INSTITUIÇÃO

**Percepção de risco ambiental e atitudes frente ao rejeito salino da
dessalinização no semiárido brasileiro**

**Environmental Risk Perception and Attitudes Toward Desalination Brine in
the Brazilian Semiarid Region**

MOSSORÓ, RN
JULHO, 2026

SUMÁRIO

RESUMO	3
ABSTRACT.....	5
1. INTRODUÇÃO	5
2. OBJETIVOS.....	7
3. HIPÓTESES.....	8
4. METODOLOGIA.....	9
5. CRONOGRAMA	14
6. ORÇAMENTO.....	15
7. RESULTADOS ESPERADOS DO DESENVOLVIMENTO DO PROJETO.....	17
8. PARTICIPANTES DO PROJETO	18
9. REFERÊNCIAS	19
10. ANEXOS.....	22

RESUMO

A dessalinização de águas subterrâneas salobras tem se consolidado como uma estratégia relevante para garantir o acesso à água potável em comunidades rurais do semiárido brasileiro, onde a escassez hídrica crônica e a variabilidade climática intensificam a dependência de fontes alternativas de abastecimento. Entretanto, esses sistemas geram rejeito salino que, quando descartado inadequadamente, pode provocar salinização do solo, degradação da vegetação, impactos sobre a fauna e riscos cumulativos aos sistemas produtivos familiares. Embora os avanços científicos sobre tecnologias de dessalinização e manejo de salmouras tenham crescido nas últimas décadas, ainda são escassos os estudos que investigam como comunidades rurais percebem esses riscos ambientais e desenvolvem estratégias locais para lidar com esse rejeito. Este estudo tem como objetivo analisar como os conhecimentos ecológicos tradicionais (CET), as percepções de manejo ambiental e as atitudes de agricultores familiares influenciam a avaliação dos riscos ambientais e a confiança no uso controlado do rejeito salino em comunidades rurais de Mossoró, Rio Grande do Norte. A pesquisa adotará uma abordagem socioecológica de métodos mistos, combinando entrevistas semiestruturadas com análises quantitativas e qualitativas. Serão avaliadas três dimensões analíticas: Percepção de Risco Ambiental (PRA), Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) e Atitudes e Confiança no Uso do Rejeito Salino (ACUR). As análises incluirão estatística descritiva, testes não paramétricos, modelos exploratórios de regressão e análise temática das respostas abertas. Espera-se que os resultados contribuam para o avanço das discussões sobre gestão sustentável de rejeitos da dessalinização, incorporando saberes locais aos debates sobre governança da água e subsidiando estratégias socialmente contextualizadas de adaptação climática, resiliência rural e uso sustentável da água em regiões semiáridas.

Palavras-chave: conhecimentos ecológicos tradicionais; rejeito salino; percepção de risco ambiental; comunidades rurais; sistemas socioecológicos; semiárido brasileiro.

ABSTRACT

Brackish groundwater desalination has become an important strategy to ensure drinking water access in rural communities of the Brazilian semiarid region, where chronic water scarcity and climate variability intensify dependence on alternative water sources. However, desalination systems generate saline brine, whose inadequate disposal may contribute to soil salinization, vegetation degradation, impacts on fauna, and long-term risks to small-scale farming systems. Despite growing research on desalination technologies and brine management, limited attention has been given to how rural communities perceive these environmental risks and develop local strategies to manage them. This study investigates how traditional ecological knowledge (TEK), environmental management perceptions, and farmers' attitudes influence environmental risk assessment and trust in the controlled use of desalination brine in family farming communities in Mossoró, Rio Grande do Norte, Brazil. A socioecological mixed-methods approach will be adopted, combining semi-structured interviews with quantitative and qualitative analyses. Three analytical dimensions will be assessed: Environmental Risk Perception (ERP), Traditional Ecological Knowledge and Environmental Management Perception (TEK-EMP), and Attitudes and Confidence in Brine Use (ACBU). Data analysis will include descriptive statistics, non-parametric tests, exploratory regression models, and thematic analysis of qualitative responses. The study is expected to advance discussions on sustainable brine management by integrating local ecological knowledge into water governance debates and supporting socially grounded strategies for climate adaptation, rural resilience, and sustainable water use in semiarid regions.

Keywords: Traditional ecological knowledge; desalination brine; environmental risk perception; rural communities; socioecological systems; semi-arid regions.

1. INTRODUÇÃO

A intensificação da escassez hídrica decorre da combinação entre mudanças climáticas, crescimento da demanda e uso insustentável de águas superficiais e subterrâneas, ampliando os “water gaps” em múltiplas bacias e projetando aumento do uso insustentável da água em cenários de aquecimento (Boretti; Rosa, 2019; Wang et al., 2024). Projeções recentes indicam que a população urbana exposta à escassez e à degradação da qualidade da água poderá alcançar quase metade dos habitantes das grandes cidades em 2050, exigindo portfólios de soluções que combinem gestão da demanda, proteção de ecossistemas e novas fontes de oferta (He et al., 2021; Liu et al., 2024).

Diante disso, a dessalinização, baseada sobretudo em tecnologias de membranas, consolida-se como componente central de estratégias de segurança hídrica em regiões áridas e semiáridas, ao permitir ampliar a oferta a partir de águas salobras, salinas e residuárias (Ingrao et al., 2023; Ricart et al., 2021). A rápida expansão de plantas de dessalinização, inclusive voltadas à exploração de aquíferos salobros e sistemas costeiros, está associada a relevantes trade-offs ambientais e energéticos, entre eles o volume crescente de rejeito salino, o consumo intensivo de energia e os custos operacionais de sistemas de osmose reversa e rotas de tratamento avançadas (Panagopoulos; Haralambous, 2020; Soliman et al., 2021).

Revisões recentes destacam que a dessalinização produz volumes substanciais de salmoura hipersalina contendo temperaturas elevadas e uma gama de aditivos químicos, cuja disposição inadequada representa um dos principais fatores de impacto ecotoxicológico desse setor (Panagopoulos; Haralambous, 2020; Omerspahic et al., 2022). Evidências de ambientes costeiros indicam que a pluma de salmoura pode alterar a química da água, comprometer fluxos biogeoquímicos em sedimentos e afetar comunidades bênticas de bactérias, macroalgas, invertebrados e corais, com efeitos que podem se estender por quilômetros a partir do ponto de descarga (Sirota et al., 2024; Sola et al., 2025).

Embora grande parte da literatura se concentre em ecossistemas marinhos, estudos de escopo global apontam que o aumento da salinidade de solos e águas constitui risco emergente para a produtividade agrícola e a segurança alimentar em regiões áridas e semiáridas, degradando propriedades físico-químicas e biológicas do solo (Cuevas et al., 2019; El-Ramady et al., 2024). Todavia, faltam sínteses específicas sobre os impactos do rejeito da dessalinização em solos continentais, vegetação nativa, fauna silvestre e sistemas produtivos familiares; trata-se de lacuna bibliográfica, pois a maior parte dos estudos aborda ambientes costeiros ou avaliações tecnológicas de dessalinização e gestão de salmouras, sem detalhar efeitos em agroecossistemas de pequena escala (Panagopoulos; Haralambous, 2020).

No semiárido brasileiro, caracterizado por chuvas irregulares, déficit hídrico anual, solos rasos sobre embasamento cristalino e águas subterrâneas frequentemente salinas, a vulnerabilidade hídrica intensifica a dependência de soluções não convencionais, como a dessalinização de águas subterrâneas salobras (Cavalcante Júnior et al., 2019). Políticas públicas federais fomentaram a disseminação de sistemas de osmose reversa alimentados por energia solar em comunidades rurais, integrando, em alguns casos, o aproveitamento produtivo do concentrado em piscicultura e agricultura familiar, ainda que a aptidão agrícola do rejeito seja frequentemente limitada em função de sua elevada salinidade (Cavalcante Júnior et al., 2019; Amaral; Navoni, 2022).

No município de Mossoró, Estado do Rio Grande do Norte, inserido integralmente no semiárido brasileiro, comunidades rurais de agricultura familiar convivem cotidianamente com os impactos e desafios associados ao uso de dessalinizadores de águas subterrâneas, incluindo a necessidade de manejo rotineiro do rejeito salino (Cavalcante Júnior et al., 2019; Amaral; Navoni, 2022). Nessas localidades, diferentes formas de descarte, reutilização e manejo do rejeito são desenvolvidas com base em limitações estruturais, condições ambientais locais e necessidades produtivas específicas, revelando dinâmicas socioambientais complexas ainda pouco documentadas pela literatura científica nacional e internacional (Amaral; Navoni, 2022; Ingrao et al., 2023).

Embora existam avanços nas discussões sobre dessalinização, impactos da salmoura e estratégias de minimização e reaproveitamento de rejeitos, ainda são escassos estudos que integrem impactos ambientais, conhecimento ecológico tradicional (CET) e percepção de risco em comunidades rurais semiáridas (Bello et al., 2021; Ricart et al., 2021). Permanecem limitadas as evidências sobre como agricultores familiares interpretam empiricamente alterações no solo, vegetação, fauna e sistemas produtivos decorrentes do rejeito salino, bem como sobre como essas percepções informam práticas de manejo e participação em processos decisórios (Amaral; Navoni, 2022; Villar-Navascués; Fragkou, 2021).

Nesta perspectiva, este estudo tem como objetivo analisar como os conhecimentos ecológicos tradicionais (CET), as percepções de manejo ambiental e as atitudes de agricultores familiares influenciam a avaliação dos riscos ambientais e a confiança no uso controlado do rejeito salino da dessalinização em comunidades rurais de Mossoró, Rio Grande do Norte, Nordeste do Brasil, contribuindo para estratégias de gestão mais sustentáveis e socialmente contextualizadas no semiárido brasileiro (Cavalcante Júnior et al., 2019; Amaral; Navoni, 2022).

2. OBJETIVOS

2.1 *Objetivo Geral:*

- Explorar e compreender os Conhecimentos Ecológicos Tradicionais e as Percepções de Manejo Ambiental (CET-PMA) dos agricultores familiares acerca dos efeitos ambientais do rejeito salino da dessalinização, com base em observações locais, interpretações ecológicas e critérios empíricos de avaliação ambiental.

2.2 *Objetivos Específicos:*

- Sistematizar e dar visibilidade aos Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) dos agricultores familiares sobre os efeitos do rejeito salino no solo, na vegetação, na fauna e nos sistemas produtivos, contribuindo para a compreensão socioecológica local desse recurso;

- Analisar a Percepção de Risco Ambiental associada ao rejeito salino (PRA) dos agricultores familiares, considerando diferentes contextos de uso, descarte e variabilidade climática;

- Avaliar as Percepções de Manejo Ambiental, operacionalizadas no índice CET-PMA, incluindo sinais ecológicos observados, critérios locais de avaliação ambiental e formas tradicionais de lidar com os impactos do rejeito salino;

- Examinar as Atitudes e a Confiança no Uso do Rejeito Salino (ACUR) nos sistemas produtivos familiares, considerando a experiência prática, a organização social comunitária e o acesso à orientação técnica;

- Investigar a influência de características sociodemográficas, produtivas e de inserção social sobre os índices CET-PMA, PRA e ACUR, analisando seus efeitos sobre o uso e a gestão ambiental do rejeito salino.

3. HIPÓTESES

3.1 Hipótese Geral:

Os níveis de Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) estão positivamente associados à Percepção de Risco Ambiental associada ao rejeito salino (PRA), enquanto as Atitudes e a Confiança no Uso do Rejeito Salino (ACUR) são influenciadas por fatores sociodemográficos, produtivos e de inserção social dos agricultores familiares do semiárido brasileiro.

3.2 Hipóteses Específicas:

3.2.1 Hipótese 1

Maiores níveis de Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) estão positivamente associados a maiores níveis de Percepção de Risco Ambiental associada ao rejeito salino (PRA) entre os agricultores familiares

3.2.2 Hipótese 2

Os níveis de CET-PMA variam significativamente em função da experiência produtiva e do tempo de exposição ao rejeito salino, de modo que agricultores com maior tempo de prática apresentam valores mais elevados desse índice.

3.2.3 Hipótese 3

As Atitudes e a Confiança no Uso do Rejeito Salino (ACUR) estão negativamente associadas à Percepção de Risco Ambiental associada ao rejeito salino (PRA) e positivamente associadas ao acesso à orientação técnica e à participação em organizações comunitárias.

3.2.4 Hipótese 4

Variáveis sociodemográficas e produtivas, tais como escolaridade, renda familiar, tipo de atividade produtiva e participação social, explicam variações significativas nos índices CET-PMA, PRA e ACUR.

4. METODOLOGIA

4.1. Área de estudo

O estudo será desenvolvido no município de Mossoró, localizado no estado do Rio Grande do Norte, na região Nordeste do Brasil, inserido integralmente no domínio do semiárido brasileiro. O município apresenta clima semiárido quente, com precipitações irregulares, elevada evapotranspiração e recorrentes períodos de estiagem, condições que intensificam a dependência de fontes alternativas de abastecimento hídrico, como a dessalinização de águas subterrâneas salobras.

A área de estudo compreende comunidades rurais de agricultura familiar que utilizam sistemas de dessalinização para o abastecimento de água potável, incluindo as comunidades de Boa Fé, Puxa Boi, Oziel Alves, São Romão, Córrego Mossoró I, Córrego Mossoró II, Santa Elza, Fazenda Nova, São José I, São José II e São Cristóvão. A escolha dessas comunidades baseia-se na presença de dessalinizadores em operação, na relevância da agricultura familiar para a subsistência local e na existência de práticas diversas de destinação e uso do rejeito salino, o que permite analisar diferentes contextos socioambientais associados à gestão desse recurso.

4.2. Coleta de dados

Antes da efetivação da coleta de dados, serão realizadas visitas sucessivas às comunidades rurais selecionadas e aos principais pontos de encontro dos agricultores familiares, com o objetivo de estabelecer um ambiente de confiança com as comunidades locais. Essa etapa inicial visa minimizar possíveis vieses na coleta de dados e promover uma colaboração mais efetiva com os participantes da pesquisa, conforme recomendado por Brook e McLachlan (2008).

As entrevistas ocorrerão imediatamente após a aprovação do projeto pelo Comitê de Ética em Pesquisa (CEP) da Universidade do Estado do Rio Grande do Norte (UERN). A coleta de dados está prevista para o período de 15/07/2026 a 15/10/2026, ocasião em que serão realizadas as visitas às comunidades selecionadas e a aplicação dos questionários semiestruturados junto aos agricultores familiares participantes do estudo.

A recolha das informações junto à população-alvo será iniciada por meio de visitas aos locais de convivência comunitária e às residências dos agricultores familiares em cada localidade, acompanhadas da apresentação e leitura do Termo de Consentimento Livre e Esclarecido (TCLE) (Anexo 1), em conformidade com as diretrizes estabelecidas pelo Comitê de Ética. O TCLE esclarecerá detalhadamente os objetivos da pesquisa, os procedimentos metodológicos adotados, os possíveis riscos envolvidos e a natureza voluntária da

participação, destacando que os participantes poderão desistir da pesquisa a qualquer momento, sem qualquer prejuízo ou penalidade.

Após a leitura do TCLE, o agricultor será convidado a manifestar seu interesse em participar do estudo. Em caso de aceite, o termo será assinado em duas vias, uma pelo participante e outra pelo pesquisador responsável. Para participantes não alfabetizados, será utilizada a impressão datiloscópica como forma de consentimento.

Os riscos envolvidos neste estudo são considerados mínimos, estando relacionados principalmente à exposição de opiniões, percepções e práticas associadas ao uso, à destinação e aos possíveis impactos ambientais do rejeito salino da dessalinização. Outros riscos menos evidentes incluem eventual desconforto, timidez ou constrangimento durante as respostas. Para mitigar esses riscos, será garantido o anonimato e a privacidade dos participantes, por meio da utilização de códigos de identificação nos questionários e da manutenção do sigilo sobre dados pessoais. O acesso às informações coletadas será restrito exclusivamente ao pesquisador responsável, que será o único autorizado a utilizar, armazenar e analisar os dados.

No que se refere ao uso de imagens ou gravações de áudio, será explicado aos participantes que esses materiais poderão ser utilizados exclusivamente para fins científicos, caso haja consentimento explícito. O uso de gravações será totalmente opcional e somente ocorrerá mediante autorização expressa do participante. O estudo será conduzido exclusivamente após a aprovação pelo Comitê de Ética em Pesquisa (CEP) da Universidade do Estado do Rio Grande do Norte (UERN), Mossoró, Brasil, e seguirá rigorosamente o Código de Ética da Sociedade Internacional de Etnobiologia (ISE), disponível em <https://ethnobiology.org/about-society-ethnobiology/ethics>.

As entrevistas serão realizadas apenas após o consentimento informado e a explicação clara dos objetivos da pesquisa, respeitando o direito do participante de interromper a entrevista a qualquer momento, garantindo o sigilo e a confidencialidade das informações. Durante as entrevistas, o pesquisador poderá utilizar anotações e, quando autorizado, gravadores eletrônicos para o registro das informações, sempre em conformidade com as melhores práticas científicas (Albuquerque et al., 2014).

Para manter elevados padrões éticos e científicos durante as atividades de campo, a pesquisa seguirá as principais diretrizes metodológicas da etnobiologia e da etnoecologia (Albuquerque et al., 2014). A coleta de dados será baseada na aplicação de questionários semiestruturados (Huntington, 2000) aos agricultores familiares das comunidades amostradas (Anexo 2). A seleção dos participantes ocorrerá por meio da técnica de bola de neve (Bailey, 2008) e/ou por amostragem oportuna, sempre que houver oportunidade para a realização

imediate da entrevista ou para o agendamento posterior (Braga et al., 2022). Serão selecionados apenas agricultores com ligação direta com o uso da água dessalinizada e/ou com a destinação do rejeito salino.

O questionário será cuidadosamente validado por meio de revisões bibliográficas aprofundadas e da realização de pré-testes nos dois primeiros meses do projeto. Essa etapa tem como objetivo fortalecer a qualidade e a precisão da coleta de dados, assegurando a consistência e a validade das informações obtidas. Os questionários abordarão tópicos relacionados ao perfil sociodemográfico e produtivo dos agricultores, aos conhecimentos ecológicos tradicionais (CET), às percepções de manejo ambiental, à percepção de riscos ambientais e às atitudes e à confiança no uso do rejeito salino da dessalinização (Anexo 2).

4.3. Análise dos dados

4.3.1 Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA)

A análise dos dados relativos aos Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) seguirá uma abordagem integradaêmica e ética (Harris, 1976), considerando tanto as interpretações locais dos agricultores familiares quanto o diálogo sistemático com a literatura científica. As informações obtidas por meio dos itens fechados e das questões abertas do questionário serão sistematizadas, codificadas e organizadas de modo a representar o conjunto dos conhecimentos observacionais, das percepções ambientais e dos critérios locais utilizados para avaliar os efeitos do rejeito salino sobre o solo, a vegetação, a fauna e os sistemas produtivos.

As observações ecológicas relatadas pelos participantes, tais como alterações na salinidade do solo, presença de crostas salinas, mudanças na coloração e no desenvolvimento das plantas e efeitos sobre animais, serão organizadas a partir do modelo de união das diversas competências individuais, empregando-se a noção de “informante onisciente” (Hays, 1976), amplamente utilizada em estudos etnoecológicos. O índice CET-PMA será cuidadosamente categorizado e referenciado, permitindo a síntese de informações provenientes de múltiplas fontes de conhecimento tradicional (Newing, 2010).

As descrições locais serão analisadas de forma comparativa com referências científicas sobre salinização de solos, uso de águas salobras e impactos ambientais associados ao rejeito salino, buscando identificar convergências e divergências entre o CET-PMA e o conhecimento técnico-científico.

4.3.2 Percepção de Risco Ambiental associada ao rejeito salino (PRA)

A Percepção de Risco Ambiental associada ao rejeito salino (PRA) será analisada a partir da construção de um índice composto, baseado nos itens do questionário relacionados

aos potenciais impactos do rejeito salino sobre o solo, a vegetação, a fauna, a infiltração da água e os processos de degradação ambiental ao longo do tempo. A percepção de risco será compreendida como a avaliação subjetiva dos agricultores familiares acerca da probabilidade e da gravidade de danos ambientais associados ao uso e à destinação do rejeito salino.

O índice PRA será calculado a partir da média das respostas válidas, respeitando o critério mínimo de completude dos itens, e posteriormente classificado em níveis baixo, médio e alto. As variações na PRA serão analisadas em relação às características sociodemográficas, produtivas e de inserção social dos participantes, bem como em relação aos níveis do CET-PMA, permitindo avaliar associações entre os conhecimentos ecológicos tradicionais, as percepções de manejo ambiental e a percepção de risco ambiental.

4.3.3 Atitude e Confiança no Uso do Rejeito Salino (ACUR)

Nesta etapa, será construído um índice médio de Atitude e Confiança no Uso do Rejeito Salino (ACUR). A atitude será compreendida como a predisposição favorável ou desfavorável do indivíduo em relação ao uso controlado do rejeito salino nos sistemas produtivos, conforme a definição clássica proposta por Eagly e Chaiken (1993).

O índice ACUR será baseado em um conjunto de itens formulados em escala Likert de cinco pontos, adaptada às condições socioculturais das comunidades estudadas. Esse índice buscará compreender a disposição dos agricultores familiares em utilizar o rejeito salino de forma controlada, considerando aspectos como experiência prática, necessidade hídrica, expectativa de benefícios produtivos e acesso à orientação técnica. O ACUR será analisado em associação com os índices PRA e CET-PMA, bem como com variáveis do perfil sociodemográfico e produtivo dos participantes.

4.3.4 Procedimentos estatísticos

Os dados quantitativos serão analisados por meio de estatística descritiva, incluindo frequências absolutas e relativas, medidas de tendência central (médias e medianas) e medidas de dispersão. A consistência interna dos índices PRA, CET-PMA e ACUR será avaliada por meio do alfa de Cronbach e do ômega de McDonald, com o objetivo de verificar a confiabilidade das escalas utilizadas.

Considerando a natureza ordinal dos dados e a distribuição não normal esperada, serão aplicados testes estatísticos não paramétricos, incluindo correlação de Spearman, testes de Mann-Whitney e Kruskal-Wallis, bem como regressões exploratórias, a fim de examinar associações entre os índices construídos e as variáveis sociodemográficas, produtivas e de inserção social dos participantes. Será adotado um nível de significância de 5% ($p < 0,05$). As

análises estatísticas serão realizadas com o auxílio dos softwares Microsoft Office 365 e R Project for Statistical Computing.

As respostas às questões abertas serão analisadas por meio de análise temática, adotando-se uma abordagem integrada emic–etic, com o objetivo de compreender os significados atribuídos localmente ao rejeito salino, às percepções de risco ambiental e às práticas de gestão ambiental, complementando e aprofundando a interpretação dos resultados quantitativos.

5. CRONOGRAMA

ANO I

Nota: Números indicam os meses. Algumas metas poderão ser modificadas de acordo com as demandas.

Atividade	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr
Ajustes após parecer do CEP	X									
Coleta de dados		X	X	X	X					
Organização dos dados			X	X	X					
Análise dos dados					X	X				
Redação da dissertação/artigo						X	X	X		
Revisão final								X	X	
Defesa/Submissão de artigo									X	X

- As atividades poderão sofrer ajustes em função da data de aprovação do CEP.

6. ORÇAMENTO

O projeto de pesquisa proposto caracteriza-se como de baixo custo operacional, sendo que os recursos destinados às despesas de capital e de custeio serão dimensionados de acordo com a disponibilidade financeira do Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA) da Universidade Federal Rural do Semi-Árido (UFERSA), conforme apresentado no Quadro 1. Parte dos custos poderá ser viabilizada mediante recursos institucionais, bolsas ou editais de fomento, quando disponíveis, contribuindo para a execução das atividades de campo, organização e análise dos dados.

Adicionalmente, o projeto poderá ser submetido a editais de fomento de agências de financiamento, como o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), com o objetivo de captar recursos complementares que viabilizem ou ampliem as atividades previstas, bem como a divulgação dos resultados científicos.

6.1 Itens de despesas de capital

- Material de expediente, destinado à execução das entrevistas e à organização das atividades de pesquisa, incluindo impressão de questionários, artigos, relatórios, manuscritos e demais documentos vinculados ao projeto;
- Material de consumo, como água e alimentação, necessários para o desenvolvimento das atividades de campo;
- Equipamento de posicionamento global (GPS) à prova d'água, com sistema de flutuação e base de mapas globais integrada, para apoio às atividades de campo;
- Material bibliográfico, destinado ao embasamento teórico e metodológico da pesquisa;
- Gravador de voz com saída USB integrada, para registro das entrevistas, mediante autorização prévia dos participantes.

6.2 Itens de despesas de custeio

- Diárias de auxílio, destinadas à cobertura de despesas com hospedagem e alimentação da equipe executora durante os trabalhos de campo;
- Transporte, para deslocamento dos integrantes da equipe até as áreas de coleta de dados;
- Serviços de terceiros – pessoa física, incluindo a contratação de serviços especializados, como revisão de língua inglesa de artigos científicos produzidos no âmbito do projeto;

- Taxas de processamento de artigos, quando aplicáveis, para viabilizar a publicação dos resultados em periódicos científicos.

Quadro 1 - Orçamento detalhado dos custos para a realização da presente pesquisa.

Atividades	Quantidade	Valor unitário	Valor Total
Seleção das áreas de estudo			
Custeio			
Combustível (litros)	1.000	R\$ 6,00	R\$ 6.000,00
Diária para auxílio (hospedagem, refeição etc.)	20	R\$ 350,00	R\$ 7.000,00
Material de consumo			
Custeio			
Material de Expediente para as entrevistas e tabuleiro (pasta, caneta esferográfica, papel, cartolina etc.)	1	R\$ 100,00	R\$ 100,00
Material de consumo (água, refeição etc.)	1	R\$ 500,00	R\$ 500,00
Gravador de Voz	1	R\$ 323,00	R\$ 323,00
GPS*	1	R\$ 0,00	R\$ 0,00
Levantamento de dados			
Combustível (litros)	1.000	R\$ 6,00	R\$ 6.000,00
Total			R\$ 19.923,00

* itens que não possuem custos pois já foram adquiridos.

7. RESULTADOS ESPERADOS DO DESENVOLVIMENTO DO PROJETO

Espera-se que o desenvolvimento do projeto contribua para ampliar a compreensão sobre a Percepção de Risco Ambiental associada ao rejeito salino (PRA) dos agricultores familiares, considerando os efeitos do rejeito salino da dessalinização sobre o solo, a vegetação, a fauna e os sistemas produtivos locais. Pretende-se, ainda, sistematizar e dar visibilidade aos Conhecimentos Ecológicos Tradicionais e Percepções de Manejo Ambiental (CET-PMA) relacionados ao uso e à destinação do rejeito salino, evidenciando os critérios locais empregados na avaliação dos impactos ambientais.

O estudo deverá, também, permitir a identificação de fatores sociais, produtivos e institucionais que influenciam as Atitudes e a Confiança no Uso do Rejeito Salino (ACUR) pelos agricultores familiares, contribuindo para a compreensão das possibilidades e dos limites de sua incorporação nos sistemas produtivos familiares. A análise dos padrões de associação entre Percepção de Risco Ambiental associada ao rejeito salino (PRA), CET-PMA e ACUR possibilitará compreender de que forma o conhecimento tradicional e a experiência prática moldam as estratégias locais de gestão ambiental.

Adicionalmente, espera-se evidenciar potenciais benefícios socioambientais decorrentes do manejo adequado do rejeito salino, incluindo a mitigação de impactos ambientais, o fortalecimento da resiliência produtiva e a melhoria das condições de vida das famílias agricultoras. Por fim, o projeto deverá gerar informações científicas capazes de subsidiar políticas públicas, programas de extensão rural e o desenvolvimento de tecnologias sociais voltadas ao uso sustentável da água e à gestão ambiental no semiárido brasileiro.

8. PARTICIPANTES DO PROJETO

Pesquisador Principal:

Dr. Heitor de Oliveira Braga – Professor Visitante no Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA) da Universidade Federal Rural do Semi-Árido (UFERSA). Doutor em Biociências (Ecologia Marinha) pela Universidade de Coimbra, com Pós-doutorado em Biologia pela Universidade de Aveiro (Portugal). Atua nas áreas de Ecologia Humana, Etnobiologia e Etnoecologia, Educação ambiental, com experiência em conservação da biodiversidade e relações sociedade–natureza em contextos socioambientais.

Pesquisador auxiliar:

Dr. Nildo da Silva Dias – Professor Titular da Universidade Federal Rural do Semi-Árido (UFERSA), Vice-Reitor da instituição e bolsista de Produtividade em Pesquisa do CNPq (Nível 1A). Doutor em Agronomia pela USP/ESALq, com Pós-doutorado pela University of California Riverside (US Salinity Laboratory). Atua em manejo da salinidade, qualidade da água para irrigação e tecnologias sociais de convivência com o semiárido, com ampla experiência em pesquisa aplicada e inovação.

Pesquisador Auxiliar:

Dr. Eric Mateus Soares Dias

Professor temporário do curso Bacharel em Gestão Ambiental na Universidade Estadual do Rio Grande do Norte. Doutor em Estudos Urbanos e Regionais pela Universidade Federal do Rio Grande do Norte. É pesquisador no Laboratório Interdisciplinar Sociedades, Ambientes e Territórios (LISAT/UFRN). Desenvolve pesquisas nas áreas de políticas públicas socioambientais, recursos hídricos e adaptação climática.

9. REFERÊNCIAS

- ALBUQUERQUE, U. P.; CUNHA, L. V. F. C.; LUCENA, R. F. P.; ALVES, R. R. N. (org.).** *Methods and techniques in ethnobiology and ethnoecology*. New York: Springer, 2014.
- AMARAL, K. D. S.; NAVONI, J.** Desalination in rural communities of the Brazilian semi-arid region: potential use of brackish concentrate in local productive activities. *Process Safety and Environmental Protection*, v. 163, p. 305–315, 2022.
- BAILEY, K. D.** *Methods of social research*. 4. ed. New York: The Free Press, 2008.
- BELLO, A. S.; ZOUARI, N.; DA'ANA, D. A.; HAHADAKIS, J.; AL-GHOUTI, M.** An overview of brine management: emerging desalination technologies, life cycle assessment, and metal recovery methodologies. *Journal of Environmental Management*, v. 288, p. 112404, 2021.
- BORETTI, A.; ROSA, L.** Reassessing the projections of the World Water Development Report. *npj Clean Water*, v. 2, p. 15, 2019.
- BRAGA, H. O.; AZEITEIRO, U. M.; SCHIAVETTI, A.; MAGALHÃES, L.** Checking the changes over time and the impacts of COVID-19 on cockle (*Cerastoderma edule*) small-scale fisheries in Ria de Aveiro coastal lagoon, Portugal. *Marine Policy*, v. 135, p. 104843, 2022.
- BROOK, R. K.; MCLACHLAN, S. M.** Trends and prospects for local knowledge in ecological and conservation research and monitoring. *Biodiversity and Conservation*, v. 17, p. 3501–3512, 2008.
- CAVALCANTE JÚNIOR, R. G.; FREITAS, M. A. V.; SILVA, N. F.; AZEVEDO FILHO, F. R.** Sustainable groundwater exploitation aiming at the reduction of water vulnerability in the Brazilian semi-arid region. *Energies*, v. 12, p. 932, 2019.
- CRISPIM, D. L.; PROGÊNIO, M. F.; FERNANDES, L. L.** Proposal for a tool for assessing access to water in rural communities: a case study in the Brazilian semi-arid. *Environmental Management*, v. 70, 2022.
- CRONBACH, L. J.** Coefficient alpha and the internal structure of tests. *Psychometrika*, v. 16, p. 297–334, 1951.
- CUEVAS, J., DALIAKOPOULOS, I. N., DEL MORAL, F., HUESO, J. J., & TSANIS, I. K.** (2019). A review of soil-improving cropping systems for soil salinization. *Agronomy*, 9(6), 295.
- EAGLY, A. H.; CHAIKEN, S.** *The psychology of attitudes*. Fort Worth: Harcourt Brace Jovanovich College Publishers, 1993.
- EL-RAMADY, H. et al.** Review of crop response to soil salinity stress: possible approaches from leaching to nano-management. *Soil Systems*, v. 8, p. 6, 2024.
- HARRIS, M.** History and significance of the emic/etic distinction. *Annual Review of Anthropology*, v. 5, p. 329–350, 1976.
- HAYS, T. E.** An empirical method for the identification of covert categories in ethnobiology. *American Ethnologist*, v. 3, n. 3, p. 489–507, 1976.

- HE, C., LIU, Z., WU, J., PAN, X., FANG, Z., LI, J., & BRYAN, B. A.** (2021). Future global urban water scarcity and potential solutions. *Nature communications*, 12(1), 4667.
- HUNTINGTON, H. P.** Using traditional ecological knowledge in science: methods and applications. *Ecological Applications*, v. 10, n. 5, p. 1270–1274, 2000.
- INGRAO, C., STRIPPOLI, R., LAGIOIA, G., & HUISINGH, D.** (2023). Water scarcity in agriculture: An overview of causes, impacts and approaches for reducing the risks. *Heliyon*, 9(8).
- INTERNATIONAL SOCIETY OF ETHNOBIOLOGY.** *International Society of Ethnobiology Code of Ethics (with 2008 additions)*. Tucson: ISE, 2006. Disponível em: <https://ethnobiology.org/about-society-ethnobiology/ethics>. Acesso em: 13 maio 2026.
- KHAN, M.; AL-GHOUTI, M.** DPSIR framework and sustainable approaches of brine management from seawater desalination plants in Qatar. *Journal of Cleaner Production*, 2021.
- KHONDOKER, M.; MANDAL, S.; GURAV, R. G.; HWANG, S.** Freshwater shortage, salinity increase, and global food production: a need for sustainable irrigation water desalination—a scoping review. *Earth*, v. 4, n. 1, p. 1–24, 2023.
- LIU, X.; LIU, W.; TANG, Q. et al.** Global agricultural water scarcity assessment incorporating blue and green water availability under future climate change. *Earth's Future*, v. 10, 2022.
- LIU, Z. et al.** Scarcity and quality risks for future global urban water supply. *Landscape Ecology*, v. 39, p. 543–560, 2024.
- MCDONALD, R. P.** *Test theory: a unified treatment*. Mahwah: Lawrence Erlbaum Associates, 1999.
- NEWING, H.** *Conducting research in conservation: social science methods and practice*. London: Routledge, 2010.
- OMERSPAHIC, M.; AL-JABRI, H.; SIDDIQUI, S. A.; SAADAoui, I.** Characteristics of desalination brine and its impacts on marine chemistry and health, with emphasis on the Persian/Arabian Gulf: a review. *Frontiers in Marine Science*, v. 9, 2022.
- PAN, S.; HADDAD, A. Z.; KUMAR, A.; WANG, S.-W.** Brackish water desalination using reverse osmosis and capacitive deionization at the water-energy nexus. *Water Research*, v. 183, 2020.
- PANAGOPOULOS, A.; HARALAMBOUS, K.** Environmental impacts of desalination and brine treatment: challenges and mitigation measures. *Marine Pollution Bulletin*, v. 161, p. 111773, 2020.
- PARK, K.; MUDGAL, A.; MUDGAL, V.; SAGI, M.; STANDING, D.; DAVIES, P.** Desalination, water re-use, and halophyte cultivation in salinized regions: a highly productive groundwater treatment system. *Environmental Science & Technology*, v. 57, 2023.
- R CORE TEAM.** *R: a language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing, 2013. Disponível em: <https://www.r-project.org/>. Acesso em: 13 maio 2026.
- RICART, S., VILLAR-NAVASCUÉS, R. A., HERNÁNDEZ-HERNÁNDEZ, M., RICO-AMORÓS, A. M., OLCINA-CANTOS, J., & MOLTÓ-MANtero, E.** (2021). Extending

natural limits to address water scarcity? The role of non-conventional water fluxes in climate change adaptation capacity: A review. *Sustainability*, 13(5), 2473.

ROSA, L.; SANGIORGIO, M. Global water gaps under future warming levels. *Nature Communications*, 2025.

SCANLON, B. R.; FAKHREDDINE, S.; RATEB, A.; DE GRAAF, I. E. M.; FAMIGLIETTI, J.; GLEESON, T. et al. Global water resources and the role of groundwater in a resilient water future. *Nature Reviews Earth & Environment*, v. 4, 2023.

SHEMER, H.; WALD, S.; SEMIAT, R. Challenges and solutions for global water scarcity. *Membranes*, v. 13, 2023.

SOLIMAN, M.; YOUSSEF, P. G.; SALEM, M. A.; ABDELRAHMAN, A. K. Energy, environmental, and economic analysis of desalination technologies: a review. *Environmental Chemistry Letters*, v. 19, p. 889–919, 2021.

SIROTA, R.; WINTERS, G.; LEVY, O.; MARQUES, J. A.; PAYTAN, A.; SILVERMAN, J. et al. Impacts of desalination brine discharge on benthic ecosystems. *Environmental Science & Technology*, v. 58, p. 4234–4250, 2024.

SOLA, I. et al. Evaluating physico-chemical and biological impacts of brine discharges for a sustainable desalination development on South America's Pacific coast. *Journal of Hazardous Materials*, v. 469, p. 133224, 2025.

VILLAR-NAVASCÚES, R.; FRAGKOU, M. Managing water scarcity futures: identifying factors influencing water quality, risk perception and daily practices in urban environments after the introduction of desalination. *Water*, v. 13, p. 2723, 2021.

WANG, M., BODIRSKY, B.L., RIJNEVELD, R. et al. A triple increase in global river basins with water scarcity due to future pollution. *Nat Commun* 15, 880 (2024).

10. ANEXOS

Anexo 1: Termo de Consentimento Livre e Esclarecido (TCLE).

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Esclarecimentos

Este é um convite para você participar da pesquisa intitulada “*Percepção de risco ambiental e atitudes frente ao rejeito salino da dessalinização no semiárido brasileiro*”, coordenada pelo Prof. Dr. Heitor de Oliveira Braga, Professor Visitante do Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA) da Universidade Federal Rural do Semi-Árido (UFERSA).

Esta pesquisa segue rigorosamente as recomendações das Resoluções nº 466/2012 e nº 510/2016 do Conselho Nacional de Saúde, bem como suas normativas complementares. Sua participação é voluntária, o que significa que você poderá desistir a qualquer momento, retirando sua autorização, sem que isso lhe traga qualquer prejuízo, constrangimento ou penalidade.

Caso você decida aceitar o convite, sua participação consistirá na resposta a um questionário semiestruturado, contendo perguntas relacionadas ao uso da água dessalinizada, à destinação do rejeito salino, às observações sobre o solo, a vegetação, a fauna e os sistemas produtivos, bem como às suas percepções sobre riscos ambientais e práticas de manejo. A entrevista será conduzida pelo pesquisador responsável e poderá ser registrada por meio de anotações e, caso você autorize, por gravação de áudio.

O objetivo geral desta pesquisa é compreender como os agricultores familiares percebem os impactos ambientais associados ao rejeito salino da dessalinização, bem como identificar os conhecimentos ecológicos tradicionais utilizados para avaliar esses impactos, as percepções de manejo ambiental e as atitudes e a confiança no uso controlado desse rejeito nos sistemas produtivos familiares. Os resultados do estudo buscam contribuir para estratégias de gestão ambiental mais sustentáveis e contextualizadas à realidade do semiárido brasileiro.

Os benefícios esperados desta pesquisa incluem a valorização dos saberes e experiências dos agricultores familiares, o fortalecimento de práticas ambientais mais adequadas ao contexto local e a produção de informações que possam subsidiar políticas públicas, programas de extensão rural e tecnologias sociais voltadas ao uso sustentável da água. Ressalta-se que não haverá benefício financeiro direto ao participante.

Os riscos envolvidos neste estudo são considerados mínimos, estando principalmente relacionados à possibilidade de desconforto, timidez ou receio ao responder perguntas sobre práticas produtivas, uso da água e manejo ambiental. Esses riscos podem ser maiores em contextos nos quais alguns participantes não sejam alfabetizados. Para minimizar esses riscos, serão adotadas as seguintes medidas: garantia de anonimato e privacidade dos participantes; utilização de códigos de identificação nos questionários; sigilo absoluto sobre informações pessoais; incentivo à livre expressão, sem qualquer tipo de pressão; leitura integral deste termo para os participantes; e garantia do direito de recusar o uso de gravadores ou qualquer outro recurso de registro.

As informações coletadas serão armazenadas em meio digital protegido por senha e em arquivos físicos mantidos nas dependências do Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA), da Universidade Federal Rural do Semi-Árido (UFERSA), sob a responsabilidade do pesquisador responsável, pelo período mínimo de cinco anos, conforme as normas éticas vigentes. Após esse período, os documentos físicos serão destruídos de forma segura e os arquivos digitais serão permanentemente excluídos. Será assegurada a confidencialidade dos dados, sendo que os resultados da pesquisa poderão ser divulgados em eventos científicos e publicados em periódicos nacionais ou internacionais, sempre de forma agregada e sem identificação dos participantes.

Você ficará com uma via original deste Termo de Consentimento Livre e Esclarecido. Em caso de dúvidas sobre a pesquisa, poderá entrar em contato diretamente com o Prof. Dr. Heitor de Oliveira Braga, no Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA), Universidade Federal Rural do Semi-Árido (UFERSA), Campus Mossoró, Av. Francisco Mota, nº 572, Bairro Costa e Silva, Mossoró/RN, CEP 59625-900. Telefone: (34) 99716-6233.

Dúvidas relacionadas aos aspectos éticos desta pesquisa poderão ser esclarecidas junto ao Comitê de Ética em Pesquisa (CEP-UERN) – Faculdade de Medicina da Universidade do Estado do

Rio Grande do Norte, Rua Miguel Antônio da Silva Neto, s/n, Bairro Aeroporto, Mossoró/RN, CEP 59607-360. Telefone: (84) 3312-7032. E-mail: cep@uern.br; Site: <http://www.uern.br>.

Caso o participante tenha qualquer gasto decorrente de sua participação na pesquisa, fica garantido o direito à indenização, conforme previsto na Resolução nº 466/2012 (item II.7), e/ou ao ressarcimento, conforme item II.21 da mesma resolução, exclusivamente para despesas comprovadas, sob responsabilidade do pesquisador responsável. Não será oferecida nenhuma forma de gratificação pela participação neste estudo. O pesquisador permanecerá à disposição para quaisquer esclarecimentos durante todo o desenvolvimento da pesquisa.

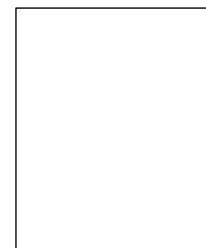
Consentimento Livre

Concordo em participar desta pesquisa intitulada “*Percepção de risco ambiental e atitudes frente ao rejeito salino da dessalinização no semiárido brasileiro*”, declarando, para os devidos fins, que fui devidamente esclarecido(a) quanto aos objetivos da pesquisa, aos procedimentos aos quais serei submetido(a) e aos possíveis riscos que possam surgir com a minha participação. Foram-me garantidos os esclarecimentos que eu venha a solicitar durante o desenvolvimento da pesquisa, bem como o direito de desistir da participação a qualquer momento, sem que essa decisão acarrete qualquer prejuízo à minha pessoa ou à minha família. Autorizo, assim, a utilização e a publicação dos dados obtidos nesta pesquisa para fins científicos, sendo garantidos o meu anonimato e o sigilo das informações referentes à minha identificação

_____, ____/____/____.
Cidade/UF

Assinatura do Pesquisador

Assinatura do Participante



IMPRESSÃO DATILOSCÓPICA
(preencher apenas quando aplicável)

Prof. Dr. Heitor de Oliveira Braga - Programa de Pós-Graduação em Desenvolvimento e Meio Ambiente (PRODEMA), Universidade Federal Rural do Semi-Árido (UFERSA), Campus Central, Av. Costa e Filho, nº 424, Presidente Costa e Silva, Mossoró-RN. Telefone: (34) 99716-6233.

Comitê de Ética em Pesquisa (CEP-UERN) - Faculdade de Medicina da UERN - Rua Miguel Antonio da Silva Neto, s/n - Aeroporto. Home page: <http://www.uern.br> - e-mail: cep@uern.br – CEP: 59607-360 - Mossoró –RN Tel: (84) 3312-7032.

Anexo 2. Roteiro de entrevista semiestruturado que será aplicado aos agricultores.

QUESTIONÁRIO SEMI-ESTRUTURADO

PERCEPÇÃO DE RISCO AMBIENTAL E ATITUDES FRENTE AO REJEITO SALINO DA DESSALINIZAÇÃO NO SEMIÁRIDO BRASILEIRO

Comunidade: _____ **Dia** ____/____/____.

Nº da Entrevista _____. **Aplicador da Entrevista:** _____.

BLOCO I – Dados Sociodemográficos, Produtivos e de Inserção Social

1) Idade (____ anos); Classe etária: () ≤30 () 31–40 () 41–50 () 51–60 () ≥61

2) Sexo: () Feminino () Masculino () Outro () Prefere não informar

3) Escolaridade:

() Classe A – Baixa (não estudou ou EF incompleto)

() Classe B – Básica (EF completo ou EM incompleto)

() Classe C – Média ou superior (EM completo ou ES)

4) Número de pessoas na família:

5) Atividade produtiva principal:

() Agricultura () Pecuária () Piscicultura () Mista

() Outras:

6) Fonte(s) de renda da família:

7) Renda familiar mensal (salário-mínimo = R\$ 1.502,00):

() Classe A – Baixa renda: ≤ R\$ 1.502,00 (até 1 salário-mínimo)

() Classe B – Renda básica: R\$ 1.502,01 a R\$ 3.004,00 (de 1 a 2 salários-mínimos)

() Classe C – Renda intermediária: R\$ 3.004,01 a R\$ 6.008,00 (de 2 a 4 salários-mínimos)

() Classe D – Renda mais elevada: ≥ R\$ 6.008,01 (acima de 4 salários-mínimos)

8) Tempo de residência na comunidade: ____ anos

9) Tempo de experiência agrícola: ____ anos

10) Acesso a dessalinizador: () Sim () Não

11) Tempo de instalação: ____ anos

12) Tempo de uso da água dessalinizada: ____ anos

13) Responsável pelo dessalinizador:

() Associação () Cooperativa () Prefeitura () ONG/empresa

() Grupo de moradores () Não sabe () Outro

14) Participa de associação/cooperativa/projeto?

() Não () Sim. Qual? _____

Nível: () Associado () Ativo () Liderança

BLOCO II – Índices Etnobiológicos

(0) Não sei / Não tenho opinião (1) Discordo totalmente (2) Discordo (3) Nem concordo nem discordo (4) Concordo (5) Concordo totalmente

- *Conhecimento Ecológico Tradicional (CET) acerca do Percepção de Manejo Ambiental (CET-PMA)*

1. O rejeito salino pode deixar o solo mais salgado com o tempo. ()
2. O rejeito pode prejudicar o crescimento das plantas cultivadas. ()
3. O rejeito é reaproveitado para alguma finalidade. ()
4. O rejeito pode fornecer suporte hídrico aos animais. ()
5. O rejeito pode prejudicar animais da criação ou do mato. ()
6. O rejeito pode afetar a vegetação nativa da região. ()
7. O rejeito pode dificultar a entrada da água no solo. ()
8. O rejeito pode estragar áreas da propriedade ao longo do tempo. ()
9. Os problemas causados pelo rejeito aumentam com os anos. ()
10. O rejeito precisa de cuidado maior principalmente nos períodos de seca. ()

- *Percepção de Risco Ambiental associada ao rejeito salino (PRA)*

(0) Não sei / Não tenho opinião (1) Discordo totalmente (2) Discordo (3) Nem concordo nem discordo (4) Concordo (5) Concordo totalmente

11. O solo que recebe rejeito salino fica mais salgado. ()
12. Percebo a presença de crosta branca de sais (superfície esbranquiçada) no solo quando despeja rejeito. ()
13. O solo fica com a superfície escura quando despeja rejeito. ()
14. O solo que recebe rejeito salino apresenta pequenas rachaduras. ()
15. Consigo perceber que a quantidade de plantas diminui com a presença do rejeito salino. ()
16. Tenho conhecimento de quais plantas sofrem mais quando há a utilização do rejeito salino. ()
17. Algumas plantas ficam com coloração amarelada onde o rejeito salino foi despejado. ()
18. A folha da planta fica escura/amarronzada nos ambientes onde o rejeito foi despejado. ()
19. Noto que a altura e o número de folhas de algumas plantas diminuem onde foi despejado o rejeito salino. ()
20. Consigo identificar os animais que conseguem tomar água do rejeito salino. ()

- *Atitude e Confiança no Uso do Rejeito Salino (ACUR)*

(0) Não sei / Não tenho opinião (1) Discordo totalmente (2) Discordo (3) Nem concordo nem discordo (4) Concordo (5) Concordo totalmente

21. Acredito que o rejeito pode ser usado para a produção animal e vegetal, desde que com cuidado.
()
22. Posso produzir uma lavoura usando o rejeito salino sem prejudicar o solo. ()

23. Posso criar meus animais usando água do rejeito. ()
24. Usaria mais o rejeito se tivesse orientação técnica. ()
25. O rejeito ajuda principalmente quando falta água. ()
26. Prefiro usar o rejeito com controle a jogar fora sem cuidado. ()
27. O rejeito pode fazer parte do sistema produtivo da propriedade. ()
28. O rejeito salino pode aumentar a produção da comunidade. ()
29. A renda familiar pode aumentar se usar o rejeito salino para agricultura. ()
30. Acredito que o uso do rejeito pode fortalecer a produção da comunidade. ()

BLOCO III – Gestão Ambiental Local e Observações Ecológicas

31. O rejeito salino é direcionado para local específico? () Sim () Não
Se sim, para onde?
32. A família observa mudanças ambientais associadas ao rejeito? () Solo () Vegetação ()
Água () Fauna () Nenhuma
Descreva:
33. Há plantas mais afetadas do que outras?
() Sim () Não
Quais?
34. Há animais mais afetados do que outros quando usam a água do rejeito? () Sim () Não
Quais?
35. Qual a principal destinação do rejeito?
() Solo () Irrigação () Para o animal beber () Outros
36. A comunidade discute como lidar com o rejeito na produção familiar?
() Sim () Não () Às vezes
Onde? Exemplo:
37. Na sua opinião, qual a melhor forma de lidar com o rejeito salino?



Universidade Federal Rural do Semi-Árido (UFERSA)
Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG)
6ª Reunião Ordinária de 2026 do CPPGIT

3º PONTO

Apreciação e deliberação sobre a solicitação do diploma do egresso Augusto César de Sousa do Programa de Pós-Graduação em Engenharia Elétrica.

MEMORIAL JUSTIFICATIVO

Ao Colegiado do Programa de Pós-Graduação em Engenharia Elétrica – PPGEE

Assunto: Justificativa para apreciação da entrega extemporânea da versão final da dissertação de mestrado.

Eu, **Augusto César de Sousa**, aprovado na defesa de dissertação do curso de mestrado do Programa de Pós-Graduação em Sistemas de Comunicação e Automação da Universidade Federal Rural do Semi-Árido - UFERSA, realizada em **17 de julho de 2014**, venho, respeitosamente, apresentar os esclarecimentos a seguir, com a finalidade de subsidiar a apreciação deste colegiado quanto ao pedido de autorização para entrega da versão final da dissertação e prosseguimento dos procedimentos necessários à obtenção do título de mestre.

Após a defesa, a banca examinadora aprovou o trabalho condicionando a homologação do título ao atendimento das recomendações constantes na ata de defesa, dentre as quais se destacavam a alteração do título da dissertação, a redefinição dos objetivos, a reestruturação dos capítulos, o aperfeiçoamento da modelagem matemática, a justificativa da utilização dos sistemas de coordenadas, a implementação do modelo para validação do controlador neural e a reformulação do capítulo de conclusões.

Logo após a defesa, iniciei a execução dessas modificações. Entretanto, durante esse processo ocorreu um problema relacionado ao arquivo principal do projeto desenvolvido em **MATLAB/Simulink**, ambiente no qual estavam implementados o controlador proposto, as simulações, os algoritmos e os códigos-fonte que fundamentavam os resultados apresentados na dissertação e que deveriam acompanhar a versão final do trabalho para possibilitar a verificação da implementação realizada.

Embora atualmente eu não consiga afirmar com absoluta certeza se o arquivo foi sobrescrito por uma versão anterior ou se ocorreu alguma falha durante o processo de salvamento, o fato é que esse arquivo principal não foi encontrado. Como consequência, tornou-se inviável concluir, dentro do prazo regulamentar, toda a reconstrução da implementação computacional, a repetição das simulações, a validação dos resultados e a organização dos códigos que deveriam acompanhar a versão definitiva da dissertação.

Após o transcurso do prazo regulamentar, compreendi, à época, que não existia mais possibilidade administrativa de concluir o processo de obtenção do título. Essa compreensão decorreu da leitura do art. 74, § 3º, do Regulamento Geral dos Programas de Pós-Graduação Stricto Sensu da Universidade Federal Rural do Semi-Árido – UFERSA, que dispõe:

"§ 3º O prazo de que trata o parágrafo anterior não pode ultrapassar 90 (noventa) dias após a data da defesa, sob pena do discente perder o direito ao título de Mestre ou Doutor."

Diante da redação desse dispositivo, entendi que minha situação acadêmica se encontrava definitivamente encerrada, razão pela qual não apresentei qualquer requerimento administrativo ao programa ou à Universidade durante os anos seguintes.

No final de 2025, tomei conhecimento de relatos de situações excepcionais que haviam sido submetidas à apreciação dos órgãos competentes da Universidade. Esse fato levou-me a reavaliar minha situação e a verificar se ainda seria possível reconstruir toda a implementação computacional exigida pela banca examinadora antes de formular qualquer solicitação administrativa.

Desde dezembro de 2025, iniciei um trabalho contínuo de reconstrução do projeto, envolvendo a reimplementação dos modelos desenvolvidos em **MATLAB/Simulink**, a recuperação dos algoritmos, a repetição das simulações, a validação dos resultados obtidos e a revisão completa da dissertação. Paralelamente, foram atualizados a fundamentação teórica, a modelagem matemática, a organização dos capítulos, as figuras, os quadros, as tabelas e as análises dos resultados, de modo a atender integralmente às recomendações registradas na ata de defesa.

Esse trabalho estendeu-se por aproximadamente sete meses e somente foi concluído em julho de 2026, quando a dissertação passou a reunir as condições técnicas necessárias para ser apresentada como versão final.

Somente após concluir esse processo de reconstrução e verificar que havia conseguido atender às recomendações estabelecidas pela banca examinadora iniciei os procedimentos administrativos para regularização da minha situação acadêmica.

Inicialmente, procurei a Biblioteca da UFRSA, onde fui atendido pela Diretora da Biblioteca, bibliotecária Keina Cristina Santos Sousa e Silva. Após expor o histórico do caso, recebi as orientações necessárias para realizar a submissão da versão final da dissertação ao repositório institucional da universidade. Seguindo essas orientações, efetuei a submissão do trabalho.

Durante a análise técnica da submissão, a equipe responsável pelo Repositório verificou que a defesa havia ocorrido em 2014 e informou que, em razão do decurso do prazo regulamentar, seria necessária manifestação da Pró-reitoria de Pesquisa e Pós-Graduação para prosseguimento do processo.

Em razão dessa orientação, procurei pessoalmente a Pró-reitoria de Pesquisa e Pós-Graduação, onde fui recebido pela pró-reitora. Após tomar conhecimento dos fatos, fui informado de que o processo deveria ser encaminhado ao colegiado do Programa de Pós-Graduação em Engenharia Elétrica – PPGE, por ser o órgão competente para apreciar a situação apresentada.

Na sequência, procurei o coordenador do PPGE, que me orientou a encaminhar toda a documentação necessária por meio do formulário de solicitação de diploma disponibilizado pelo Programa, incluindo este memorial justificativo e o relatório técnico

que demonstra o atendimento das determinações estabelecidas pela banca examinadora.

Registro que meu objetivo não é afastar a responsabilidade pelo descumprimento do prazo regulamentar. Reconheço que a versão final da dissertação não foi entregue dentro do período originalmente previsto. O presente pedido busca apenas submeter ao colegiado as circunstâncias excepcionais que impediram a conclusão do processo naquele momento e demonstrar que, após a reconstrução integral da implementação computacional e a revisão completa da dissertação, todas as exigências acadêmicas estabelecidas pela banca examinadora foram efetivamente cumpridas, conforme detalhado no relatório técnico que acompanha este memorial.

Diante do exposto, submeto respeitosamente minha situação à apreciação deste colegiado, solicitando que sejam consideradas as circunstâncias aqui relatadas e, caso assim entendam seus membros, seja autorizada a continuidade dos procedimentos administrativos necessários à homologação da versão final da dissertação e à posterior emissão do diploma de mestre.

Por fim, manifesto meu respeito às normas institucionais e às decisões que vierem a ser adotadas por este colegiado, colocando-me à disposição para prestar quaisquer esclarecimentos adicionais ou apresentar a documentação complementar que se fizer necessária.

Nestes termos,

Pede deferimento.

 Informamos aos concluintes que após a validação do orientador no sistema e processar os dados. Por isso, NÃO DEIXE PARA ENVIAR O TCC (graduação repositório@ufersa.edu.br 

BIBLIOTECA PRECISA DE 7 DIAS ÚTEIS para tratar pós-graduação) DE ÚLTIMA HORA. Dúvidas:

[Início](#) • MeuDSpace

 Arraste e solte seus arquivos aqui, ou [Navegar](#)



dos

Todo o DSpace

Pesquise no repositório...

 Buscar



Ferramentas de busca

Suas submissões

Agora exibindo 1 - 1 de 1



Nenhuma
Miniatura dispo-
nível

Fluxo de trabalho

Item

Proposta de um controlador PID-Neural para estabilização em voo pairado de um quadrotor

(2014-04-17) Sousa, Augusto César de; Villarreal, Elmer Rolando Llanos; elmerll

Este trabalho aborda o controle de atitude de um Veículo Aéreo Não Tripulado (VANT) do tipo quadrotor por meio da integração de uma Rede Neural Artificial (RNA) ao controlador clássico Proporcional Integral Derivativo (PID). A natureza não linear da dinâmica desses veículos torna

[Mostrar mais](#)

 Visualizar



 Informamos aos concluintes que após a validação do orientador no sistema e processar os dados. Por isso, NÃO DEIXE PARA ENVIAR O TCC (graduação repositório@ufersa.edu.br 

BIBLIOTECA PRECISA DE 7 DIAS ÚTEIS para tratar pós-graduação) DE ÚLTIMA HORA. Dúvidas:

[Início](#) • Visualização do FLuxo de ...

Proposta de um controlador PID-Neural para estabilização em voo pairado de um quadrotor

dc.contributor.advisor	Villarreal, Elmer Rolando Llanos
dc.contributor.advisormail	elmerllanos@ufersa.edu.br
dc.contributor.author	Sousa, Augusto César de
dc.contributor.coadvisor	Silva, Alex Sandro de
dc.contributor.referee1	Villarreal, Elmer Rolando Llanos
dc.contributor.referee2	Silva, Alex Sandro de
dc.contributor.referee3	Neto, André Pedro Fernandes
dc.contributor.referee4	Vargas, José Alfredo Ruiz
dc.contributor.referee5	Brandão, Alexandre Santos
dc.coverage.spatial	Mossoró
dc.date.issued	2014-04-17
dc.description.abstract	Este trabalho aborda o controle de atitude de um Veículo Aéreo Não Tripulado (VANT) do tipo quadrotor por meio da integração de uma Rede Neural Artificial (RNA) ao controlador clássico Proporcional Integral Derivativo (PID). A natureza não linear da dinâmica desses veículos torna o desenvolvimento de estratégias de controle capazes de garantir estabilidade, robustez e desempenho um dos principais desafios para sua operação autônoma. Nesse contexto, o objetivo desta pesquisa consiste em propor e avaliar um controlador PID-Neural para estabilização de um quadrotor em voo pairado (hover), empregando uma estratégia de compensação residual aplicada ao canal de empuxo, de modo a aprimorar a dinâmica vertical sem comprometer a estabilidade dos canais de atitude controlados pelo PID convencional. A metodologia compreendeu a revisão da literatura sobre modelagem e controle de quadrotores, a modelagem matemática da dinâmica da aeronave pelo formalismo de Euler-Lagrange, o desenvolvimento e treinamento de uma Rede Neural Artificial do tipo Multi-Layer Perceptron (MLP), a implementação computacional dos controladores PID convencional e PID-Neural em ambiente MATLAB/Simulink e a realização de simulações em diferentes cenários operacionais, incluindo variações paramétricas de massa, momento de inércia e amortecimento. Os resultados demonstraram que a estratégia proposta foi capaz de manter a estabilidade do sistema em todos os cenários analisados, preservando o desempenho dos canais de atitude e reduzindo significativamente o erro de altitude quando comparada ao controlador PID convencional. Nos cenários com variação de massa, a redução do erro quadrático médio da altitude atingiu aproximadamente 44%, evidenciando maior robustez frente às incertezas do modelo. Além disso, a contribuição residual da RNA produziu correções suaves no sinal de empuxo, sem introduzir oscilações relevantes nos demais canais de controle. Conclui-se que a estratégia híbrida proposta mostrou-se eficaz para o controle de voo pairado de quadrotor, reunindo a estabilidade do controlador PID com a capacidade adaptativa das Redes Neurais Artificiais, constituindo uma alternativa promissora para aplicações futuras em sistemas de controle de aeronaves não tripuladas.

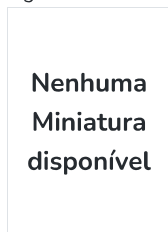
dc.description.abstract2	This work addresses the attitude control of a quadrotor Unmanned Aerial Vehicle (UAV) through the integration of an Artificial Neural Network (ANN) with a classical Proportional-Integral-Derivative (PID) controller. The nonlinear dynamics of quadrotor systems make the design of control strategies capable of ensuring stability, robustness, and satisfactory performance one of the main challenges for autonomous flight. In this context, this research proposes and evaluates a PID-Neural controller for quadrotor hover stabilization based on a residual neural compensation strategy applied exclusively to the thrust control channel, while maintaining the roll, pitch, and yaw torques under the conventional PID controller. The proposed approach aims to improve the vertical dynamic response without compromising the attitude stabilization provided by the classical controller. The methodology included a literature review on quadrotor modeling and control, mathematical modeling of the vehicle dynamics using the Euler-Lagrange formulation, development and training of a Multi-Layer Perceptron (MLP) neural network, computational implementation of the conventional PID and PID-Neural controllers in the MATLAB/Simulink environment, and simulation-based evaluation under different operating conditions, including variations in mass, inertia, and damping parameters. The simulation results demonstrated that the proposed strategy maintained system stability in all evaluated scenarios while significantly reducing the altitude tracking error compared with the conventional PID controller. In scenarios involving mass variations, the root mean square error (RMSE) of the altitude was reduced by approximately 44%, demonstrating improved robustness against modeling uncertainties. Furthermore, the residual neural compensation generated smooth corrections in the thrust command without introducing significant oscillations in the attitude control channels. These results indicate that the proposed hybrid strategy successfully combines the robustness and simplicity of the PID controller with the adaptive capability of Artificial Neural Networks, representing a promising approach for quadrotor hover control and future autonomous flight applications.
dc.format.mimetype	pdf
dc.identifier.advisorLattes	http://lattes.cnpq.br/5447452711376721
dc.identifier.advisorOrcid	https://orcid.org/0000-0002-3059-3340
dc.identifier.authorLattes	https://lattes.cnpq.br/2740097293615818
dc.identifier.authorOrcid	https://orcid.org/0009-0008-5946-4797
dc.identifier.bibliographicCitation	SOUSA, Augusto César de. Proposta de um controlador PID-neural para estabilização em voo pairado de um quadrotor. 2014. 210 f. Dissertação (Mestrado em Sistemas de Comunicação e Automação) — Universidade Federal Rural do Semi-Árido, Mossoró, 2014.
dc.identifier.coadvisorLattes	http://lattes.cnpq.br/1016660828775047
dc.identifier.coadvisorOrcid	https://orcid.org/0000-0002-0130-7210
dc.language.iso	pt_BR
dc.publisher.center	CE
dc.publisher.country	Brasil
dc.publisher.department	DET
dc.publisher.initials	UFERSA
dc.publisher.institution	Universidade Federal Rural do Semi-Árido
dc.publisher.program	Programa de Pós-Graduação em Engenharia Elétrica - PPGEE
dc.rights	info:eu-repo/semantics/openAccess
dc.rights.holder	UFERSA

dc.rights.license	Attribution-ShareAlike 3.0 Brazil	en
dc.rights.uri	http://creativecommons.org/licenses/by-sa/3.0/br/	
dc.subject.cnpq	ENGENHARIAS::ENGENHARIA ELETRICA	
dc.subject.keyword	Quadrotor	
dc.subject.keyword	Controle PID	
dc.subject.keyword	Redes Neurais Artificiais	
dc.subject.keyword	Controle Neural	
dc.subject.keyword	Voo Pairado	
dc.subject.keyword	Quadrotor	
dc.subject.keyword	PID Control	
dc.subject.keyword	Artificial Neural Networks	
dc.subject.keyword	Neural Control	
dc.subject.keyword	Hover Flight	
dc.title	Proposta de um controlador PID-Neural para estabilização em voo pairado de um quadrotor	
dc.title.alternative	Proposal of a PID-Neural Controller for Quadrotor Hover Stabilization	
dc.type	info:eu-repo/semantics/masterThesis	

Arquivos

Pacote Original

Agora exibindo 1 - 1 de 1



Nome:
Augusto Cesar de Sousa - Dissertação.pdf

Tamanho:
5.34 MB

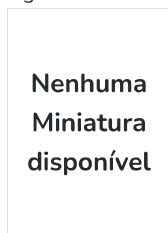
Formato:
Adobe Portable Document Format

Descrição:

[Baixar](#)

Licença do Pacote

Agora exibindo 1 - 1 de 1



Nome:
license.txt

Tamanho:
1.58 KB

Formato:
Item-specific license agreed upon to submission

Descrição:

[Baixar](#)

[← Voltar](#)

@ 2026 UFERSA - Todos os direitos reservados -
[contato com a administração](#) | [Estatísticas](#)





UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
SISTEMA INTEGRADO DE GESTÃO DE ATIVIDADES
ACADÊMICAS



EMITIDO EM 22/06/2026 18:33

SIGAA - BIBLIOTECAS

DECLARAÇÃO DE QUITAÇÃO

MATRÍCULA: 2013101640 (*identificador*)

USUÁRIO: Sr(a). AUGUSTO CESAR DE SOUSA

VÍNCULO DO USUÁRIO: ALUNO DE PÓS-GRADUAÇÃO

CENTRO: UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO

CURSO: MESTRADO EM ENGENHARIA ELÉTRICA

Declaramos, para os devidos fins, que em nome do usuário(a) supracitado(a), não existe débitos nas bibliotecas da Ufersa feitos com o vínculo ALUNO DE PÓS-GRADUAÇÃO acima mostrado.

Esse vínculo foi quitado e não poderá mais ser usado para realizar empréstimos.

ATENÇÃO

Para verificar a autenticidade deste documento acesse <https://sigaa.ufersa.edu.br/sigaa/documentos/> informando o identificador, a data de emissão e o código de verificação **2cbf4a66a1**

SIGAA | Copyright © 2006-2026 - Superintendência de Tecnologia da Informação e Comunicação - UFRN - (84)
3317-8210 - sig-prd-sigaa02.sigaa02

Histórico Acadêmico - Emitido em: 20/07/2026 às 15:17

Dados Pessoais

Nome: **AUGUSTO CESAR DE SOUSA** Matrícula: **[REDACTED]**
Data de Nascimento: **[REDACTED]** Local de Nascimento: **[REDACTED]**
Nome do Pai: **[REDACTED]**
Nome da Mãe: **[REDACTED]**
Endereço: **[REDACTED]** Bairro: **[REDACTED]**
Município: **[REDACTED]** UF: **RN**

Dados do Vínculo do Discente

Programa: **PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA** Índices Acadêmicos
Nível: **MESTRADO** CR: **8.3286**
Curso: **MESTRADO EM ENGENHARIA ELÉTRICA**
Currículo: **2011** Status: **DEFENDIDO**
Área de Concentração: **ENGENHARIA ELÉTRICA**
Linha de Pesquisa: **TELECOMUNICAÇÕES E ELETROMAGNETISMO APLICADO**
Orientador: **1673887 - ELMER ROLANDO LLANOS VILLARREAL**
Coorientador(es):

Forma de Ingresso: **SELECAO POS-GRADUACAO**
Mês/Ano Inicial: **MAR/2013** Mês Atual: **160°**
Trancamentos: **0 meses** Prazo para Conclusão **FEV/2015**
Prorrogações: **0 meses**
Tipo Saída:
Mês/Ano de Saída: Data da Defesa: **18/07/2014**

Disciplinas/Atividades Cursadas/Cursando

Início	Fim	Componente Curricular	Turma	CH	Freq %	Nota	Situação
01/2013	07/2013	PSC0001 FUNDAMENTOS DE MODELAGEM COMPUTACIONAL <i>Dr. IDALMIR DE SOUZA QUEIROZ JÚNIOR (60h)</i>	01	60	--	7.5	APROVADO
01/2013	06/2013	PSC0007 SISTEMAS DE CONTROLE <i>Dra. DANIELLE SIMONE DA SILVA CASILLO (60h)</i>	01	60	--	7.0	APROVADO
01/2013	06/2013	PSC0036 METODOLOGIA DE PESQUISA E REDAÇÃO CIENTÍFICA <i>Dra. FABIANA KARLA DE OLIVEIRA MARTINS VARELLA (60h)</i>	01	60	--	7.0	APROVADO
06/2013	06/2013	PSC0048 EXAME DE PROFICIÊNCIA EM LÍNGUA ESTRANGEIRA null	--	0	--	--	APROVADO
08/2013	11/2013	PSC0011 SISTEMAS LINEARES <i>Dr. ELMER ROLANDO LLANOS VILLARREAL (60h)</i>	01	60	--	9.8	APROVADO
08/2013	--	PSC0038 ESTUDOS DIRIGIDOS I null	--	0	--	--	MATRICULADO
08/2013	02/2014	PSC0047 EXAME DE QUALIFICAÇÃO null	--	0	--	--	APROVADO
08/2013	11/2013	PSC0049 TÓPICOS ESPECIAIS Teoria dos Autônomos Finitos <i>Dr. ANTONIO RONALDO GOMES GARCIA (60h)</i>	01	60	--	8.0	APROVADO
03/2014	05/2014	PSC0012 SISTEMAS DE CONTROLE MULTIVARIÁVEIS <i>Dr. ELMER ROLANDO LLANOS VILLARREAL (60h)</i>	01	60	--	10.0	APROVADO
03/2014	06/2014	PSC0034 TÉCNICAS DE CONTROLE NÃO LINEARES <i>Dr. ALEX SANDRO DE ARAUJO SILVA (60h)</i>	01	60	--	9.0	APROVADO
07/2014	07/2014	PSC0046 TRABALHO DE DISSERTAÇÃO null	--	0	--	--	APROVADO

Carga Horária Integralizada/Pendente

	Obrigatórias	Optativos	Total
Exigido	60 h	240 h	300 h
Integralizado	120 h	300 h	420 h
Pendente*	0 h	0 h	0 h

*Contabilizado com base no valor estabelecido no mínimo exigido da estrutura curricular.

Histórico Acadêmico - Emitido em: 20/07/2026 às 15:17

Nome: **AUGUSTO CESAR DE SOUSA**

Matrícula: **2013101640**

Dados da Defesa

Título: DESENVOLVER UM CONTROLADOR DE ALTITUDE DE UM QUADROTOR ARDRONE 2.0 (PARROT) UTILIZANDO REDES NEURAIAS

Palavras-Chave:

Quadrotor, Ardrone 2.0, Modelagem Matemática, Redes Neurais Artificiais.

Páginas: 70

Data: 18/07/2014

Grande Área: Engenharias

Área: Engenharia Elétrica

Sub-Área:

Especialidade:

Membros: Presidente - 1673887 - ELMER ROLANDO LLANOS VILLARREAL - UFERSA
Interno - 1810852 - ANDRE PEDRO FERNANDES NETO - UFERSA
Externo ao Programa - 1941853 - ALEX SANDRO DE ARAUJO SILVA - UFERSA
Externo à Instituição - ALEXANDRE SANTOS BRANDÃO - UFV
Externo à Instituição - JOSE ALFREDO RUIZ VARGAS - UnB

Atenção, agora o histórico possui uma verificação automática de autenticidade e consistência, sendo portanto dispensável a assinatura da coordenação do curso ou PROPPG. Favor, ler instruções no rodapé.



DECLARAÇÃO

Revista DCS, ISSN 2224-4131, declara para os devidos fins, que o artigo intitulado **Proposta de um controlador PID-neural para estabilização em voo pairado**, de autoria de Augusto César de Sousa, Alex Sandro de Araújo Silva, Elmer Rolando Llanos Villarreal, foi publicado no v.23, n.90, de 2026.

A revista é on-line, e os artigos podem ser encontrados ao acessar o link:

<https://ojs.revistadcs.com/index.php/revista/issue/view/92>

DOI: <https://doi.org/10.54899/dcs.v23i90.5490>

Por ser a expressão da verdade, firmamos a presente declaração.

Curitiba, 22 maio 2026

Equipe Editorial



RELATÓRIO DE ALTERAÇÕES SUGERIDAS PELA BANCA

Atendimento às Determinações da Banca Examinadora

Em atendimento à solicitação da Coordenação do Programa de Pós-Graduação em Engenharia Elétrica – PPGEE, apresenta-se o presente relatório com a finalidade de demonstrar o cumprimento das determinações estabelecidas pela banca examinadora na ata de defesa da dissertação realizada em **17 de julho de 2014**.

A análise foi elaborada mediante comparação entre as recomendações constantes na ata de defesa, a versão originalmente apresentada à banca examinadora e a versão final da dissertação submetida ao repositório institucional da UFERSA.

A primeira recomendação da banca consistiu na alteração do título da dissertação. Essa determinação foi integralmente atendida, passando o trabalho a adotar o título "**Proposta de um controlador PID-Neural para estabilização em voo pairado de um quadrotor**", refletindo de forma mais precisa o escopo da pesquisa e a solução efetivamente desenvolvida.

Também foi determinada a redefinição dos objetivos da pesquisa. Na versão final, o objetivo geral e os objetivos específicos foram reformulados para estabelecer maior coerência entre a proposta da pesquisa, a metodologia empregada e os resultados obtidos, enfatizando o desenvolvimento do controlador PID-Neural, sua implementação computacional e a avaliação comparativa em relação ao controlador PID convencional.

Em relação à estrutura da dissertação, a banca recomendou a reorganização dos capítulos. Essa recomendação foi plenamente atendida mediante reestruturação do texto, proporcionando melhor organização dos conteúdos, maior integração entre fundamentação teórica, modelagem matemática, metodologia, implementação computacional, resultados e conclusões, conferindo ao trabalho maior consistência técnica e melhor sequência lógica.

Outra recomendação consistiu no aperfeiçoamento da modelagem matemática pelo método de Euler-Lagrange. Na versão definitiva, essa parte foi substancialmente ampliada, contemplando a descrição dos referenciais adotados, a modelagem cinemática e dinâmica do quadrotor, a dedução das equações de movimento, a representação em espaço de estados e o modelo reduzido para voo pairado, proporcionando fundamentação matemática mais completa para o desenvolvimento do controlador proposto.

Quanto à justificativa para utilização dos sistemas de coordenadas empregados na modelagem, a versão final passou a apresentar fundamentação específica acerca dos referenciais inercial e solidário ao corpo, explicitando sua importância para a representação da dinâmica do quadrotor e para a obtenção das equações utilizadas ao longo do trabalho.

A implementação do modelo para validação do controlador neural representou a principal recomendação da banca examinadora e correspondeu à etapa de maior

complexidade técnica da revisão. Na versão final foram desenvolvidos e documentados a implementação completa em **MATLAB/Simulink**, a arquitetura do controlador PID-Neural, a geração e o tratamento do conjunto de dados para treinamento, o treinamento e a validação da rede neural artificial, os cenários de simulação, os indicadores quantitativos de desempenho e a comparação entre o controlador PID convencional e a estratégia PID-Neural proposta. Além disso, a dissertação passou a apresentar a documentação necessária da implementação computacional, incluindo algoritmos, fluxogramas, figuras, tabelas e demais elementos que permitem compreender e verificar o desenvolvimento realizado.

Por fim, o capítulo de conclusões foi integralmente reformulado, passando a apresentar uma síntese dos resultados alcançados, as contribuições da pesquisa, suas limitações e sugestões para trabalhos futuros, estabelecendo correspondência direta entre os objetivos inicialmente propostos e os resultados obtidos nas simulações.

Diante da análise realizada, verifica-se que todas as determinações constantes na ata de defesa foram contempladas na versão final da dissertação. As alterações implementadas ultrapassam a realização de ajustes pontuais, abrangendo ampla revisão estrutural, atualização da fundamentação teórica, aperfeiçoamento da modelagem matemática, reconstrução e documentação da implementação computacional, reorganização dos resultados e reformulação das conclusões.

Assim, **sob o aspecto estritamente acadêmico**, conclui-se que a versão final da dissertação atende às recomendações estabelecidas pela banca examinadora, não sendo identificadas pendências técnicas relacionadas às determinações registradas na ata de defesa.



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE COMUNICAÇÃO E
AUTOMAÇÃO
MESTRADO EM SISTEMAS DE COMUNICAÇÃO E AUTOMAÇÃO

AUGUSTO CÉSAR DE SOUSA

**Proposta de um controlador PID-Neural para estabilização em voo pairado de um
quadroto**

MOSSORÓ

2014

Augusto César de Sousa

Proposta de um controlador PID-Neural para estabilização em voo pairado de um quadrotor

Dissertação apresentada ao mestrado em Sistemas de Comunicação e Automação do Programa de Pós-Graduação em Sistemas de Comunicação e Automação, da Universidade Federal Rural do Semi-Árido como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

Linha de Pesquisa: Controle e Sistemas de Energia.

Orientador: Elmer Rolando Llanos Villarreal, Prof. Dr.

Coorientador: Alex Sandro de Araújo Silva, Prof. Dr.

MOSSORÓ

2014

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S725p Sousa, Augusto César de.
Proposta de um controlador PID-neural para
estabilização em voo pairado de um quadrotor /
Augusto César de Sousa. - 2014.
210 f. : il.

Orientador: Elmer Rolando Llanos Villarreal.
Coorientador: Alex Sandro de Araújo Silva.
Dissertação (Mestrado) - Universidade Federal
Rural do Semi-árido, Programa de Pós-graduação em
Sistemas de Comunicação e Automação, 2014.

1. Quadrotor. 2. Controle PID. 3. Redes
Neurais Artificiais. 4. Controle Neural. 5. Voo
Pairado. I. Villarreal, Elmer Rolando Llanos,
orient. II. Silva, Alex Sandro de Araújo, co-
orient. III. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo autor(a).

Biblioteca Campus Mossoró / Setor de Informação e Referência

Bibliotecária: Keina Cristina Santos Sousa e Silva

CRB: 15/120

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

Augusto César de Sousa

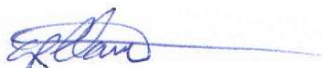
**Proposta de um controlador PID-Neural para estabilização em voo pairado de um
quadrotor**

Dissertação apresentada ao mestrado em
Sistemas de Comunicação e Automação do
Programa de Pós-Graduação em Sistemas de
Comunicação e Automação, da Universidade
Federal Rural do Semi-Árido como requisito
para a obtenção do título de Mestre em
Sistemas de Comunicação e Automação.

Linha de Pesquisa: Controle e Sistemas de
Energia

APROVADA EM: 17 / 07 / 2014.

BANCA EXAMINADORA



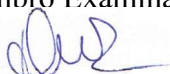
Elmer Rolando Llanos Villarreal, Prof. Dr. (UFERSA)
Presidente



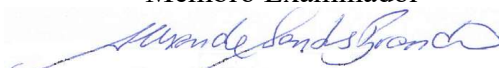
Alex Sandro de Araújo Silva, Prof. Dr. (UFERSA)
Membro Examinador



André Pedro Fernandes Neto, Prof. Dr. (UFERSA)
Membro Examinador



José Alfredo Ruiz Vargas, Prof. Dr. (UNB)
Membro Examinador



Alexandre Santos Brandão, Prof. Dr. (UFV)
Membro Examinador

DEDICATÓRIA

Este trabalho é dedicado, em primeiro lugar, a DEUS, aos meus pais Edinor Cirilo de Sousa (*in memoriam*) e Maria de Lourdes de Sousa (*in memoriam*), a minha esposa Maria Rosenilza Pereira Feitosa de Sousa e as minhas filhas Beatriz Ariadne Feitosa de Sousa e Letícia Feitosa de Sousa. A todos aqueles que contribuíram para esta conquista.

AGRADECIMENTOS

A Deus, que esteve sempre do meu lado nos momentos difíceis.

Aos meus pais, irmãos, minha esposa, minhas filhas e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida acadêmica.

À secretária Livia Lara Lessa Alves pela atenção, carinho e dedicação.

Aos professores colaboradores Prof. Dr. Idalmir de Souza Queiroz Júnior, Prof. Dr. Antônio Ronaldo Gomes Garcia, Profa. Dra. Fabiana Karla de Oliveira Martins Varella e Profa. Dra. Danielle Simone da Silva Casillo pela dedicação e profissionalismo.

Ao orientador e coorientador respectivamente Prof. Dr. Elmer Rolando Llanos Villarreal e Prof. Dr. Alex Sandro de Araújo Silva, pela paciência na orientação e confiança que tornaram possível a conclusão desta dissertação.

Aos amigos e colegas do mestrado, em especial Claudio Marcio Medeiros De Azevedo, Sameque Farias Cunha de Oliveira, Sâmia Lorena de Medeiros Confessor, Tayara Crystina Pereira Benigno e Monique Fernandes da Silva, pelo incentivo e colaboração.

ÉPIGRAFE

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota.”

Madre Teresa de Calcutá

RESUMO

Este trabalho aborda o controle de atitude de um Veículo Aéreo Não Tripulado (VANT) do tipo quadrotor por meio da integração de uma Rede Neural Artificial (RNA) ao controlador clássico Proporcional Integral Derivativo (PID). A natureza não linear da dinâmica desses veículos torna o desenvolvimento de estratégias de controle capazes de garantir estabilidade, robustez e desempenho um dos principais desafios para sua operação autônoma. Nesse contexto, o objetivo desta pesquisa consiste em propor e avaliar um controlador PID-Neural para estabilização de um quadrotor em voo pairado (*hover*), empregando uma estratégia de compensação residual aplicada ao canal de empuxo, de modo a aprimorar a dinâmica vertical sem comprometer a estabilidade dos canais de atitude controlados pelo PID convencional. A metodologia compreendeu a revisão da literatura sobre modelagem e controle de quadrotores, a modelagem matemática da dinâmica da aeronave pelo formalismo de Euler-Lagrange, o desenvolvimento e treinamento de uma Rede Neural Artificial do tipo *Multi-Layer Perceptron* (MLP), a implementação computacional dos controladores PID convencional e PID-Neural em ambiente *MATLAB/Simulink* e a realização de simulações em diferentes cenários operacionais, incluindo variações paramétricas de massa, momento de inércia e amortecimento. Os resultados demonstraram que a estratégia proposta foi capaz de manter a estabilidade do sistema em todos os cenários analisados, preservando o desempenho dos canais de atitude e reduzindo significativamente o erro de altitude quando comparada ao controlador PID convencional. Nos cenários com variação de massa, a redução do erro quadrático médio da altitude atingiu aproximadamente 44%, evidenciando maior robustez frente às incertezas do modelo. Além disso, a contribuição residual da RNA produziu correções suaves no sinal de empuxo, sem introduzir oscilações relevantes nos demais canais de controle. Conclui-se que a estratégia híbrida proposta mostrou-se eficaz para o controle de voo pairado de quadrotor, reunindo a estabilidade do controlador PID com a capacidade adaptativa das Redes Neurais Artificiais, constituindo uma alternativa promissora para aplicações futuras em sistemas de controle de aeronaves não tripuladas.

Palavras-Chave: Quadrotor, Controle PID, Redes Neurais Artificiais, Controle Neural, Voo Pairado.

ABSTRACT

This work addresses the attitude control of a quadrotor Unmanned Aerial Vehicle (UAV) through the integration of an Artificial Neural Network (ANN) with a classical Proportional–Integral–Derivative (PID) controller. The nonlinear dynamics of quadrotor systems make the design of control strategies capable of ensuring stability, robustness, and satisfactory performance one of the main challenges for autonomous flight. In this context, this research proposes and evaluates a PID-Neural controller for quadrotor *hover* stabilization based on a residual neural compensation strategy applied exclusively to the thrust control channel, while maintaining the roll, pitch, and yaw torques under the conventional PID controller. The proposed approach aims to improve the vertical dynamic response without compromising the attitude stabilization provided by the classical controller. The methodology included a literature review on quadrotor modeling and control, mathematical modeling of the vehicle dynamics using the Euler–Lagrange formulation, development and training of a *Multi-Layer Perceptron (MLP)* neural network, computational implementation of the conventional PID and PID-Neural controllers in the *MATLAB/Simulink* environment, and simulation-based evaluation under different operating conditions, including variations in mass, inertia, and damping parameters. The simulation results demonstrated that the proposed strategy maintained system stability in all evaluated scenarios while significantly reducing the altitude tracking error compared with the conventional PID controller. In scenarios involving mass variations, the root mean square error (RMSE) of the altitude was reduced by approximately 44%, demonstrating improved robustness against modeling uncertainties. Furthermore, the residual neural compensation generated smooth corrections in the thrust command without introducing significant oscillations in the attitude control channels. These results indicate that the proposed hybrid strategy successfully combines the robustness and simplicity of the PID controller with the adaptive capability of Artificial Neural Networks, representing a promising approach for quadrotor *hover* control and future autonomous flight applications.

Keywords: Quadrotor, PID Control, Artificial Neural Networks, Neural Control, *Hover* Flight.

LISTA DE FIGURAS

Figura 1	– Evolução da terminologia relacionada a sistemas de aeronaves não tripuladas	26
Figura 2	– Evolução histórica dos quadrotores e principais marcos tecnológicos	27
Figura 3	– Evolução das principais estratégias de controle aplicadas a quadrotores	31
Figura 4	– Sistemas de coordenadas para utilizados na modelagem do quadrotor	35
Figura 5	– Estrutura básica de um controlador PID em malha fechada	43
Figura 6	– Evolução histórica das Redes Neurais Artificiais	50
Figura 7	– Modelo de um neurônio artificial	52
Figura 8	– Arquitetura geral de uma rede neural multicamadas (RNM)	53
Figura 9	– Fluxo simplificado do algoritmo <i>Backpropagation</i>	54
Figura 10	– Principais formas de utilização de Redes Neurais Artificiais em sistemas de controle	55
Figura 11	– Classificação das principais estratégias de controle encontradas na literatura	57
Figura 12	– Ar.Drone 2.0 Parrot	62
Figura 13	– Movimentos do quadrotor obtidos variando as velocidades angulares dos rotores	63
Figura 14	– Referenciais e variáveis cinemáticas do quadrotor	66
Figura 15	– Rotação dos eixos para os ângulos de <i>Tait-Bryan</i>	69
Figura 16	– Diagrama de blocos do modelo dinâmico do quadrotor	74
Figura 17	– Modelo de um controlador Proporcional Integral Derivativo	76
Figura 18	– Estrutura do controlador PID-Neural	81

Figura 19	– Fluxo operacional do controlador PID-Neural	87
Figura 20	– Fluxograma do processo de geração e preparação da base de dados para treinamento da Rede Neural Artificial	89
Figura 21	– Desempenho (<i>Performance</i>) da Rede Neural Artificial durante o treinamento	96
Figura 22	– Estado do treinamento (<i>Training State</i>) da Rede Neural Artificial ..	97
Figura 23	– Regressão (<i>Regression</i>) entre as saídas desejadas e estimadas pela Rede Neural Artificial	98
Figura 24	– Histograma dos erros (<i>Error Histogram</i>) da Rede Neural Artificial	99
Figura 25	– Organização computacional do controlador PID-Neural implementado em <i>MATLAB/Simulink</i>	100
Figura 26	– Estrutura da simulação em malha fechada para voo pairado	102
Figura 27	– Fluxo operacional da simulação do controlador PID-Neural	103
Figura 28	– Modelo reduzido adotado nas simulações em voo pairado	108
Figura 29	– Resposta da altitude durante a estabilização em voo pairado	119
Figura 30	– Resposta do ângulo de rolagem (<i>roll</i>) durante a estabilização em voo pairado	120
Figura 31	– Resposta do ângulo de arfagem (<i>pitch</i>) durante a estabilização em voo pairado	120
Figura 32	– Resposta do ângulo de guinada (<i>yaw</i>) durante a estabilização em voo pairado	121
Figura 33	– Sinal de empuxo gerado durante a estabilização em voo pairado	122
Figura 34	– Resposta da altitude no cenário com massa reduzida em 20%	123
Figura 35	– Resposta da altitude no cenário com massa aumentada em 20%	124
Figura 36	– Resposta do ângulo de rolagem (<i>roll</i>) no cenário combinado	125
Figura 37	– Resposta do ângulo de arfagem (<i>pitch</i>) no cenário combinado	125
Figura 38	– Resposta do ângulo de guinada (<i>yaw</i>) no cenário combinado	126
Figura 39	– Sinal de empuxo no cenário combinado	127
Figura D.1	– Resposta de rolagem (<i>roll</i>) com PID-neural residual aplicado a todos os canais	203

Figura D.2	– Resposta de arfagem (<i>pitch</i>) com PID-neural residual aplicado a todos os canais	204
Figura D.3	– Resposta de arfagem (<i>yaw</i>) com PID-neural residual aplicado a todos os canais	204
Figura D.4	– Resposta de altitude com PID-neural residual aplicado ao empuxo no cenário de inércias aumentadas	205
Figura D.5	– Resposta de altitude com PID-neural residual aplicado ao empuxo no cenário de amortecimento reduzido	206
Figura D.6	– Sinal de empuxo com PID-neural residual aplicado ao empuxo no cenário de massa aumentada em 20%	206

LISTA DE QUADROS

Quadro 1	– Hipóteses adotadas na modelagem matemática do quadrotor	36
Quadro 2	– Comparação entre as principais estratégias clássicas de controle	48
Quadro 3	– Síntese das lacunas identificadas na literatura	58
Quadro 4	– Características da Rede Neural Artificial utilizada no controlador PID-Neural	58
Quadro 5	– Configurações utilizadas no treinamento e retreinamento da Rede Neural Artificial	90
Quadro 6	– Organização dos módulos computacionais	101
Quadro 7	– Organização dos principais arquivos do projeto	104
Quadro 8	– Configurações utilizadas no retreinamento da RNA residual	113
Quadro 9	– Cenários de simulação avaliados na configuração inicial	117
Quadro 10	– Cenários avaliados para o controlador PID-Neural residual	118
Quadro 11	– Critérios de comparação entre o PID e o PID-Neural residual aplicado ao canal de empuxo	128

LISTA DE TABELAS

Tabela 1	– Principais estratégias de controle aplicadas a quadrotores	33
Tabela 2	– Influência qualitativa dos ganhos do controlador PID	45
Tabela 3	– Parâmetros do modelo de quadrotor	75
Tabela 4	– Parâmetros físicos utilizados na simulação	109
Tabela 5	– Variações paramétricas utilizadas nos testes complementares	110
Tabela 6	– Condições iniciais padrão da simulação	110
Tabela 7	– Referências de <i>hover</i> usadas na simulação	111
Tabela 8	– Configurações numéricas da simulação	112
Tabela 9	– Ganhos utilizados no controlador PID	115
Tabela 10	– Síntese dos resultados de estabilização em voo pairado	122
Tabela 11	– Indicadores quantitativos do PID e do PID-neural residual aplicado ao canal de empuxo	130
Tabela D.1	– Indicadores da comparação inicial entre PID e PID-Neural	203
Tabela D.2	– Indicadores completos do PID-Neural residual aplicado ao canal de empuxo	209
Tabela D.3	– Cenários avaliados para o controlador PID-Neural residual	210
Tabela D.4	– Variações paramétricas utilizadas nos testes complementares	211

LISTA DE ABREVIATURAS E SIGLAS

AVIBRAS	Empresa Brasileira de Engenharia Aero Espacial
CG	Centro de gravidade
DCTA	Departamento de Ciência e Tecnologia Aeroespacial
DoD	<i>Department of Defense</i>
LQR	<i>Linear Quadratic Regulator</i>
<i>MATLAB</i>	Software interativo de alto desempenho voltado para o cálculo numérico
MEMS	MEMS (<i>Micro-Electro-Mechanical Systems</i>)
MLP	<i>Multi-Layer Perceptron</i> (Perceptron Multicamadas)
NA	Neurônio Artificial
NED	<i>North-East-Down</i>
PID	Proporcional, Integral e Derivativo.
PPGSCA	Programa de Pós-Graduação em Sistemas de Comunicação e Automação
RNA	Rede Neural Artificial
RNM	Rede Neural Multicamadas
SIMULINK	Ferramenta do <i>MATLAB</i> utilizada para modelagem, simulação e análise de sistemas dinâmicos.
UAS	<i>Unmanned Aircraft Systems</i>
UAV	<i>Unmanned Aerial Vehicles</i>
UFERSA	Universidade Federal Rural do Semi-árido
VANT	Veículo Aéreo Não-Tripulado

LISTA DE SÍMBOLOS

Letras latinas

Símbolo	Descrição	Unidade
b	Coeficiente de arrasto aerodinâmico	$\text{N} \cdot \text{m} \cdot \text{s}^2 \cdot \text{rad}^{-2}$
d	Coeficiente de arrasto	$\text{N} \cdot \text{m} \cdot \text{s}^2 \cdot \text{rad}^{-2}$
$e(t)$	Erro de controle	—
F_i	Empuxo produzido pelo rotor i	N
$f(\cdot)$	Função de ativação	—
g	Aceleração da gravidade	$\text{m} \cdot \text{s}^{-2}$
h_j	Saída do neurônio oculto	—
I_{xx}	Momento de inércia em torno do eixo x	$\text{kg} \cdot \text{m}^2$
I_{yy}	Momento de inércia em torno do eixo y	$\text{kg} \cdot \text{m}^2$
I_{zz}	Momento de inércia em torno do eixo z	$\text{kg} \cdot \text{m}^2$
J_r	Momento de inércia do rotor	$\text{kg} \cdot \text{m}^2$
k_d	Ganho derivativo	—
k_i	Ganho integral	—
k_p	Ganho proporcional	—
L	Lagrangeano do sistema	J
l	Distância entre o centro de massa e cada rotor	m
m	Massa do quadrotor	kg
p	Velocidade angular de rolagem	$\text{rad} \cdot \text{s}^{-1}$
q	Velocidade angular de arfagem	$\text{rad} \cdot \text{s}^{-1}$
r	Velocidade angular de guinada	$\text{rad} \cdot \text{s}^{-1}$
R	Matriz de rotação	—
s_j	Potencial de ativação do neurônio	—

T	Energia cinética	J
u, v, w	Velocidades lineares do corpo	$\text{m}\cdot\text{s}^{-1}$
$u(t)$	Sinal de controle	—
u_1	Empuxo total	N
u_2	Torque de rolagem	$\text{N}\cdot\text{m}$
u_3	Torque de arfagem	$\text{N}\cdot\text{m}$
u_4	Torque de guinada	$\text{N}\cdot\text{m}$
V	Energia potencial	J
w_{ji}	Peso sináptico entre neurônios	—
x, y, z	Coordenadas cartesianas	m
x_i	Entrada da Rede Neural Artificial	—
$y(t)$	Saída do sistema	—

Letras gregas

Símbolo	Descrição	Unidade
α	Taxa de aprendizagem	—
ϕ	Ângulo de rolagem	rad
θ	Ângulo de arfagem	rad
ψ	Ângulo de guinada	rad
τ_x	Torque em torno do eixo x	$\text{N}\cdot\text{m}$
τ_y	Torque em torno do eixo y	$\text{N}\cdot\text{m}$
τ_z	Torque em torno do eixo z	$\text{N}\cdot\text{m}$
ω_1	Velocidade angular do rotor 1	$\text{rad}\cdot\text{s}^{-1}$
ω_2	Velocidade angular do rotor 2	$\text{rad}\cdot\text{s}^{-1}$
ω_3	Velocidade angular do rotor 3	$\text{rad}\cdot\text{s}^{-1}$
ω_4	Velocidade angular do rotor 4	$\text{rad}\cdot\text{s}^{-1}$

Símbolos específicos da Rede Neural Artificial

Símbolo	Descrição	Unidade
b_j	Viés (bias) do neurônio oculto	—
b_k	Viés (bias) do neurônio de saída	—
h_j	Saída da camada oculta	—
v_{kj}	Peso sináptico da camada de saída	—
$\hat{y}_k$	Saída estimada da Rede Neural Artificial	—

Operadores e notação matemática

Símbolo	Descrição	Unidade
$(\cdot)^T$	Transposta	—
$(\dot{\cdot})$	Derivada temporal de primeira ordem	—
$(\ddot{\cdot})$	Derivada temporal de segunda ordem	—
Σ	Somatório	—
$\widehat{(\cdot)}$	Valor estimado	—

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

1 INTRODUÇÃO

1.1	Motivação	24
1.2	Definição de VANT	25
1.3	História dos quadrotores	26
1.4	Objetivos	28
1.4.1	Geral	28
1.4.2	Específicos	28
1.5	Contribuições da pesquisa	29
1.6	Organização da dissertação	29

2 FUNDAMENTAÇÃO TEÓRICA

2.1	Estado da arte em controle de quadrotores	30
2.2	Modelagem matemática do quadrotor	33
2.2.1	Sistemas de coordenadas	34
2.2.2	Hipóteses adotadas na modelagem	36
2.2.3	Equações de movimento	37
2.2.3.1	Dinâmica translacional	37
2.2.3.2	Dinâmica rotacional	38
2.2.3.3	Entradas de controle	39
2.2.3.4	Relação entre rotores e entradas	39
2.2.4	Modelo em espaço de estados	40
2.2.4.1	Modelo utilizado na implementação	41
2.2.5	Modelo reduzido para voo pairado	41
2.3	Estratégias clássicas de controle	42
2.3.1	Controlador Proporcional–Integral–Derivativo (PID)	43

2.3.1.1	Ação proporcional	44
2.3.1.2	Ação integral	45
2.3.1.3	Ação derivativa	45
2.3.1.4	Influência dos ganhos do controlador	45
2.3.1.5	Aplicação ao controle de quadrotores	46
2.3.2	Outras estratégias clássicas de controle	46
2.4	Redes neurais artificiais	48
2.4.1	Evolução histórica das redes neurais artificiais	49
2.4.2	Neurônio artificial e <i>Perceptron</i>	50
2.4.3	Redes neurais multicamadas (RNM)	53
2.4.4	Algoritmo <i>Backpropagation</i>	53
2.4.5	Redes neurais artificiais aplicadas ao controle	55
2.4.6	Redes neurais artificiais aplicadas ao controle de quadrotores	56
2.4.7	Considerações finais	58
2.5	Trabalhos relacionados	59
2.5.1	Introdução	59
2.5.2	Primeira categoria	59
2.5.3	Segunda categoria	59
2.5.4	Terceira categoria	60
2.5.5	Lacuna	60
2.5.6	Encerramento	60
2.6	Considerações finais	61
3	MODELAGEM DO QUADROTOR	
3.1	O modelo de quadrotor.....	62
3.2	Princípio de funcionamento	62
3.3	Modelagem matemática	64
3.3.1	Referenciais e variáveis da modelagem	65
3.3.2	Modelagem Cinemática	66
3.3.3	Modelagem Dinâmica	70
3.4	Parâmetros do quadrotor	74

4	PROJETO DO CONTROLADOR PID-NEURAL	
4.1	Controlador PID	76
4.2	Controlador PID em tempo discreto	78
4.3	Projeto do controlador PID-Neural	79
4.4	Equações de adaptação do PID-Neural	82
4.5	Estrutura interna do controlador PID-Neural	84
4.5.1	Variáveis de entrada da rede neural	84
4.5.2	Arquitetura da rede neural artificial	85
4.5.3	Cálculo da saída da rede neural	86
4.5.4	Integração entre a RNA e o controlador PID	87
4.6	Treinamento e Validação da RNA	88
4.6.1	Geração da base de dados	88
4.6.2	Configuração do treinamento	89
4.6.3	Processo de treinamento	91
4.6.4	Validação da rede neural	95
4.7	Estrutura computacional do sistema de controle	100
4.7.1	Organização da malha fechada	101
4.7.2	Fluxo operacional da simulação	102
4.7.3	Organização dos arquivos <i>MATLAB</i>	104
5	RESULTADOS DE SIMULAÇÃO	
5.1	Considerações iniciais	105
5.2	Ambiente de simulação	106
5.2.1	Modelo reduzido para simulação em voo pairado	107
5.2.2	Parâmetros físicos e condições iniciais	108
5.2.3	Configurações numéricas da simulação	111
5.3	Estratégia de controle avaliada	113
5.4	Cenários de simulação	116
5.5	Resultados da estabilização em voo pairado	118
5.6	Comparação entre PID convencional e PID-Neural	127
5.7	Análise quantitativa dos resultados	129

5.8	Discussão dos resultados	131
5.9	Limitações das simulações	133
5.10	Considerações finais do capítulo	134
6	CONCLUSÕES E TRABALHOS FUTUROS	
6.1.	Conclusões	135
6.2.	Trabalhos futuros	136
	REFERÊNCIAS BIBLIOGRÁFICAS	138
	APÊNDICE	
	APÊNDICE A - SCRIPTS DA SIMULAÇÃO E DOS CONTROLADORES	
A.1	Script de parâmetros físicos do quadrotor	140
A.2	Script de referências de voo pairado	141
A.3	Script da planta reduzida em <i>hover</i>	141
A.4	Script principal de simulação	143
A.5	Scripts dos controladores	145
A.5.1	Controlador PID Convencional/Adaptativo	145
A.5.2	Controlador PID-Neural original	148
A.5.3	Controlador PID-Neural residual aplicado ao canal de empuxo	150
	APÊNDICE B - TREINAMENTO E RETREINAMENTO DA REDE NEURAL	
B.1	Geração da base de dados	159
B.1.1	Geração do dataset	159
B.1.2	Pré-processamento do dataset	162
B.1.3	Análise do dataset	163
B.2	Configuração da rede neural	166
B.3	Retreinamento da RNA residual	167
B.4	Formulação final do PID-Neural residual	176
	APÊNDICE C - GERAÇÃO DE FIGURAS, TABELAS E MÉTRICAS	
C.1	Script de geração do diagrama de blocos	178
C.2	Scripts de geração das figuras de resposta temporal	181
C.3	Scripts de geração das tabelas	189
C.4	Cálculo das métricas quantitativas	200

APÊNDICE D - RESULTADOS COMPLEMENTARES

D.1 Comparação Inicial entre PID e PID-Neural	202
D.2 PID-Neural residual aplicado a todos os canais	202
D.3 PID-Neural residual aplicado apenas ao canal de empuxo	205
D.4 Tabelas complementares de resultados	207

1 INTRODUÇÃO

Neste capítulo apresenta-se o tema da elaboração desta pesquisa, o que motivou a pesquisa, o objetivo geral, objetivos específicos da pesquisa e a definição dos principais conceitos necessários à compreensão do trabalho. Será apresentada também uma pequena introdução sobre Veículos Aéreos Não Tripulados (VANT) e um breve histórico sobre quadrotores. Além disso, é apresentada a organização e estrutura que norteará este trabalho.

1.1 Motivação

O desenvolvimento dos Veículos Aéreos Não Tripulados (VANTs) intensificou-se nas últimas décadas, impulsionado por aplicações em monitoramento ambiental, inspeção de infraestrutura, agricultura de precisão, segurança pública, defesa e logística. Entre as diferentes configurações, os quadrotores destacam-se pela elevada capacidade de manobra, decolagem e pouso vertical (VTOL) e facilidade de operação em ambientes confinados.

A versatilidade dessas aeronaves impulsionou o crescimento das pesquisas relacionadas à modelagem dinâmica e ao desenvolvimento de estratégias de controle. Devido ao forte acoplamento entre os movimentos translacionais e rotacionais e à presença de não linearidades, o controle de quadrotor é reconhecido como um problema clássico de controle não linear.

Além da elevada manobrabilidade, os quadrotores apresentam vantagens operacionais em missões que exigem voo pairado, acompanhamento de alvos e aquisição contínua de imagens por sensores embarcados. Essas características explicam sua ampla utilização em atividades de vigilância, inspeção e monitoramento.

Entretanto, a complexidade de sua dinâmica exige controladores capazes de garantir estabilidade, robustez e desempenho mesmo na presença de incertezas e perturbações externas. Nesse contexto, modelos computacionais e ambientes de simulação tornam-se ferramentas essenciais para o desenvolvimento, a avaliação e a validação de novas estratégias de controle.

Diversas abordagens têm sido propostas para esse propósito, abrangendo desde controladores clássicos, como o PID, até técnicas robustas, adaptativas e baseadas em Inteligência Artificial. Entre essas estratégias, destaca-se o controlador PID-Neural, que combina a robustez do controlador convencional com a capacidade de aprendizagem e aproximação de funções não lineares proporcionada pelas Redes Neurais Artificiais.

Diante desse cenário, o desenvolvimento de estratégias capazes de aumentar a estabilidade e a robustez de quadrotores permanece como um importante desafio científico e tecnológico. É nesse contexto que se insere a proposta desta dissertação, voltada ao desenvolvimento de um controlador PID-Neural para estabilização de um quadrotor em voo pairado.

1.2 Definição de VANT

O emprego de aeronaves não tripuladas tem crescido continuamente nas últimas décadas em decorrência dos avanços da eletrônica embarcada, dos sistemas de navegação, dos sensores inerciais e das telecomunicações. Esse desenvolvimento ampliou significativamente suas aplicações civis, científicas e militares, incluindo monitoramento ambiental, agricultura de precisão, inspeção de estruturas, mapeamento, segurança pública e defesa.

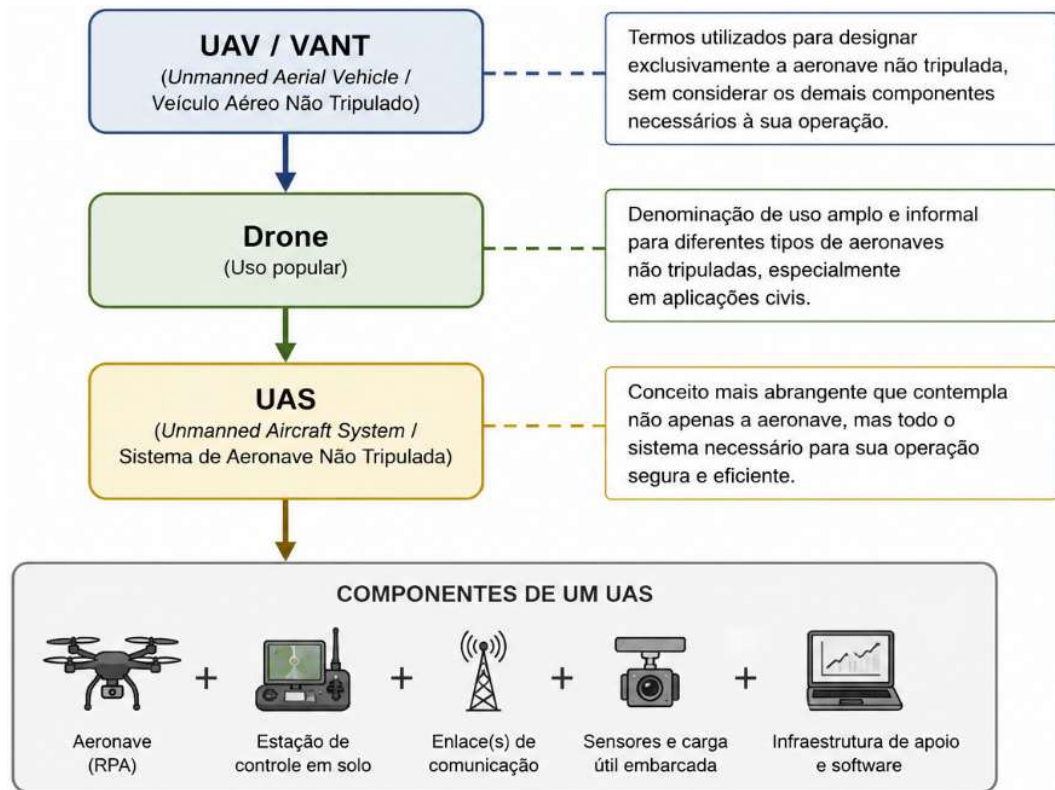
A expansão dessas plataformas também impulsionou a padronização da terminologia adotada na literatura. Entre as definições mais difundidas destaca-se a do Department of Defense (DoD), que caracteriza o Veículo Aéreo Não Tripulado (VANT) como uma aeronave motorizada operando sem piloto embarcado, capaz de realizar voos autônomos ou remotamente pilotados e transportar diferentes tipos de carga útil (UNITED STATES, 2005).

A evolução tecnológica também promoveu mudanças na nomenclatura utilizada para designar esses sistemas. Inicialmente, os termos Unmanned Aerial Vehicle (UAV) e Veículo Aéreo Não Tripulado (VANT) referiam-se exclusivamente à aeronave. Com sua popularização, a denominação drone passou a ser amplamente empregada, principalmente em aplicações civis.

Posteriormente, consolidou-se o termo Unmanned Aircraft System (UAS), que representa uma abordagem mais abrangente ao incluir, além da aeronave, a estação de controle em solo, os enlaces de comunicação, os sensores embarcados e toda a infraestrutura necessária à operação do sistema.

A evolução dessa terminologia é apresentada na Figura 1, que sintetiza a transição de uma visão centrada na aeronave para o conceito sistêmico atualmente adotado na literatura especializada.

Figura 1 – Evolução da terminologia relacionada a sistemas de aeronaves não tripuladas.



Fonte: Elaborada pelo Autor (2014).

Embora UAS seja atualmente a terminologia mais abrangente, optou-se por utilizar, nesta dissertação, a expressão Veículo Aéreo Não Tripulado (VANT), por ser predominante nas referências nacionais e por representar adequadamente o objeto deste estudo, centrado na modelagem e no controle de um quadrotor.

Independentemente da nomenclatura adotada, essas aeronaves dependem de sistemas de controle para garantir estabilidade, segurança e desempenho durante o voo. Entre as diferentes configurações, os veículos de asas rotativas destacam-se pela elevada manobrabilidade e pela capacidade de decolagem e pouso vertical, características que justificam sua ampla utilização em pesquisas científicas.

1.3 História dos quadrotores

A configuração de aeronaves com quatro rotores distribuídos simetricamente surgiu no início do século XX como uma alternativa para viabilizar o voo vertical controlado. Desde os primeiros experimentos, essa arquitetura demonstrou a capacidade de compensar

naturalmente os torques gerados pelos rotores, princípio que permanece presente nos quadrotores atuais.

O primeiro quadrotor descrito na literatura foi desenvolvido em 1907 por Louis e Jacques Bréguet, em parceria com Charles Richet. Denominado *Bréguet-Richet Quad-Rotor Helicopter*, o protótipo realizou pequenos voos experimentais, porém apresentava baixa estabilidade e dependia do auxílio de operadores para limitar seus movimentos laterais (LEISHMAN, 2002).

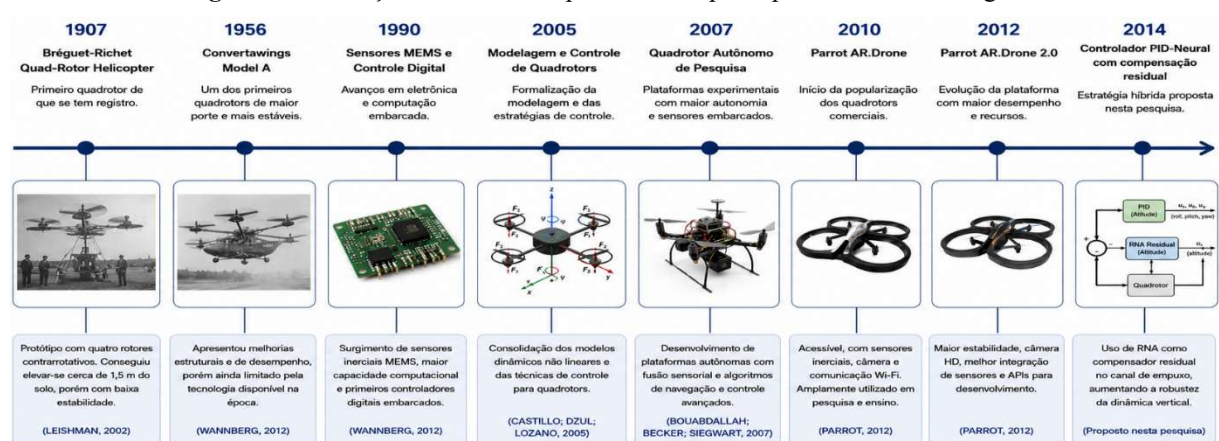
Nas décadas seguintes, diversos protótipos buscaram superar as limitações tecnológicas da época. Entretanto, a baixa potência dos motores, a ausência de sensores embarcados e a inexistência de sistemas eletrônicos de controle restringiram sua aplicação prática durante grande parte do século XX (WANNBERG, 2012).

A partir da década de 1990, o desenvolvimento de sensores MEMS, sistemas embarcados e algoritmos digitais de controle impulsionou a evolução dos quadrotores. Como resultado, surgiram plataformas mais leves, compactas e confiáveis, ampliando sua utilização em pesquisas sobre controle, robótica aérea e navegação autônoma (CASTILLO; DZUL; LOZANO, 2005).

Nesse contexto, destacam-se o Parrot AR.Drone, lançado em 2010, e sua evolução, o AR.Drone 2.0, disponibilizado em 2012. Equipadas com sensores inerciais, câmeras e comunicação sem fio, essas plataformas passaram a ser amplamente empregadas em pesquisas sobre modelagem e controle de quadrotor (PARROT, 2012).

A Figura 2 apresenta uma síntese da evolução histórica dos quadrotores, destacando os principais avanços tecnológicos responsáveis por sua consolidação como plataforma experimental de pesquisa.

Figura 2 – Evolução histórica dos quadrotores e principais marcos tecnológicos.



Fonte: Adaptado pelo autor a partir de Leishman (2002), Castillo, Dzul e Lozano (2005), Bouabdallah, Becker e Siegwart (2007), Wannberg (2012) e Parrot (2012).

Observa-se que a evolução dos quadrotores foi fortemente influenciada pelo desenvolvimento dos sistemas eletrônicos embarcados e das técnicas de controle. Nesse contexto, o aperfeiçoamento de estratégias capazes de aumentar a robustez e o desempenho dessas aeronaves permanece como um dos principais desafios da área, motivando o desenvolvimento do controlador PID-Neural proposto nesta dissertação.

1.4 Objetivos

1.4.1 Geral

O objetivo principal deste trabalho é desenvolver um controlador PID-Neural para estabilização de um quadrotor em voo pairado por meio de uma estratégia de compensação residual da Rede Neural Artificial aplicada ao canal de empuxo, visando aumentar a robustez da dinâmica vertical sem comprometer a estabilidade dos canais de atitude controlados pelo PID convencional.

1.4.2 Específicos

- ✓ Modelar matematicamente a dinâmica do quadrotor, considerando os movimentos translacionais e rotacionais associados aos eixos do sistema de coordenadas, de modo a obter uma representação adequada para simulação do voo pairado.
- ✓ Implementar, em ambiente *MATLAB/Simulink*, o modelo dinâmico do quadrotor e o sistema de controle PID, permitindo a análise do comportamento do veículo em condição de *hover*.
- ✓ Desenvolver e integrar uma RNA ao controlador PID, de forma que a RNA atue no ajuste ou adaptação do controle aplicado à estabilização do quadrotor.
- ✓ Analisar, por meio de simulações, a resposta dinâmica dos ângulos de Tait-Bryan, especialmente rolagem, arfagem e guinada, verificando sua influência na estabilidade do quadrotor em voo pairado.
- ✓ Avaliar o desempenho do controlador PID-Neural em comparação ao controlador PID convencional, considerando critérios como estabilidade, tempo de acomodação, sobressinal, redução de oscilações e manutenção da condição de *hover*.

1.5 Contribuições da pesquisa

As principais contribuições desta pesquisa são:

- ✓ desenvolvimento de um controlador PID-Neural híbrido;
- ✓ utilização da RNA como compensador residual apenas no canal de empuxo;
- ✓ preservação da estabilidade angular;
- ✓ validação em cinco cenários distintos;
- ✓ análise quantitativa baseada em RMSE, MAE e esforço de controle.

A principal contribuição desta pesquisa consiste na proposição de uma estratégia de integração entre um controlador PID convencional e uma Rede Neural Artificial, na qual a contribuição neural atua como compensador residual exclusivamente sobre o canal de empuxo do quadrotor.

Essa abordagem busca aumentar a robustez da dinâmica vertical frente às incertezas do modelo, preservando simultaneamente a estabilidade dos canais responsáveis pelo controle da atitude da aeronave.

1.6 Organização da dissertação

Esta dissertação está organizada em seis capítulos, estruturados de forma a conduzir o leitor desde a contextualização do problema até a análise dos resultados obtidos com a estratégia de controle proposta.

O Capítulo 1 apresenta a introdução da pesquisa, contemplando a motivação, a contextualização dos Veículos Aéreos Não Tripulados (VANTs) e dos quadrotores, os objetivos gerais e específicos, as contribuições da pesquisa e a organização deste trabalho.

O Capítulo 2 apresenta a fundamentação teórica, abordando o estado da arte em controle de quadrotores, a modelagem matemática dessas aeronaves, as principais estratégias clássicas de controle, os fundamentos das Redes Neurais Artificiais e os trabalhos relacionados que embasam a proposta desenvolvida nesta dissertação.

O Capítulo 3 descreve a modelagem matemática do quadrotor, apresentando o princípio de funcionamento da aeronave, os sistemas de coordenadas utilizados, as hipóteses adotadas, a formulação cinemática e dinâmica baseada no formalismo de Euler-Lagrange e os parâmetros empregados nas simulações.

O Capítulo 4 apresenta o desenvolvimento do controlador PID-Neural proposto, incluindo a formulação do controlador PID discreto, a arquitetura da Rede Neural Artificial,

sua integração ao controlador clássico, o processo de geração da base de dados, o treinamento, a validação da rede e a organização computacional da implementação em ambiente *MATLAB/Simulink*.

O Capítulo 5 apresenta os resultados das simulações realizadas em voo pairado, descrevendo os cenários de avaliação, a estratégia de compensação residual aplicada ao canal de empuxo, a comparação entre o controlador PID convencional e o controlador PID-Neural, bem como a análise quantitativa dos indicadores de desempenho obtidos.

Por fim, o Capítulo 6 reúne as conclusões da pesquisa, destacando as principais contribuições alcançadas, as limitações identificadas durante o desenvolvimento do trabalho e as sugestões para pesquisas futuras relacionadas ao controle inteligente de quadrotor.

Diante desse contexto, esta pesquisa busca responder à seguinte questão de pesquisa: a utilização de uma estratégia de compensação residual baseada em Rede Neural Artificial, integrada a um controlador PID convencional, é capaz de melhorar o desempenho da estabilização de um quadrotor em voo pairado sem comprometer a estabilidade dos canais de atitude?

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta um histórico da literatura referente às pesquisas sobre VANTs, em especial os quadrotores, redes neurais artificiais e controlador clássico PID. Define também os principais conceitos relevantes para implementação do controlador PID-Neural, tais como: modelo do quadrotor; o princípio de funcionamento, com ênfase nos movimentos de elevação, rolagem, arfagem e guinada assim como a justificativa para escolha da técnica de modelagem.

Apresentar a evolução científica das pesquisas em controle de quadrotores, destacando as principais estratégias desenvolvidas na literatura e justificando a proposta desta dissertação.

2.1 Estado da arte em controle de quadrotores

O controle de quadrotor constitui um dos principais desafios da área de Sistemas de Controle devido à natureza fortemente não linear da dinâmica dessas aeronaves. Embora sua configuração mecânica seja relativamente simples, a interação entre os movimentos translacionais e rotacionais, associada à presença de acoplamentos dinâmicos e incertezas

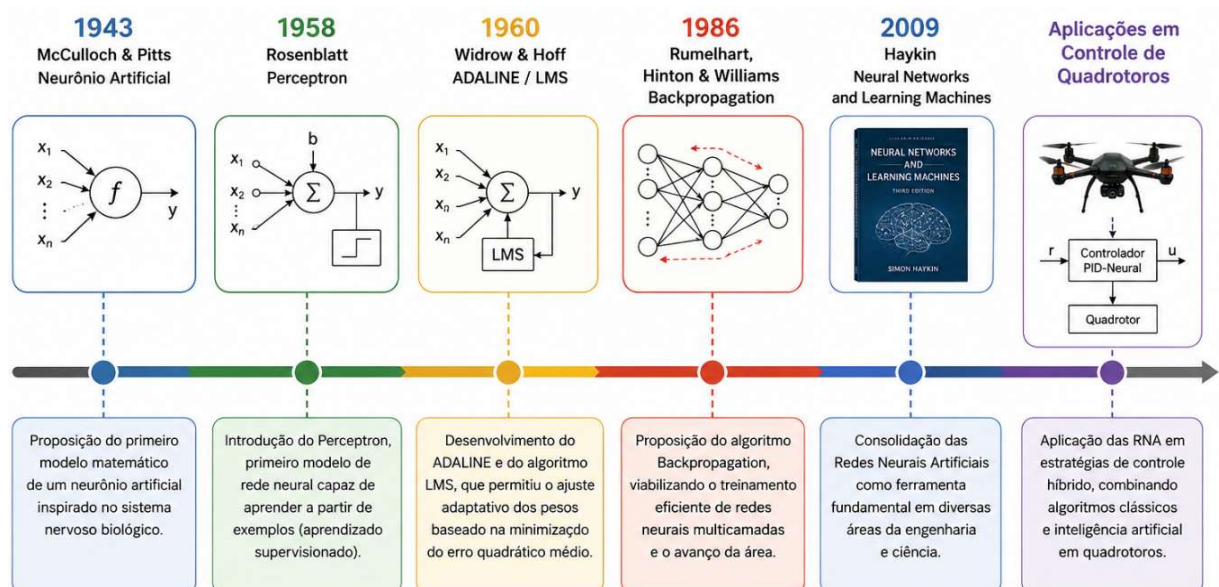
paramétricas, exige estratégias capazes de garantir estabilidade e desempenho em diferentes condições de operação.

Nas últimas décadas, o avanço dos sensores inerciais, dos sistemas embarcados e da capacidade de processamento dos microcontroladores impulsionou o desenvolvimento de novas técnicas de controle. Como consequência, os quadrotoros passaram a ser utilizados não apenas em aplicações comerciais e militares, mas também como plataformas experimentais para pesquisas em robótica aérea, navegação autônoma e sistemas inteligentes.

Esse cenário favoreceu o surgimento de diferentes estratégias para modelagem e controle dessas aeronaves. Inicialmente, os estudos concentraram-se na obtenção de modelos matemáticos capazes de representar adequadamente a dinâmica do sistema. Posteriormente, as pesquisas passaram a investigar métodos capazes de lidar com as não linearidades do modelo e com as perturbações inerentes ao voo.

A evolução dessas estratégias é apresentada na Figura 3, que sintetiza os principais marcos científicos relacionados ao desenvolvimento dos métodos de controle aplicados a quadrotoros. Observa-se que o avanço das técnicas de controle ocorreu de forma paralela ao desenvolvimento dos sistemas eletrônicos embarcados e ao aumento da capacidade computacional disponível para essas plataformas.

Figura 3 – Evolução das principais estratégias de controle aplicadas a quadrotoros.



Fonte: Adaptado pelo autor a partir de Castillo, Dzul e Lozano (2005), Bouabdallah, Becker e Siegwart (2007) e Brandão et al. (2012).

Entre os primeiros trabalhos de referência, destaca-se a pesquisa desenvolvida por Castillo, Dzul e Lozano (2005), que apresentou uma sistematização da modelagem dinâmica de mini-helicópteros e quadrotores. Essa contribuição estabeleceu fundamentos importantes para pesquisas posteriores, especialmente aquelas voltadas ao desenvolvimento de controladores baseados em modelos matemáticos.

Posteriormente, Bouabdallah, Becker e Siegwart (2007) demonstraram que estratégias clássicas de controle, associadas a técnicas de linearização, poderiam proporcionar desempenho satisfatório em condições próximas ao voo pairado. Os resultados obtidos evidenciaram que controladores relativamente simples permanecem eficientes quando corretamente ajustados ao ponto de operação do sistema.

No contexto nacional, destacam-se os estudos de Brandão et al. (2012), que apresentaram uma modelagem não linear subatuada aplicada ao quadrotor AR.Drone. Além de ampliar a compreensão da dinâmica dessa plataforma, esses trabalhos forneceram uma importante referência para pesquisas brasileiras relacionadas ao desenvolvimento de sistemas de controle para veículos aéreos não tripulados.

Com a evolução da capacidade computacional dos sistemas embarcados, técnicas baseadas em inteligência artificial passaram a receber maior atenção da comunidade científica. Redes Neurais Artificiais, lógica fuzzy e métodos adaptativos passaram a ser investigados como alternativas para compensar limitações dos modelos matemáticos convencionais, principalmente em situações caracterizadas por incertezas paramétricas e perturbações externas (HAYKIN, 2009).

Apesar dos avanços observados na literatura, verifica-se que grande parte das estratégias propostas busca substituir completamente o controlador clássico ou adaptar simultaneamente todos os canais de controle do quadrotor. Embora essas abordagens apresentem resultados expressivos, elas normalmente aumentam a complexidade computacional do controlador e tornam o processo de ajuste dos parâmetros mais dependente da etapa de treinamento da rede neural.

Sob essa perspectiva, esta pesquisa adota uma estratégia distinta das abordagens tradicionalmente encontradas na literatura. Em vez de substituir o controlador PID ou aplicar a compensação neural a todos os canais de controle, propõe-se utilizar a Rede Neural Artificial exclusivamente como um compensador residual aplicado ao canal de empuxo. Essa estratégia procura preservar a estabilidade proporcionada pelo controlador PID nos canais de atitude e, simultaneamente, explorar a capacidade de aproximação não linear da RNA para reduzir os erros associados à dinâmica vertical.

Com base nos trabalhos anteriormente discutidos, a Tabela 1 sintetiza as principais estratégias de controle aplicadas a quadrotoros, destacando suas contribuições, vantagens e limitações em relação ao problema de estabilização dessas aeronaves.

Tabela 1: Principais estratégias de controle aplicadas a quadrotoros.

Autor	Ano	Principais vantagens	Principal contribuição	Principais limitações
Castillo, Dzul e Lozano	2005	Controle não linear.	Modelagem dinâmica completa do quadrotor.	Elevada complexidade matemática.
Bouabdallah, Becker e Siegwart	2007	PID, <i>Linear Quadratic Regulator</i> (LQR) e Linearização.	Estabilização de atitude.	Dependência do modelo nominal.
Brandão et al.	2012	PID aplicado ao AR.Drone.	Modelagem não linear subatuada.	Não contempla adaptação online.
Cui e Shin	1992	Controle Neural Adaptativo.	Atualização online dos pesos neurais.	Aplicação genérica, não específica para quadrotoros.
Trabalho proposto.	2014	PID-Neural com compensação residual.	Compensação neural aplicada exclusivamente ao canal de empuxo.	Restrito ao voo pairado.

Fonte: Adaptado pelo autor a partir de Castillo, Dzul e Lozano (2005), Bouabdallah, Becker e Siegwart (2007), Cui e Shin (1992) e Brandão et al. (2012).

A análise comparativa apresentada na Tabela 1 evidencia que ainda existe espaço para estratégias híbridas capazes de combinar a simplicidade e a robustez do controlador PID com a capacidade adaptativa das Redes Neurais Artificiais. Nesse contexto, a arquitetura proposta nesta dissertação busca explorar essas características por meio de uma compensação neural residual aplicada exclusivamente ao canal de empuxo, preservando os torques de rolagem, arfagem e guinada sob responsabilidade do controlador PID convencional.

Dessa forma, antes de apresentar a arquitetura do controlador desenvolvido nesta pesquisa, torna-se necessário compreender a modelagem matemática utilizada para representar a dinâmica do quadrotor. Esses fundamentos são apresentados na próxima seção.

2.2 Modelagem matemática do quadrotor

Este tópico de como objetivo apresentar os fundamentos matemáticos empregados para representar a dinâmica do quadrotor, estabelecendo o modelo utilizado no desenvolvimento e validação do controlador PID-Neural proposto.

A modelagem matemática constitui uma das etapas fundamentais no desenvolvimento de sistemas de controle para veículos aéreos não tripulados, pois permite descrever

quantitativamente a dinâmica da aeronave e prever seu comportamento frente às ações de controle. No caso dos quadrotores, essa modelagem apresenta elevado grau de complexidade devido à natureza não linear, multivariável e subatuada do sistema, bem como ao forte acoplamento existente entre os movimentos translacionais e rotacionais.

Diversos modelos foram propostos na literatura para representar a dinâmica de quadrotores, variando desde modelos lineares simplificados até formulações completas baseadas nas equações de Newton-Euler ou na Mecânica de Lagrange. Independentemente da abordagem adotada, todos procuram representar adequadamente as forças e os momentos gerados pelos rotores, permitindo o desenvolvimento de controladores capazes de estabilizar o veículo em diferentes condições de voo (CASTILLO; DZUL; LOZANO, 2005; BOUABDALLAH; BECKER; SIEGWART, 2007).

Nesta dissertação foi adotado o modelo dinâmico baseado nas equações de Newton-Euler, amplamente empregado na literatura por apresentar boa precisão na representação dos movimentos de voo pairado e permitir implementação computacional relativamente simples. A formulação considera o quadrotor como um corpo rígido submetido às forças aerodinâmicas produzidas pelos quatro rotores e aos efeitos gravitacionais atuantes sobre a aeronave.

Antes da apresentação das equações diferenciais que descrevem a dinâmica do sistema, faz-se necessária a definição dos sistemas de coordenadas utilizados para representar a posição, a orientação e os movimentos do quadrotor.

A Subseção 2.2.1 apresenta os sistemas de coordenadas adotados nesta pesquisa e as convenções utilizadas para descrição dos movimentos translacionais e rotacionais da aeronave.

2.2.1 Sistemas de coordenadas

A descrição matemática do movimento de um quadrotor requer a definição de sistemas de coordenadas capazes de representar sua posição e orientação no espaço tridimensional. Tradicionalmente, são utilizados dois referenciais ortogonais: um referencial inercial, fixo em relação à Terra, e um referencial solidário ao corpo da aeronave, responsável por descrever sua dinâmica local.

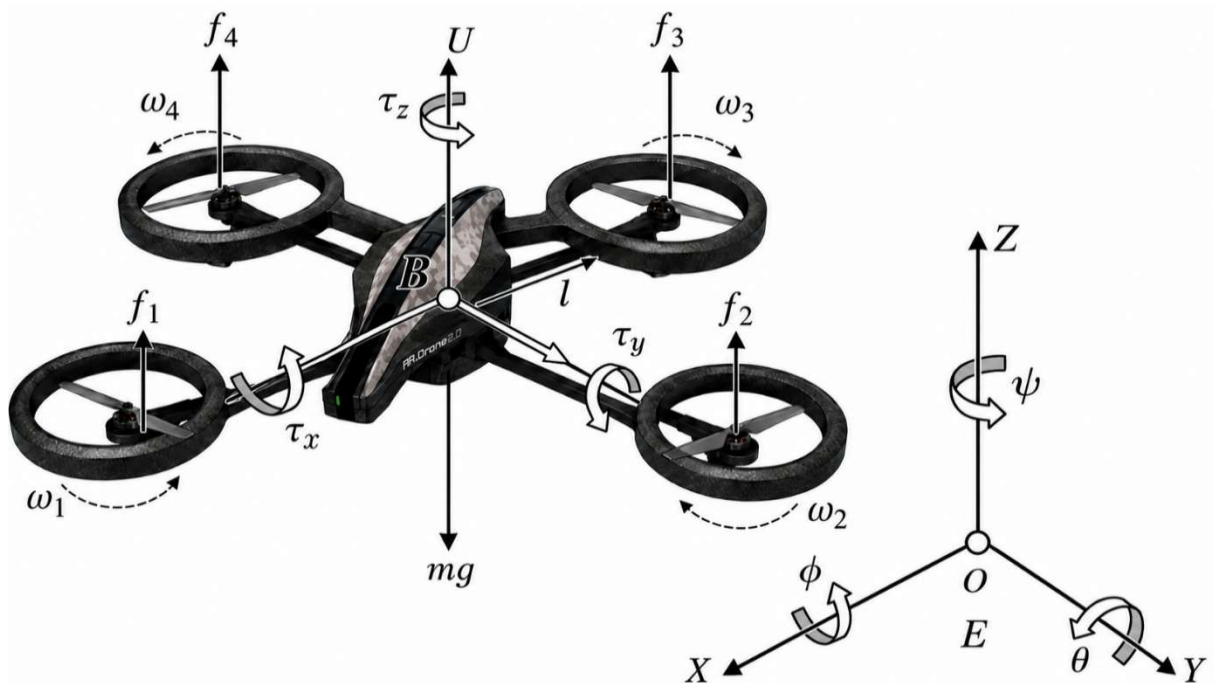
O referencial inercial, também denominado sistema de coordenadas global, é utilizado para representar a posição do centro de massa do veículo por meio das coordenadas cartesianas x , y e z . Já o referencial do corpo movimenta-se juntamente com o quadrotor,

sendo empregado na descrição das velocidades lineares, velocidades angulares, forças e torques atuantes na estrutura da aeronave.

A orientação do quadrotor em relação ao sistema inercial é descrita pelos ângulos de Euler: rolagem ϕ , arfagem θ e guinada ψ . Esses ângulos definem completamente a atitude da aeronave e são amplamente utilizados na modelagem de sistemas de controle devido à sua interpretação física intuitiva.

A Figura 4 apresenta os sistemas de coordenadas utilizados nesta pesquisa, bem como os eixos associados ao referencial inercial e ao referencial solidário ao corpo do quadrotor.

Figura 4 – Sistemas de coordenadas para utilizados na modelagem do quadrotor.



Fonte: Elaborada pelo Autor (2014).

Observa-se que a utilização simultânea desses dois referenciais permite descrever de forma consistente tanto os deslocamentos da aeronave no espaço quanto sua dinâmica rotacional. A transformação entre os sistemas de coordenadas é realizada por meio de matrizes de rotação construídas a partir dos ângulos de Euler, permitindo relacionar grandezas expressas em ambos os referenciais.

Essas transformações constituem a base para a obtenção das equações diferenciais que descrevem o movimento do quadrotor, apresentadas nas subseções seguintes.

Uma vez definidos os sistemas de referência utilizados na modelagem, torna-se necessário estabelecer as hipóteses adotadas para simplificar a representação matemática da dinâmica do quadrotor. Essas hipóteses são apresentadas na Subseção 2.2.2.

2.2.2 Hipóteses adotadas na modelagem

A obtenção de um modelo matemático capaz de representar adequadamente a dinâmica do quadrotor exige a adoção de hipóteses simplificadoras. Essas hipóteses permitem reduzir a complexidade do sistema sem comprometer a fidelidade necessária às análises desenvolvidas nesta pesquisa.

Considerou-se que o quadrotor possui estrutura rígida e simétrica, com distribuição uniforme de massa em torno do centro de gravidade (CG). Admitiu-se ainda que os rotores são idênticos, produzindo forças de sustentação e torques proporcionais ao quadrado de suas velocidades de rotação, conforme amplamente adotado na literatura (CASTILLO; DZUL; LOZANO, 2005; BOUABDALLAH; BECKER; SIEGWART, 2007).

Também foram desconsiderados efeitos aerodinâmicos de segunda ordem, tais como turbulências, flexibilidade estrutural, interferência entre rotores e deformações mecânicas da estrutura, assumindo condições típicas de voo pairado em baixas velocidades. As principais hipóteses empregadas na modelagem são resumidas no Quadro 1.

Quadro 1: Hipóteses adotadas na modelagem matemática do quadrotor

Hipótese	Justificativa
Corpo rígido	Simplifica a modelagem dinâmica.
Estrutura simétrica	Facilita o cálculo dos momentos de inércia.
Hélices idênticas	Permite uniformidade dos coeficientes aerodinâmicos.
Baixas velocidades	Compatível com voo pairado.
Pequenas perturbações	Aproximação válida para estabilização.

Fonte: Elaborada pelo autor (2014).

Observa-se que essas hipóteses são amplamente empregadas em estudos envolvendo estabilização de quadrotores e fornecem um modelo suficientemente representativo para o desenvolvimento do controlador proposto nesta dissertação.

Com base nos referenciais e nas hipóteses estabelecidas, podem ser deduzidas as equações diferenciais que descrevem o movimento translacional e rotacional do quadrotor, apresentadas na Subseção 2.2.3.

2.2.3 Equações de movimento

O objetivo deste tópico é apresentar as equações que descrevem a dinâmica translacional e rotacional do quadrotor, estabelecendo o modelo matemático utilizado no desenvolvimento do controlador PID-Neural.

A dinâmica de um quadrotor pode ser descrita por meio das equações de Newton-Euler, as quais representam simultaneamente os movimentos translacionais e rotacionais do corpo rígido. Essa abordagem é amplamente utilizada na literatura devido à sua consistência física e à facilidade de implementação computacional, permitindo representar de forma adequada as forças e os momentos gerados pelos quatro rotores da aeronave (CASTILLO; DZUL; LOZANO, 2005; BOUABDALLAH; BECKER; SIEGWART, 2007).

Considerando os referenciais apresentados na Seção 2.2.1 e as hipóteses estabelecidas na Seção 2.2.2, o movimento do quadrotor pode ser decomposto em duas partes independentes: a dinâmica translacional, responsável pelo deslocamento do centro de massa, e a dinâmica rotacional, responsável pela orientação da aeronave.

Inicialmente, a posição do centro de massa do veículo é representada pelas coordenadas cartesianas x , y e z , enquanto sua orientação é descrita pelos ângulos de Euler ϕ , θ e ψ . Dessa forma, o vetor de estados generalizados pode ser escrito como:

$$q(\gamma, \eta) = [x, y, z, \phi, \theta, \psi]^T \in \mathbb{R}^6 \quad (1)$$

Em que $q(\gamma, \eta) \in \mathbb{R}^6$ representa o vetor de coordenadas generalizadas do quadrotor, composto pela posição $\gamma \in \mathbb{R}^3$ do centro de massa em relação ao sistema de referência inercial e pela orientação $\eta \in \mathbb{R}^3$, descrita pelos ângulos de Euler. O vetor de posição é formado pelas coordenadas cartesianas x , y e z , enquanto o vetor de orientação é constituído pelos ângulos de rolagem ϕ , arfagem θ e guinada ψ , os quais definem a atitude do veículo durante o voo.

2.2.3.1 Dinâmica translacional

As forças produzidas pelos quatro rotores são responsáveis pelo movimento do centro de massa da aeronave. Considerando a projeção do vetor de empuxo no referencial inercial, as equações que descrevem a dinâmica translacional do quadrotor podem ser escritas como:

$$\ddot{x} = \frac{u_1}{m} (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \quad (2)$$

$$\ddot{y} = \frac{u_1}{m} (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \quad (3)$$

$$\ddot{z} = \frac{u_1}{m} (\cos \phi \cos \theta) - g \quad (4)$$

Observa-se que as acelerações translacionais dependem simultaneamente da orientação da aeronave e do empuxo total produzido pelos rotores. Dessa forma, os movimentos translacionais encontram-se diretamente acoplados à dinâmica de atitude, característica inerente aos quadrotores e responsável por parte da complexidade do problema de controle.

2.2.3.2 Dinâmica rotacional

A dinâmica rotacional descreve a evolução temporal da atitude do quadrotor em torno dos eixos longitudinal, transversal e vertical. Considerando os momentos de inércia principais e os torques produzidos pelos rotores, as equações de movimento podem ser expressas pelas Equações (5) a (7).

$$I_{xx} \ddot{\phi} = (I_{yy} - I_{zz}) \dot{\theta} \dot{\psi} + u_2 \quad (5)$$

$$I_{yy} \ddot{\theta} = (I_{zz} - I_{xx}) \dot{\phi} \dot{\psi} + u_3 \quad (6)$$

$$I_{zz} \ddot{\psi} = (I_{xx} - I_{yy}) \dot{\phi} \dot{\theta} + u_4 \quad (7)$$

As Equações (5), (6) e (7) evidenciam que a dinâmica rotacional apresenta termos de acoplamento giroscópico representados pelos produtos entre as velocidades angulares. Esses termos caracterizam o comportamento não linear do quadrotor e justificam a necessidade de estratégias de controle capazes de lidar com tais acoplamentos.

2.2.3.3 Entradas de controle

As quatro ações de controle utilizadas nesta pesquisa correspondem ao empuxo total e aos torques responsáveis pelos movimentos de rolagem, arfagem e guinada. Assim, o vetor de entradas pode ser representado por:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (8)$$

em que u_1 corresponde ao empuxo total gerado pelos quatro rotores, enquanto u_2 , u_3 e u_4 representam, respectivamente, os torques de rolagem, arfagem e guinada. Essas entradas constituem as variáveis de controle do sistema e atuam diretamente sobre a dinâmica translacional e rotacional do quadrotor, sendo utilizadas posteriormente na implementação dos controladores PID e PID-Neural desenvolvidos nesta pesquisa.

2.2.3.4 Relação entre rotores e entradas

A relação entre as velocidades angulares dos rotores e as entradas de controle pode ser expressa de forma compacta por meio da matriz de mistura (*Mixer Matrix*), apresentada na Equação (9). Essa representação é amplamente utilizada na modelagem de quadrotores por estabelecer diretamente a correspondência entre os comandos dos motores e o empuxo total, bem como os torques de rolagem, arfagem e guinada aplicados à aeronave.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (9)$$

em que b representa o coeficiente de sustentação das hélices, responsável por converter a velocidade angular dos rotores em força de empuxo; d corresponde ao coeficiente de arrasto aerodinâmico, associado ao torque de reação gerado pela rotação das hélices; l representa a distância entre o centro de massa do AR.Drone 2.0 e o eixo de cada rotor; e ω_i corresponde à

velocidade angular do i -ésimo rotor. Em conjunto, esses parâmetros definem a matriz de mistura (*Mixer Matrix*), responsável por estabelecer a relação entre os comandos individuais aplicados aos quatro motores e as entradas de controle u_1 , u_2 , u_3 e u_4 , utilizadas para controlar a altitude e a atitude da aeronave durante as simulações desenvolvidas nesta pesquisa.

As equações apresentadas nesta seção constituem o modelo matemático empregado no desenvolvimento do controlador proposto nesta pesquisa. Embora representem a dinâmica completa do quadrotor com seis graus de liberdade, algumas simplificações serão posteriormente adotadas para representar especificamente o voo pairado, condição de operação considerada nas simulações realizadas ao longo deste trabalho. A representação do sistema em espaço de estados, utilizada na implementação computacional do controlador, é apresentada na seção seguinte.

2.2.4 Modelo em espaço de estados

O objetivo deste tópico é apresentar a representação em espaço de estados adotada nesta pesquisa, estabelecendo o modelo utilizado no desenvolvimento dos controladores implementados posteriormente.

A representação em espaço de estados constitui uma das formas mais utilizadas para descrever sistemas dinâmicos multivariáveis, permitindo reunir em um único conjunto de equações diferenciais todas as variáveis responsáveis pela evolução temporal do sistema. Essa abordagem facilita a análise da dinâmica do quadrotor e fornece uma estrutura conveniente para implementação computacional dos algoritmos de controle empregados nesta pesquisa.

Considerando a modelagem completa do quadrotor com seis graus de liberdade, o sistema pode ser representado por um vetor de estados composto por doze variáveis, reunindo as posições e velocidades lineares, os ângulos de Euler e as velocidades angulares da aeronave, conforme apresentado na Equação (10).

$$x = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ \phi \ \theta \ \psi \ p \ q \ r]^T \quad (10)$$

em que x , y e z representam as coordenadas do centro de massa do quadrotor no referencial inercial; $\dot{x}$, $\dot{y}$ e $\dot{z}$ correspondem às velocidades lineares; ϕ , θ e ψ representam, respectivamente, os ângulos de rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*); e p , q e r

representam as velocidades angulares em torno dos eixos do referencial do corpo. Essas doze variáveis descrevem completamente a dinâmica da aeronave, permitindo representar simultaneamente seus movimentos translacionais e rotacionais.

Embora o modelo completo seja adotado como base para a modelagem matemática, nesta pesquisa foi utilizada uma representação reduzida nas simulações de voo pairado, preservando apenas os estados diretamente relacionados ao controle da altitude e da atitude do quadrotor.

Diferentemente de sistemas lineares, a dinâmica do quadrotor apresenta forte não linearidade decorrente das funções trigonométricas presentes nas matrizes de rotação e dos termos de acoplamento entre os movimentos translacionais e rotacionais. Dessa forma, a representação em espaço de estados utilizada nesta pesquisa corresponde a um modelo não linear, implementado diretamente no ambiente *MATLAB/Simulink*.

2.2.4.1 Modelo utilizado na implementação

Uma característica importante do presente trabalho consiste no fato de que o modelo em espaço de estados foi empregado tanto na implementação do controlador PID convencional quanto na arquitetura PID-Neural proposta. Dessa forma, ambos os controladores foram avaliados sobre a mesma planta dinâmica, garantindo que as diferenças observadas nos resultados decorressem exclusivamente da estratégia de controle adotada.

Essa consideração é particularmente importante para a comparação realizada no Capítulo 6, uma vez que elimina possíveis interferências decorrentes de alterações estruturais na planta durante os experimentos computacionais.

Embora a representação em espaço de estados considere a dinâmica completa do quadrotor, o objetivo desta pesquisa concentra-se na estabilização em voo pairado. Assim, torna-se conveniente adotar um modelo reduzido capaz de preservar as principais características dinâmicas da aeronave durante essa condição específica de operação. Essa simplificação é apresentada na próxima subseção.

2.2.5 Modelo reduzido para voo pairado

Embora o modelo apresentado anteriormente descreva a dinâmica completa do quadrotor, incluindo os seis graus de liberdade, a estratégia de controle desenvolvida nesta pesquisa foi direcionada à estabilização em voo pairado (*hover*). Nessa condição de operação,

a aeronave permanece próxima de uma configuração de equilíbrio, permitindo simplificações que reduzem a complexidade computacional sem comprometer a representação das dinâmicas de interesse.

Assim, para fins de desenvolvimento do controlador e realização das simulações, foram considerados prioritariamente os estados associados ao controle da altitude e da atitude da aeronave, mantendo-se a representação das variáveis diretamente relacionadas aos movimentos de rolagem, arfagem, guinada e deslocamento vertical.

Essa abordagem mostrou-se adequada aos objetivos desta pesquisa, uma vez que permitiu concentrar a análise na estabilização do quadrotor durante o voo pairado, condição em que foram realizados todos os experimentos computacionais apresentados posteriormente.

A representação detalhada desse modelo reduzido, bem como sua implementação computacional e os parâmetros utilizados durante as simulações, são apresentados no Capítulo 6, dedicado à validação do controlador proposto.

2.3 Estratégias clássicas de controle

O controle de sistemas dinâmicos constitui uma das áreas fundamentais da Engenharia de Controle e Automação. No caso dos quadrotores, o objetivo principal consiste em manter a estabilidade da aeronave diante de perturbações externas, incertezas do modelo e variações das condições de operação. Para isso, diferentes estratégias de controle foram desenvolvidas ao longo das últimas décadas, variando desde métodos clássicos até abordagens baseadas em inteligência artificial.

Entre as técnicas clássicas, o controlador Proporcional–Integral–Derivativo (PID) permanece como uma das soluções mais utilizadas tanto em aplicações industriais quanto em sistemas embarcados. Sua ampla aceitação deve-se à simplicidade de implementação, ao baixo custo computacional e à capacidade de fornecer respostas satisfatórias para uma grande variedade de sistemas dinâmicos quando corretamente ajustado.

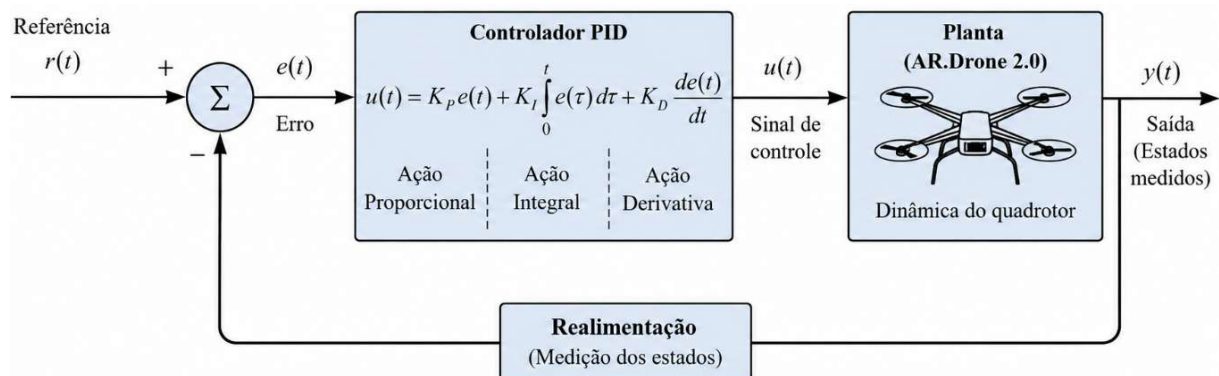
Apesar do desenvolvimento de métodos mais sofisticados, como controle ótimo, controle robusto e controle preditivo, o PID continua sendo amplamente empregado em veículos aéreos não tripulados. Em aplicações próximas ao voo pairado, sua capacidade de estabilizar a atitude do quadrotor torna-o uma alternativa eficiente e de fácil implementação, especialmente quando o sistema opera em torno de um ponto de equilíbrio.

A utilização do controlador PID nesta pesquisa não tem como objetivo competir com técnicas mais complexas, mas estabelecer uma base de controle robusta sobre a qual seja

possível incorporar mecanismos inteligentes de compensação. Essa característica torna o PID particularmente adequado para arquiteturas híbridas, nas quais a inteligência artificial atua apenas como complemento do controlador convencional.

A Figura 5 apresenta a estrutura básica de um sistema de controle em malha fechada utilizando um controlador PID.

Figura 5 – Estrutura básica de um controlador PID em malha fechada.



Fonte: Elaborada pelo Autor (2014).

Observa-se que o controlador calcula continuamente o sinal de controle a partir da diferença entre a referência desejada e a saída medida do sistema. Essa diferença, denominada erro de controle, constitui a informação utilizada para determinar a ação aplicada aos atuadores, buscando reduzir progressivamente o desvio em relação ao ponto de operação desejado.

2.3.1 Controlador Proporcional–Integral–Derivativo (PID)

O controlador PID combina três ações distintas de controle, denominadas proporcional, integral e derivativa. A atuação conjunta dessas componentes permite corrigir o erro instantâneo, reduzir erros acumulados ao longo do tempo e antecipar tendências de variação do sistema, proporcionando respostas mais estáveis e precisas.

A contribuição proporcional produz uma ação diretamente relacionada ao valor instantâneo do erro. Dessa forma, quanto maior o desvio entre a saída do sistema e a referência desejada, maior será o esforço de controle aplicado ao sistema.

A ação integral considera o histórico do erro ao longo do tempo. Sua principal finalidade consiste em eliminar o erro em regime permanente, corrigindo pequenas diferenças que permanecem mesmo após a estabilização do sistema.

Por sua vez, a ação derivativa atua sobre a taxa de variação do erro. Essa componente contribui para reduzir oscilações e melhorar o amortecimento da resposta dinâmica, antecipando mudanças no comportamento do sistema antes que elas provoquem grandes desvios.

A combinação dessas três ações pode ser representada pela Equação (11), que expressa matematicamente o sinal de controle produzido pelo controlador PID.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (11)$$

em que $u(t)$ representa o sinal de controle gerado pelo controlador PID e aplicado à planta; $e(t)$ corresponde ao erro entre o valor de referência $r(t)$ e a saída do sistema $y(t)$; e K_p , K_i e K_d representam, respectivamente, os ganhos das ações proporcional, integral e derivativa.

A ação proporcional responde imediatamente ao erro instantâneo, a ação integral elimina ou reduz o erro em regime permanente mediante sua acumulação ao longo do tempo, enquanto a ação derivativa considera a taxa de variação do erro, contribuindo para a redução de oscilações e para a melhoria da estabilidade do sistema.

Nesta pesquisa, esses três ganhos constituem a base do controlador PID empregado na estabilização do quadrotor, servindo também como referência para a integração da Rede Neural Artificial proposta no controlador PID-Neural apresentado nos capítulos seguintes.

2.3.1.1 Ação proporcional

A ação proporcional produz um sinal de controle diretamente proporcional ao erro instantâneo do sistema. Seu principal efeito consiste em aumentar a velocidade de resposta do controlador, reduzindo o tempo necessário para que a variável controlada se aproxime da referência.

Entretanto, a utilização exclusiva da ação proporcional normalmente não elimina completamente o erro em regime permanente e, quando empregada com ganhos elevados, pode provocar oscilações e comprometer a estabilidade do sistema.

2.3.1.2 Ação integral

A ação integral considera o histórico acumulado do erro ao longo do tempo, permitindo eliminar o erro estacionário observado em controladores puramente proporcionais.

Embora essa ação aumente significativamente a precisão do sistema em regime permanente, valores excessivos do ganho integral podem reduzir o amortecimento da resposta e aumentar o tempo necessário para estabilização do sistema.

2.3.1.3 Ação derivativa

A ação derivativa atua sobre a taxa de variação do erro, antecipando tendências futuras do comportamento do sistema. Como consequência, essa parcela contribui para reduzir o sobressinal e melhorar o amortecimento da resposta transitória. Entretanto, sua elevada sensibilidade ao ruído torna necessária a utilização criteriosa do ganho derivativo, especialmente em sistemas reais sujeitos a medições ruidosas.

2.3.1.4 Influência dos ganhos do controlador

A influência individual de cada ganho sobre o desempenho do controlador pode ser analisada de forma qualitativa. Em geral, alterações nos ganhos proporcional, integral e derivativo produzem efeitos distintos sobre a estabilidade, o tempo de acomodação, o sobressinal e o erro em regime permanente. A Tabela 2 resume esses efeitos de forma comparativa, servindo como referência para o processo de sintonia do controlador PID.

Tabela 2: Influência qualitativa dos ganhos do controlador PID.

Ganho	Ano	Técnica de Controle	Aplicação	Principal contribuição
K_p	Diminui	Aumenta	Diminui	Pode reduzir
K_i	Diminui	Aumenta	Elimina	Pode reduzir
K_d	Diminui	Aumenta	Diminui	Pode reduzir

Fonte: Adaptado pelo autor a partir de Åström e Hägglund (2006).

Embora a sintonia adequada dos ganhos permita obter respostas satisfatórias em diversas aplicações, o desempenho do controlador PID permanece diretamente relacionado à fidelidade do modelo utilizado e às condições de operação do sistema. Em situações caracterizadas por não linearidades acentuadas ou incertezas paramétricas, podem ocorrer degradações de desempenho que motivam o emprego de estratégias complementares de controle.

2.3.1.5 Aplicação ao controle de quadrotores

Em quadrotores, o controlador PID é amplamente empregado para estabilização da altitude e dos ângulos de atitude devido à sua simplicidade de implementação e ao reduzido esforço computacional exigido durante sua execução em tempo real. Diversos trabalhos demonstram que essa estratégia apresenta desempenho satisfatório em condições próximas ao voo pairado, especialmente quando a aeronave opera em torno de um ponto de equilíbrio previamente conhecido (BOUABDALLAH; BECKER; SIEGWART, 2007; BRANDÃO et al., 2012).

Entretanto, por depender diretamente do modelo utilizado durante sua sintonia, o controlador PID pode apresentar degradação de desempenho quando submetido a variações paramétricas, perturbações externas ou incertezas não representadas no modelo matemático. Nessas situações, torna-se necessário empregar estratégias capazes de aumentar a robustez do sistema sem comprometer sua estabilidade.

Essa limitação motivou o desenvolvimento de abordagens híbridas que combinam o controlador PID com técnicas baseadas em inteligência artificial, entre elas as Redes Neurais Artificiais. Nesta dissertação, essa integração é realizada por meio de uma arquitetura na qual a RNA atua como compensador residual exclusivamente sobre o canal correspondente ao empuxo, preservando a estrutura do controlador PID responsável pela estabilização da atitude do quadrotor.

Embora o controlador PID permaneça como uma das técnicas mais utilizadas em sistemas embarcados, outras estratégias clássicas também têm sido empregadas para estabilização de quadrotores. A próxima subseção apresenta uma visão geral dessas abordagens, destacando suas principais características e aplicações.

2.3.2 Outras estratégias clássicas de controle

O objetivo deste tópico é a apresentar as principais estratégias clássicas utilizadas no controle de quadrotores, destacando suas características, vantagens e limitações em relação ao controlador PID.

Embora o controlador PID permaneça como uma das estratégias mais empregadas na estabilização de quadrotores, diversas técnicas de controle foram desenvolvidas ao longo das últimas décadas com o objetivo de melhorar o desempenho do sistema diante das características não lineares da dinâmica da aeronave. Entre essas abordagens destacam-se a linearização por realimentação (*Feedback Linearization*), o controle por *Backstepping*, o controle por modos deslizantes (*Sliding Mode Control*), o controle robusto e o controle adaptativo.

A linearização por realimentação busca transformar a dinâmica não linear do sistema em um modelo aproximadamente linear por meio de mudanças de variáveis e cancelamento das não linearidades. Embora apresente bom desempenho em condições próximas ao modelo nominal, essa técnica depende significativamente da precisão da modelagem matemática e pode apresentar degradação de desempenho na presença de incertezas paramétricas (CASTILLO; DZUL; LOZANO, 2005).

Outra estratégia amplamente investigada é o controle por *Backstepping*, especialmente indicado para sistemas não lineares subatuados, como os quadrotores. Essa técnica realiza o projeto do controlador de forma recursiva, garantindo estabilidade assintótica sob determinadas condições. Entretanto, sua implementação normalmente conduz a leis de controle relativamente complexas, dificultando sua aplicação em plataformas embarcadas de menor capacidade computacional.

O controle por modos deslizantes (*Sliding Mode Control*) destaca-se pela elevada robustez frente a perturbações externas e incertezas do modelo. Em contrapartida, a presença do fenômeno conhecido como *chattering* pode provocar oscilações indesejáveis nos sinais de controle, exigindo técnicas adicionais para sua atenuação.

Também merecem destaque as estratégias de controle adaptativo, nas quais os parâmetros do controlador são ajustados automaticamente durante a operação do sistema, permitindo compensar alterações na dinâmica da planta. Embora apresentem boa capacidade de adaptação, esses métodos geralmente exigem algoritmos mais complexos e maior esforço computacional quando comparados ao controlador PID convencional. O Quadro 2 apresenta uma comparação qualitativa entre as principais estratégias clássicas de controle encontradas na literatura.

Quadro 2: Comparação entre as principais estratégias clássicas de controle.

Estratégia	Vantagens	Limitações
PID	Simplicidade e baixo custo computacional	Sensível às variações paramétricas
Feedback Linearization	Boa resposta dinâmica	Dependência do modelo
Backstepping	Estabilidade garantida	Elevada complexidade
Sliding Mode	Robustez elevada	Chattering
Controle Adaptativo	Ajuste automático dos parâmetros	Maior custo computacional

Fonte: Adaptado pelo autor a partir de Castillo, Dzul e Lozano (2005), Bouabdallah, Becker e Siegwart (2007) e Brandão et al. (2012).

Observa-se que nenhuma das estratégias apresentadas consegue, isoladamente, reunir simultaneamente simplicidade de implementação, robustez frente às incertezas e reduzido esforço computacional. Essa constatação motivou o desenvolvimento de abordagens híbridas capazes de combinar as vantagens dos controladores clássicos com técnicas baseadas em inteligência artificial.

Entre essas abordagens, destacam-se aquelas fundamentadas em Redes Neurais Artificiais, apresentadas na próxima seção.

A evolução das técnicas inteligentes de controle permitiu incorporar mecanismos de aprendizagem e aproximação não linear aos sistemas de controle convencionais. Entre essas técnicas, as Redes Neurais Artificiais destacam-se pela capacidade de representar relações altamente não lineares, tornando-se uma alternativa promissora para compensação de incertezas presentes na dinâmica de quadrotoros.

A análise das características do controlador PID evidencia que sua simplicidade de implementação e robustez justificam sua ampla utilização em aplicações industriais e aeronáuticas. Entretanto, seu desempenho depende diretamente da qualidade do modelo da planta e da sintonia adequada dos ganhos de controle. Em sistemas sujeitos a não linearidades acentuadas ou variações paramétricas, essas limitações podem comprometer a precisão da resposta dinâmica, motivando o emprego de técnicas complementares, como as Redes Neurais Artificiais, abordadas na seção seguinte.

2.4 Redes neurais artificiais

As Redes Neurais Artificiais (RNAs) constituem uma das principais técnicas de Inteligência Artificial aplicadas à modelagem, identificação e controle de sistemas dinâmicos. Sua capacidade de aproximar relações não lineares complexas permite representar comportamentos difíceis de descrever por modelos matemáticos convencionais, característica particularmente relevante em sistemas sujeitos a incertezas e variações paramétricas.

No contexto do controle de quadrotor, essas propriedades tornam as RNAs uma alternativa interessante para complementar controladores clássicos, especialmente quando o modelo da planta não representa integralmente a dinâmica real do sistema. Em vez de substituir completamente o controlador convencional, as redes neurais podem atuar como mecanismos de compensação capazes de reduzir erros decorrentes de simplificações do modelo ou de perturbações externas.

Essa abordagem motivou a utilização de uma Rede Neural Artificial nesta pesquisa. Entretanto, para compreender a estratégia proposta, torna-se necessário apresentar os principais conceitos relacionados à estrutura das RNAs, ao funcionamento do neurônio artificial, ao algoritmo de treinamento e às aplicações dessas redes em sistemas de controle.

2.4.1 Evolução histórica das redes neurais artificiais

O desenvolvimento das Redes Neurais Artificiais ocorreu de forma gradual, acompanhando a evolução da computação e das técnicas de Inteligência Artificial. Desde as primeiras tentativas de reproduzir computacionalmente o comportamento dos neurônios biológicos até as modernas arquiteturas profundas, diversos avanços contribuíram para ampliar o campo de aplicação dessas redes.

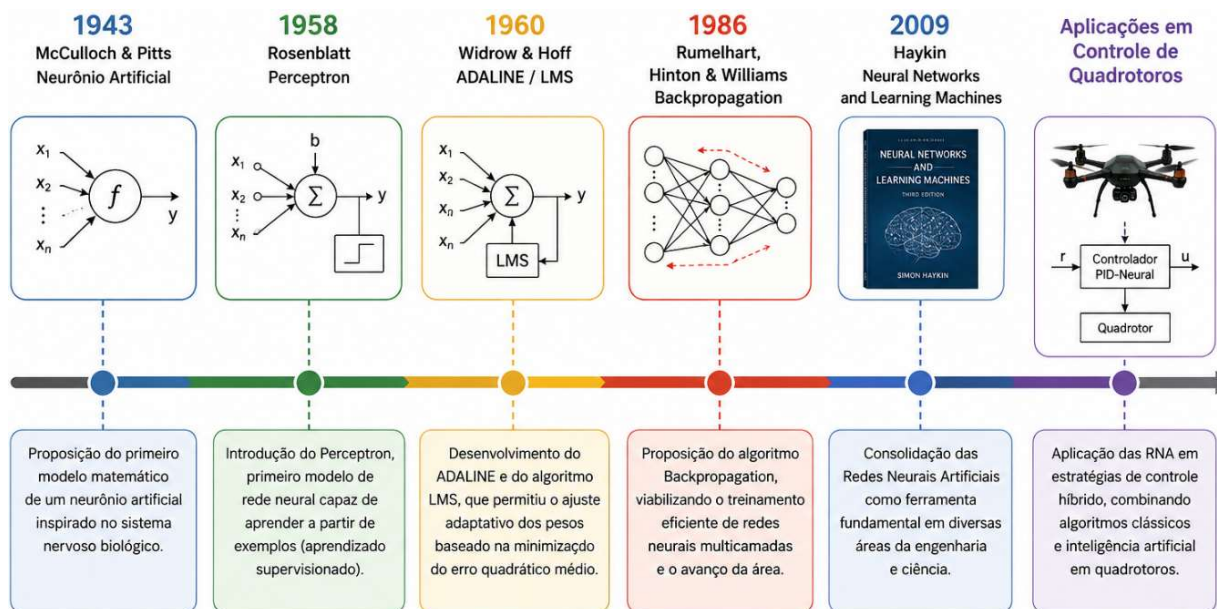
Os primeiros trabalhos remontam à década de 1940, quando McCulloch e Pitts propuseram um modelo matemático simplificado de neurônio artificial. Posteriormente, Rosenblatt introduziu o *Perceptron*, demonstrando a possibilidade de realizar tarefas de classificação por meio de aprendizagem baseada em exemplos.

Embora as limitações do *Perceptron* tenham reduzido temporariamente o interesse pelas Redes Neurais, o desenvolvimento do algoritmo *Backpropagation* na década de 1980 impulsionou uma nova fase de pesquisas. A partir desse momento, tornou-se possível treinar redes multicamadas capazes de representar funções não lineares significativamente mais complexas.

Atualmente, as Redes Neurais Artificiais encontram aplicações em diferentes áreas da engenharia, incluindo identificação de sistemas, reconhecimento de padrões, processamento

de sinais e controle inteligente. A Figura 6 apresenta os principais marcos históricos dessa evolução.

Figura 6 – Evolução histórica das Redes Neurais Artificiais.



Fonte: Adaptado pelo autor a partir de Rosenblatt (1958), Widrow e Hoff (1960), Rumelhart, Hinton e Williams (1986) e Haykin (2009).

Observa-se que a evolução das Redes Neurais Artificiais esteve diretamente relacionada ao desenvolvimento de algoritmos de aprendizagem mais eficientes e ao aumento da capacidade computacional disponível. Esses avanços permitiram sua aplicação em sistemas de controle não linear, contexto no qual se insere a proposta desenvolvida nesta dissertação.

Os modelos neurais modernos são constituídos por unidades de processamento interligadas, denominadas neurônios artificiais. A estrutura e o funcionamento dessas unidades são apresentados na subseção seguinte.

2.4.2 Neurônio artificial e *Perceptron*

O neurônio artificial representa a unidade básica de processamento de uma Rede Neural Artificial. Seu funcionamento baseia-se na combinação ponderada dos sinais de entrada, seguida da aplicação de uma função de ativação responsável por produzir o sinal de saída da rede. Matematicamente, o potencial de ativação pode ser representado por

$$s_j = \sum_{i=1}^n \omega_{ji} x_i + b_j \quad (12)$$

Após a combinação linear das entradas, aplica-se uma função de ativação não linear, obtendo-se a saída do neurônio conforme a Equação (13).

$$h_j = f(s_j) \quad (13)$$

Nas redes multicamadas, as saídas produzidas pelos neurônios da camada oculta são novamente combinadas para gerar a resposta final da rede, conforme descrito pela Equação (14).

$$\hat{y}_k = \sum_{j=1}^n v_{kj} h_j + b_k \quad (14)$$

em que x_i representa a i -ésima variável de entrada da rede neural; ω_{ji} corresponde ao peso sináptico associado à conexão entre a entrada i e o neurônio oculto j ; b_j representa o viés (bias) do neurônio oculto; s_j é o potencial de ativação resultante da combinação linear das entradas; $f(\cdot)$ corresponde à função de ativação; h_j representa a saída do neurônio da camada oculta; v_{kj} é o peso sináptico entre o neurônio oculto j e o neurônio de saída k ; b_k representa o viés do neurônio de saída; e $\hat{y}_k$ corresponde à saída estimada da Rede Neural Artificial.

Em conjunto, essas equações descrevem o processamento realizado por um neurônio artificial em uma Rede Neural Multicamadas. Inicialmente, os sinais de entrada são combinados de forma ponderada pelos respectivos pesos sinápticos. Em seguida, o potencial de ativação é submetido a uma função não linear, produzindo a saída do neurônio. Nas redes multicamadas, essas saídas passam a constituir novas entradas para as camadas subsequentes, permitindo representar relações não lineares entre as variáveis do sistema.

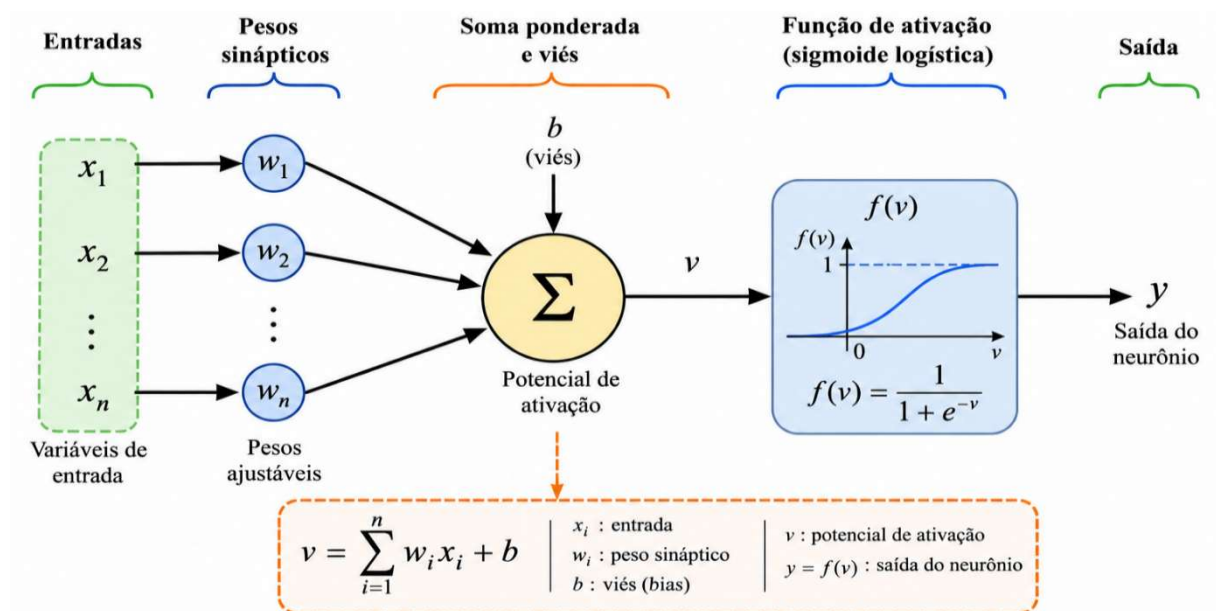
A não linearidade do neurônio é introduzida pela função de ativação. Entre as diversas funções propostas na literatura, adotou-se nesta pesquisa a função sigmoide logística, apresentada na Equação (15), em razão de sua continuidade, diferenciabilidade e ampla utilização no treinamento supervisionado de Redes Neurais Artificiais.

$$y_j = \frac{1}{1 + e^{-v_j}} \quad (15)$$

em que v_j representa o potencial de ativação do neurônio e y_j corresponde ao sinal produzido em sua saída. A função sigmoide limita os valores de saída ao intervalo $(0,1)$, característica que contribui para a estabilidade do processo de treinamento e para a representação de relações não lineares entre as variáveis de entrada e saída.

A Figura 7 apresenta a estrutura básica de um neurônio artificial, destacando os sinais de entrada, os pesos sinápticos, o viés, a função de ativação e a saída produzida pelo modelo.

Figura 7 – Modelo de um neurônio artificial.



Fonte: Elaborada pelo Autor (2014).

O modelo de neurônio proposto por Rosenblatt deu origem ao *Perceptron*, considerado a arquitetura mais simples de Rede Neural Artificial. Essa estrutura é composta por uma única camada de processamento e apresenta capacidade para resolver apenas problemas linearmente separáveis.

Apesar de sua importância histórica, o *Perceptron* apresenta limitações na representação de relações não lineares mais complexas. Essas limitações motivaram o desenvolvimento das Redes Neurais Multicamadas, capazes de aproximar funções não lineares por meio da utilização de camadas ocultas e algoritmos de treinamento mais robustos.

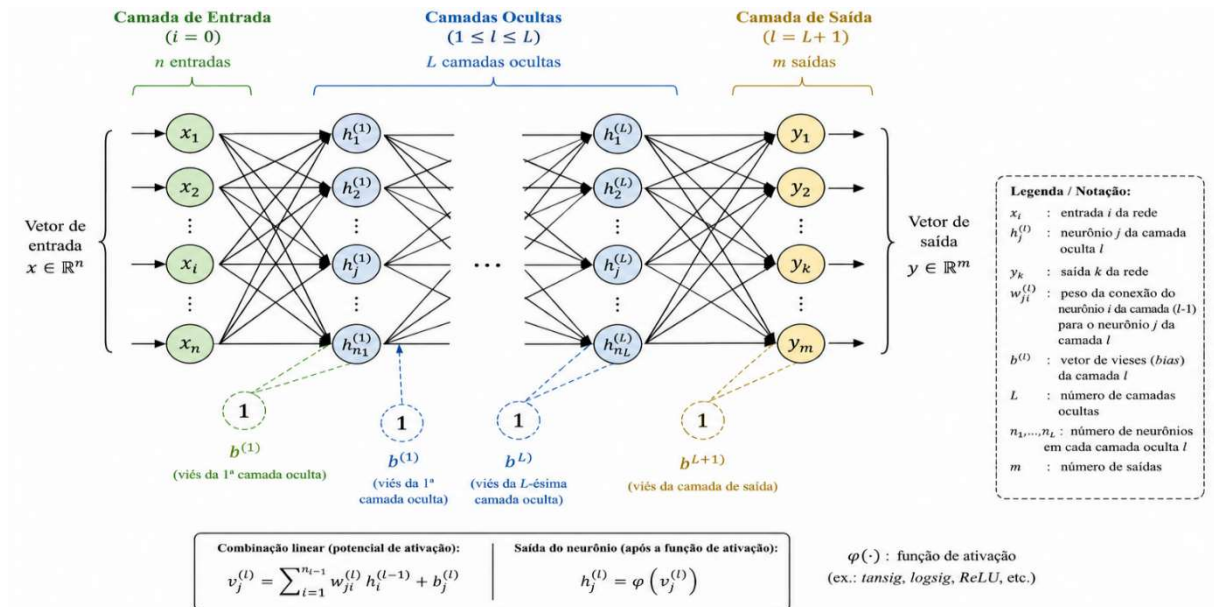
2.4.3 Redes neurais multicamadas

Embora o *Perceptron* simples seja capaz de resolver problemas linearmente separáveis, diversas aplicações práticas exigem modelos capazes de representar relações significativamente mais complexas. Essa limitação motivou o desenvolvimento das Redes Neurais Multicamadas (RNM), constituídas por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

A introdução das camadas ocultas amplia significativamente a capacidade de aproximação não linear da rede, permitindo representar funções complexas por meio da combinação sucessiva de operações simples realizadas pelos neurônios artificiais.

A arquitetura geral utilizada nesta pesquisa é apresentada na Figura 8.

Figura 8 – Arquitetura geral de uma rede neural multicamadas (RNM).



Fonte: Elaborada pelo Autor (2014).

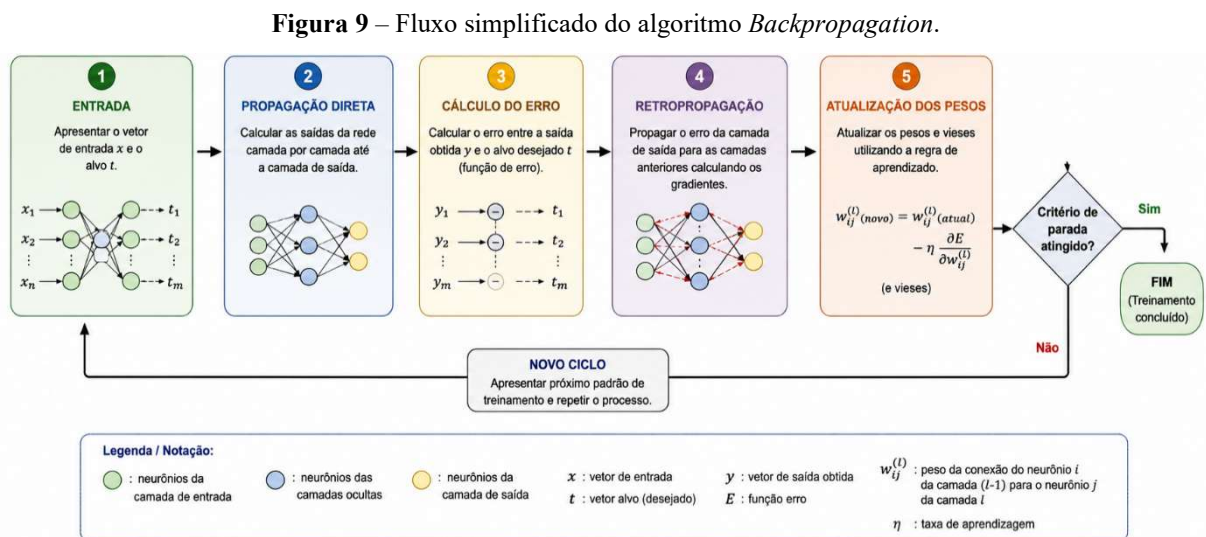
Entre as diferentes arquiteturas disponíveis na literatura, optou-se pela utilização de uma Rede Neural Multicamadas do tipo MLP treinada pelo algoritmo *Backpropagation*. Essa escolha fundamenta-se na elevada capacidade de aproximação de funções não lineares, na ampla utilização dessa arquitetura em problemas de controle e na facilidade de integração com o controlador PID desenvolvido nesta pesquisa.

2.4.4 Algoritmo *Backpropagation*

O treinamento de uma Rede Neural Artificial consiste no ajuste iterativo dos pesos sinápticos, de modo que a saída produzida pela rede se aproxime da resposta desejada. Entre os diversos métodos de aprendizagem supervisionada, o algoritmo *Backpropagation* tornou-se um dos mais utilizados em Redes Neurais Multicamadas devido à sua eficiência e simplicidade de implementação.

O princípio de funcionamento do *Backpropagation* baseia-se na propagação do erro entre a saída estimada pela rede e a saída desejada. Esse erro é utilizado para atualizar os pesos sinápticos das conexões internas, reduzindo progressivamente a função de custo durante o processo de treinamento.

A Figura 9 apresenta um fluxograma simplificado das principais etapas executadas pelo algoritmo *Backpropagation*.



Fonte: Elaborada pelo Autor (2014).

Inicialmente, as variáveis de entrada são apresentadas à rede, produzindo uma resposta por meio da propagação direta (*forward propagation*). Em seguida, a diferença entre a saída estimada e a saída desejada é utilizada para calcular o erro de treinamento.

Na etapa seguinte, esse erro é propagado no sentido inverso da rede (*backward propagation*), permitindo o ajuste dos pesos sinápticos de acordo com uma regra de otimização baseada no gradiente descendente. Esse procedimento é repetido sucessivamente até que seja atingido um critério de parada previamente estabelecido, como número máximo de épocas ou erro mínimo desejado.

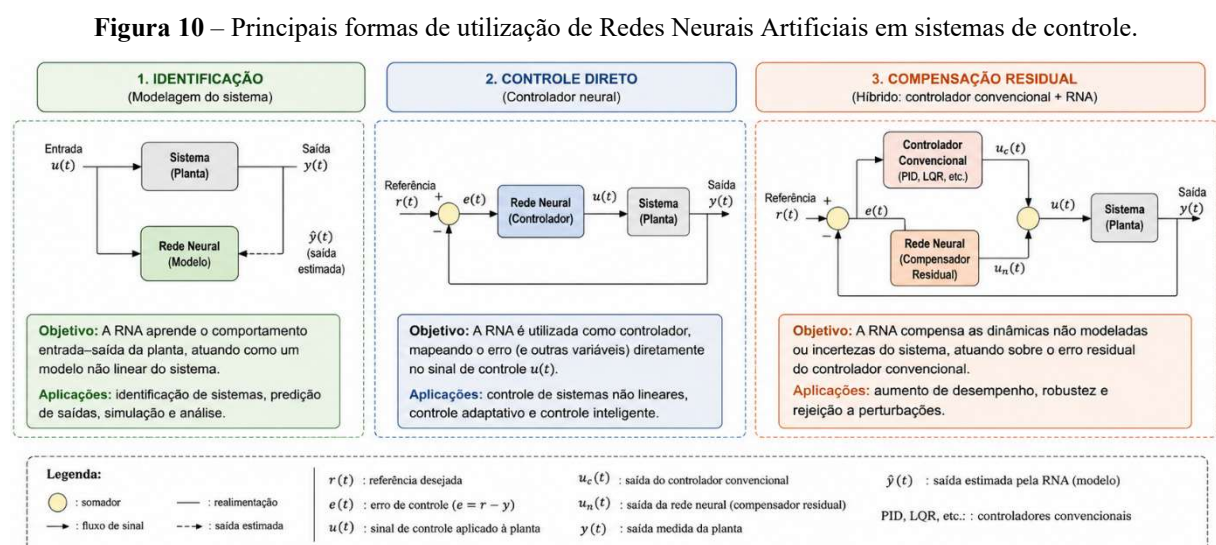
Embora existam diferentes algoritmos de treinamento para Redes Neurais Artificiais, o *Backpropagation* permanece como uma das estratégias mais empregadas em aplicações de identificação de sistemas e controle inteligente. Nesta pesquisa, esse algoritmo foi utilizado no treinamento da Rede Neural Multicamadas responsável pela compensação residual aplicada ao controlador PID.

2.4.5 Redes neurais artificiais aplicadas ao controle

A utilização de Redes Neurais Artificiais em sistemas de controle tornou-se uma alternativa importante para aplicações em que os modelos matemáticos convencionais não representam integralmente a dinâmica da planta. Nessas situações, a capacidade das RNAs de aproximar funções não lineares permite compensar incertezas do modelo e melhorar o desempenho do sistema de controle.

Dependendo da estratégia adotada, uma Rede Neural Artificial pode desempenhar diferentes funções dentro da arquitetura de controle. Entre as aplicações mais comuns destacam-se a identificação de sistemas, a adaptação de parâmetros do controlador, a substituição parcial do controlador convencional e a compensação residual de erros associados à dinâmica da planta.

A Figura 10 apresenta as principais formas de utilização das Redes Neurais Artificiais em sistemas de controle.



Fonte: Elaborada pelo Autor (2014).

Observa-se que essas estratégias diferem principalmente quanto ao nível de participação da rede neural no processo de controle. Enquanto algumas arquiteturas atribuem à RNA a responsabilidade integral pela geração do sinal de controle, outras preservam o controlador convencional e utilizam a rede apenas como elemento auxiliar de compensação.

Essa segunda abordagem apresenta vantagens importantes em aplicações nas quais o controlador clássico já fornece respostas satisfatórias, sendo desejável apenas reduzir erros decorrentes de simplificações do modelo ou de perturbações externas.

Sob essa perspectiva, a estratégia desenvolvida nesta pesquisa enquadra-se na categoria de compensação residual, em que a Rede Neural Artificial atua exclusivamente como elemento complementar ao controlador PID, preservando a estrutura responsável pela estabilização da atitude do quadrotor.

2.4.6 Redes neurais artificiais aplicadas ao controle de quadrotores

As características dinâmicas dos quadrotores tornam essas aeronaves um dos principais campos de aplicação das Redes Neurais Artificiais em sistemas de controle. A presença de não linearidades, acoplamentos entre movimentos e incertezas paramétricas faz com que técnicas baseadas em aprendizado apresentem potencial para complementar estratégias clássicas de estabilização.

Diversos trabalhos propõem a utilização de Redes Neurais Artificiais para identificação da dinâmica do sistema, adaptação automática dos parâmetros do controlador ou substituição parcial do modelo matemático. Em muitos casos, essas abordagens buscam melhorar a robustez do sistema diante de perturbações externas e variações dos parâmetros físicos da aeronave.

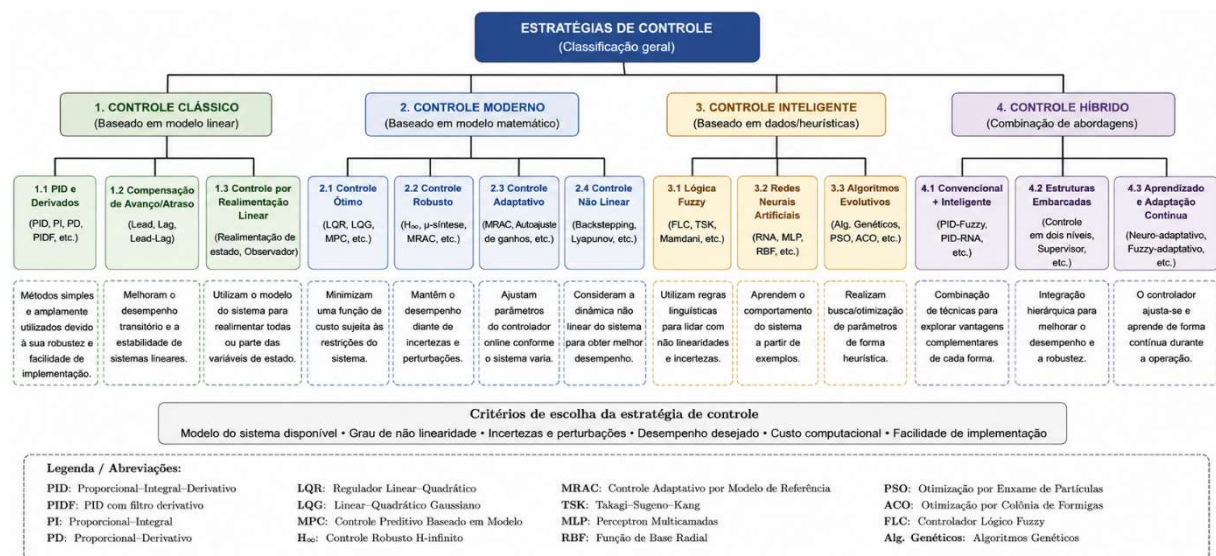
Apesar dos resultados obtidos, observa-se que grande parte dessas estratégias concentra a atuação da rede neural sobre todos os canais de controle do quadrotor. Embora essa abordagem aumente a capacidade adaptativa do controlador, também tende a elevar sua complexidade computacional e a dificultar o processo de treinamento da rede.

Nesta dissertação foi adotada uma estratégia distinta. Em vez de aplicar a compensação neural simultaneamente aos quatro canais de controle, optou-se por restringir sua atuação exclusivamente ao canal de empuxo. Essa escolha procura preservar a estabilidade proporcionada pelo controlador PID nos movimentos de rolagem, arfagem e guinada, ao mesmo tempo em que explora a capacidade de aproximação não linear da RNA para melhorar a resposta dinâmica da altitude.

Essa estratégia reduz o acoplamento entre os canais de controle, simplifica o treinamento da rede neural e mantém a interpretação física do controlador convencional. Além disso, favorece a integração entre técnicas clássicas e inteligentes de controle, objetivo central desta pesquisa.

A Figura 11 apresenta uma classificação das principais estratégias encontradas na literatura, indicando o posicionamento da arquitetura proposta nesta dissertação.

Figura 11 – Classificação das principais estratégias de controle encontradas na literatura.



Fonte: Elaborada pelo Autor (2014).

Observa-se que a estratégia desenvolvida neste trabalho combina características dos controladores híbridos e dos métodos de compensação residual, preservando o controlador PID como elemento principal da malha de controle e utilizando a Rede Neural Artificial apenas para compensar limitações da dinâmica vertical do sistema.

A análise dos trabalhos revisados permite identificar tanto as contribuições já consolidadas quanto as lacunas que ainda permanecem no desenvolvimento de estratégias de controle para quadrotoros. Com o objetivo de sintetizar essas observações e evidenciar o posicionamento da proposta desenvolvida nesta pesquisa, o Quadro 3 apresenta uma comparação entre as principais abordagens encontradas na literatura e a estratégia metodológica adotada nesta dissertação.

Quadro 3: Síntese das lacunas identificadas na literatura.

Aspecto analisado	Literatura	Proposta desta dissertação
Controlador clássico	Sim	Sim
RNA substitui PID	Frequente	Não
RNA adapta todos os canais	Frequente	Não
Compensação residual	Pouco explorada	Sim
Atuação apenas em u_1	Não identificada	Sim
Validação em cenários paramétricos	Parcial	Sim

Fonte: Elaborada pelo Autor (2014).

A comparação apresentada no Quadro 3 evidencia que a contribuição desta pesquisa não consiste na proposição de um novo algoritmo de controle, mas na definição de uma estratégia de integração entre um controlador PID e uma Rede Neural Artificial. A principal inovação está na utilização da RNA como compensador residual restrito ao canal de empuxo, preservando a estrutura clássica responsável pela estabilização da atitude do quadrotor.

2.4.7 Considerações finais

A análise apresentada ao longo desta seção evidenciou que as Redes Neurais Artificiais evoluíram de modelos computacionais inspirados no funcionamento dos neurônios biológicos para ferramentas amplamente empregadas em problemas de identificação e controle de sistemas dinâmicos.

Entre as diferentes arquiteturas disponíveis, as Redes Neurais Multicamadas treinadas pelo algoritmo *Backpropagation* destacam-se pela elevada capacidade de aproximação de funções não lineares e pela ampla utilização em aplicações de engenharia. Essas características justificam sua adoção na estratégia de controle proposta nesta pesquisa.

Também se observou que diferentes arquiteturas podem ser empregadas em sistemas de controle de quadrotor. Entretanto, a maior parte dos trabalhos encontrados na literatura concentra a atuação da Rede Neural Artificial sobre todos os canais de controle da aeronave.

Diferentemente dessas abordagens, esta pesquisa propõe a utilização da RNA exclusivamente como compensador residual do canal de empuxo, preservando a atuação do controlador PID nos canais responsáveis pela estabilização da atitude. Essa estratégia

constitui a principal característica da arquitetura desenvolvida neste trabalho e será detalhada nos capítulos seguintes.

2.5 Trabalhos relacionados

2.5.1 Introdução

A literatura apresenta um número crescente de pesquisas dedicadas ao desenvolvimento de estratégias de controle para quadrotor. Essas contribuições diferem não apenas quanto ao algoritmo empregado, mas também em relação aos objetivos de controle, ao nível de complexidade do modelo adotado e à forma de integração entre técnicas clássicas e métodos baseados em inteligência artificial.

De maneira geral, os trabalhos encontrados podem ser agrupados em três categorias principais. A primeira reúne pesquisas fundamentadas em controladores clássicos. A segunda contempla estratégias baseadas em controle não linear e métodos robustos. A terceira engloba abordagens que utilizam técnicas de inteligência artificial, especialmente Redes Neurais Artificiais, para aumentar a capacidade adaptativa do sistema.

2.5.2 Primeira categoria

Os primeiros estudos concentraram-se na utilização de controladores clássicos, principalmente o controlador Proporcional–Integral–Derivativo (PID). Essas estratégias apresentam baixo custo computacional e facilidade de implementação, sendo particularmente adequadas para condições próximas ao voo pairado.

Bouabdallah, Becker e Siegwart (2007), por exemplo, demonstraram que controladores PID e técnicas de linearização são capazes de proporcionar desempenho satisfatório quando o modelo representa adequadamente a dinâmica do sistema.

2.5.3 Segunda categoria

Posteriormente, diferentes pesquisadores passaram a investigar estratégias baseadas em controle não linear, controle robusto, modos deslizantes, *backstepping* e linearização por realimentação. Essas abordagens procuram aumentar a robustez do controlador diante de

perturbações externas e incertezas paramétricas, embora frequentemente apresentem maior complexidade matemática e computacional.

No cenário nacional, Brandão et al. (2012) desenvolveram uma modelagem não linear subatuada aplicada ao quadrotor *AR.Drone*, contribuindo para o avanço das pesquisas brasileiras envolvendo modelagem dinâmica e sistemas de controle para veículos aéreos não tripulados.

2.5.4 Terceira categoria

Mais recentemente, observa-se um crescimento significativo das pesquisas baseadas em inteligência artificial. Redes Neurais Artificiais passaram a ser empregadas para identificação de sistemas, ajuste automático de parâmetros e compensação de não linearidades presentes na dinâmica do quadrotor.

Entre as principais vantagens dessas abordagens destacam-se a capacidade de aprendizagem, a adaptação a diferentes condições de operação e a representação de relações não lineares complexas. Entretanto, muitos desses trabalhos substituem integralmente o controlador convencional ou aplicam a compensação neural simultaneamente a todos os canais de controle da aeronave.

2.5.5 Lacuna

Apesar da diversidade de estratégias encontradas na literatura, observa-se que ainda existem poucas pesquisas dedicadas à utilização da Rede Neural Artificial exclusivamente como compensador residual do canal de empuxo, preservando o controlador PID como responsável pela estabilização da atitude do quadrotor.

Essa observação evidencia uma oportunidade de pesquisa relevante, uma vez que essa arquitetura procura combinar a robustez do controlador clássico com a capacidade adaptativa das Redes Neurais Artificiais, reduzindo simultaneamente a complexidade computacional do sistema.

2.5.6 Encerramento

Com base nessa análise, verifica-se que a estratégia proposta nesta dissertação diferencia-se das abordagens tradicionalmente encontradas na literatura ao restringir

deliberadamente a atuação da Rede Neural Artificial ao canal de empuxo. Essa característica constitui a principal contribuição metodológica deste trabalho e fundamenta o desenvolvimento do controlador PID-Neural apresentado nos capítulos seguintes.

A análise comparativa dos trabalhos revisados permitiu identificar diferentes estratégias para o controle de quadrotor, evidenciando avanços importantes tanto nas abordagens clássicas quanto nas técnicas baseadas em inteligência artificial. Entretanto, também revelou oportunidades de pesquisa relacionadas à utilização de arquiteturas híbridas que preservem a robustez do controlador PID e empreguem a Rede Neural Artificial apenas como elemento de compensação residual. Essa lacuna fundamenta a proposta metodológica desenvolvida nesta dissertação.

2.6 Considerações finais

A revisão bibliográfica apresentada neste capítulo permitiu compreender a evolução das principais estratégias de modelagem e controle aplicadas a quadrotor, bem como identificar as contribuições e limitações das abordagens encontradas na literatura.

Observou-se que os controladores clássicos permanecem amplamente utilizados devido à simplicidade de implementação e ao reduzido custo computacional. Em contrapartida, métodos baseados em Redes Neurais Artificiais apresentam elevada capacidade para representar relações não lineares e compensar incertezas do modelo, tornando-se alternativas promissoras para aplicações de controle inteligente.

A análise dos trabalhos relacionados também evidenciou que a maior parte das pesquisas concentra a atuação da Rede Neural Artificial sobre todos os canais de controle do quadrotor ou propõe a substituição parcial do controlador convencional.

Diferentemente dessas abordagens, esta pesquisa propõe uma estratégia híbrida baseada na utilização da Rede Neural Artificial como compensador residual aplicado exclusivamente ao canal de empuxo, preservando a atuação do controlador PID na estabilização da atitude da aeronave.

Com base nesses fundamentos, o capítulo seguinte apresenta a modelagem matemática do quadrotor utilizada no desenvolvimento e na validação do controlador PID-Neural proposto.

3 MODELAGEM MATEMÁTICA DO QUADROTOR

3.1 Modelo do quadrotor

O Quadrotor é um VANT que possuiu quatro rotores, conforme Figura 12, pode ser controlado por computadores, *smartphones* ou *tablets*, ficou muito conhecido a partir do seu lançamento por ser um VANT de baixo custo de modo a ser utilizado tanto pela academia, em pesquisas científicas, quanto para o uso civil, como um simples brinquedo que voa controlado pelo celular ou até mesmo para os fotógrafos profissionais quando o assunto é tirar uma foto em grandes alturas.

Figura 12 – Ar.Drone 2.0 Parrot.



Fonte: Lopes *et al.* (2014).

O modelo dinâmico do *drone* é descrito mediante as leis de *Euler-Lagrange*. O momento inercial é calculado assumindo que o quadrotor pode possuir em seu centro uma esfera de raio r e massa M suportando quatro outras massas, os motores no caso. Considerando que cada motor tem a massa m e estão anexados ao centro por um eixo de tamanho l .

3.2 Princípio de funcionamento

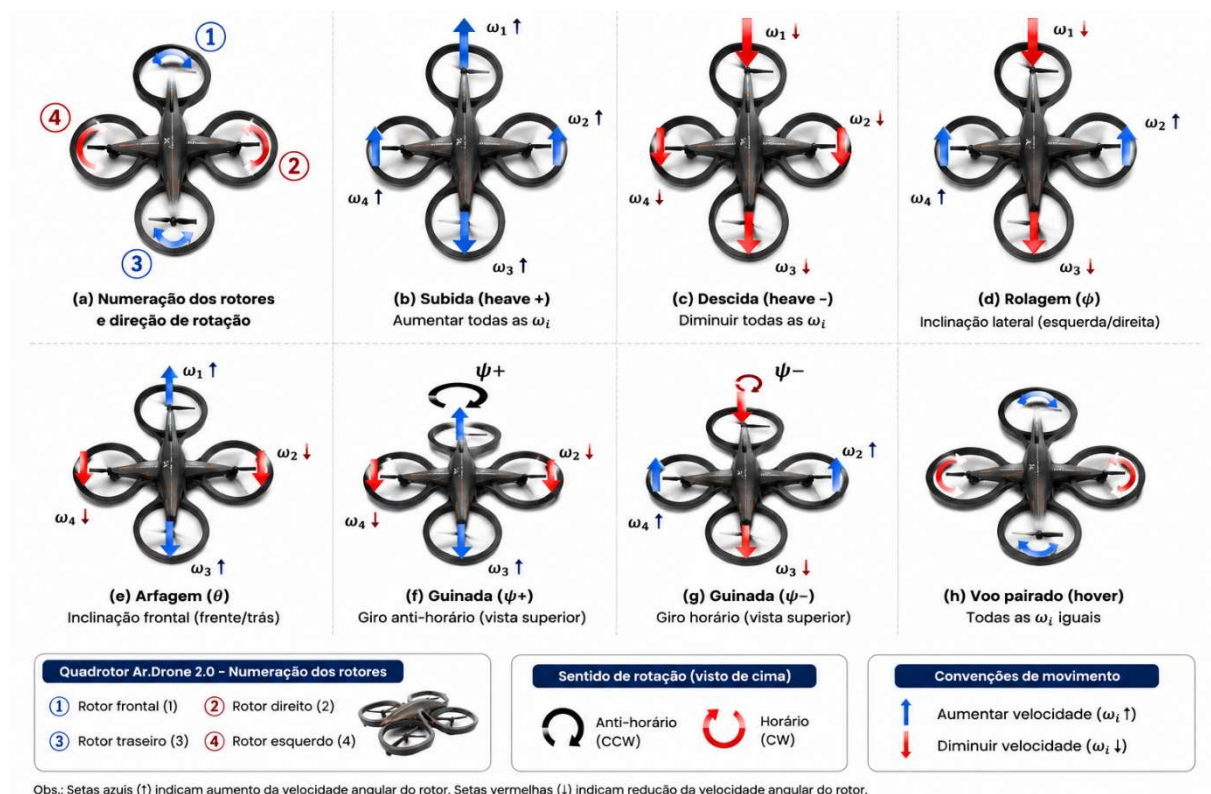
O princípio de funcionamento do quadrotor baseia-se na variação controlada das velocidades angulares dos quatro rotores, responsáveis pela geração do empuxo e dos momentos de controle necessários para estabilizar e manobrar a aeronave. Durante o voo

pairado, todos os rotores giram com velocidades aproximadamente iguais, produzindo um empuxo total suficiente para equilibrar o peso do veículo. A realização de movimentos de translação ou rotação ocorre por meio de pequenas variações nas velocidades dos rotores, alterando a distribuição das forças e dos torques atuantes.

O empuxo gerado por cada rotor é aproximadamente proporcional ao quadrado de sua velocidade angular, de modo que o aumento ou a redução da rotação de um ou mais motores modifica simultaneamente a força total aplicada ao quadrotor e os momentos em torno dos seus eixos principais. Dessa forma, torna-se possível controlar os movimentos de subida e descida (heave), bem como as rotações de rolagem (ϕ), arfagem (θ) e guinada (ψ).

A Figura 13 ilustra os movimentos fundamentais do quadrotor obtidos por meio da variação das velocidades angulares dos rotores, evidenciando a relação entre o acionamento dos motores e os movimentos resultantes da aeronave. Essa estratégia de atuação constitui a base para o desenvolvimento dos controladores utilizados neste trabalho.

Figura 13 – Movimentos do quadrotor obtidos variando as velocidades angulares dos rotores.



Fonte: Adaptado pelo autor a partir de Bouabdallah *et al* (2007).

A Figura 13(a) apresenta a numeração adotada para os rotores do quadrotor *Ar.Drone* 2.0 e os respectivos sentidos de rotação. Nessa configuração, os rotores 1 e 3 giram no sentido

anti-horário (CCW), enquanto os rotores 2 e 4 giram no sentido horário (CW). Essa convenção será adotada ao longo deste trabalho para a descrição das estratégias de controle e da modelagem dinâmica do veículo.

Durante o voo pairado (*hover*), ilustrado na Figura 13(h), os quatro rotores giram com velocidades angulares aproximadamente iguais ($\omega_1 = \omega_2 = \omega_3 = \omega_4$), produzindo um empuxo total suficiente para equilibrar o peso da aeronave e mantê-la em equilíbrio no eixo vertical. Para realizar o movimento de subida (*heave* positivo), representado na Figura 13(b), as velocidades angulares dos quatro rotores são aumentadas simultaneamente e na mesma proporção, elevando o empuxo total e promovendo a ascensão do veículo. De forma análoga, a Figura 13(c) apresenta o movimento de descida (*heave* negativo), obtido pela redução simultânea das velocidades angulares dos quatro rotores, diminuindo o empuxo total e permitindo a perda controlada de altitude.

O controle da guinada (*yaw*) é obtido pela geração de um torque resultante em torno do eixo vertical z, mantendo praticamente constante o empuxo total. Conforme ilustrado nas Figuras 13(f) e 13(g), esse movimento é produzido pela variação diferencial das velocidades dos rotores que giram em sentidos opostos. Em uma configuração convencional de quadrotor, um par de rotores gira no sentido horário (CW) e o outro par no sentido anti-horário (CCW), de modo que o equilíbrio entre os momentos reativos é rompido temporariamente, produzindo a rotação desejada em torno do eixo z.

As rotações em torno dos eixos longitudinal e lateral, correspondentes aos movimentos de rolagem (*roll*) e arfagem (*pitch*), são obtidas por meio da variação diferencial das velocidades dos rotores, conforme ilustrado nas Figuras 13(d) e 13(e), respectivamente. No movimento de rolagem, aumenta-se a velocidade de um par de rotores enquanto se reduz a do par oposto, produzindo um torque em torno do eixo x. De maneira análoga, no movimento de arfagem, a diferença de velocidades entre os rotores dianteiros e traseiros gera um torque em torno do eixo y, promovendo a inclinação frontal ou traseira da aeronave.

Uma vez estabelecido um ângulo de rolagem ou de arfagem, o vetor de empuxo deixa de atuar exclusivamente na direção vertical e passa a apresentar uma componente horizontal. Essa componente é responsável pela aceleração translacional do quadrotor, permitindo seu deslocamento ao longo dos eixos x e y do referencial inercial. Dessa forma, o controle das velocidades angulares dos rotores constitui o mecanismo fundamental para a realização de todos os movimentos básicos do quadrotor.

3.3 Modelagem matemática

Neste capítulo, são apresentadas as discussões detalhadas do modelo do quadrotor. A organização deste capítulo é feita da seguinte forma: os conceitos básicos e generalizações utilizadas neste trabalho são apresentados na Seção 3.3.1. A explicação cinemática dos movimentos de rotação e translação do quadrotor e a matriz de rotação são apresentadas na Seção 3.3.2.

A modelagem dinâmica do quadrotor incluindo a derivação utilizando o formalismo de Euler-Lagrange, as equações não-lineares derivadas do movimento e sua análise detalhada são mostrados na Seção 3.3.3. Os parâmetros do modelo quadrotor utilizadas na simulação são discutidos na Seção 3.4 e, finalmente, as conclusões finais deste capítulo estão incluídas na Seção 3.5.

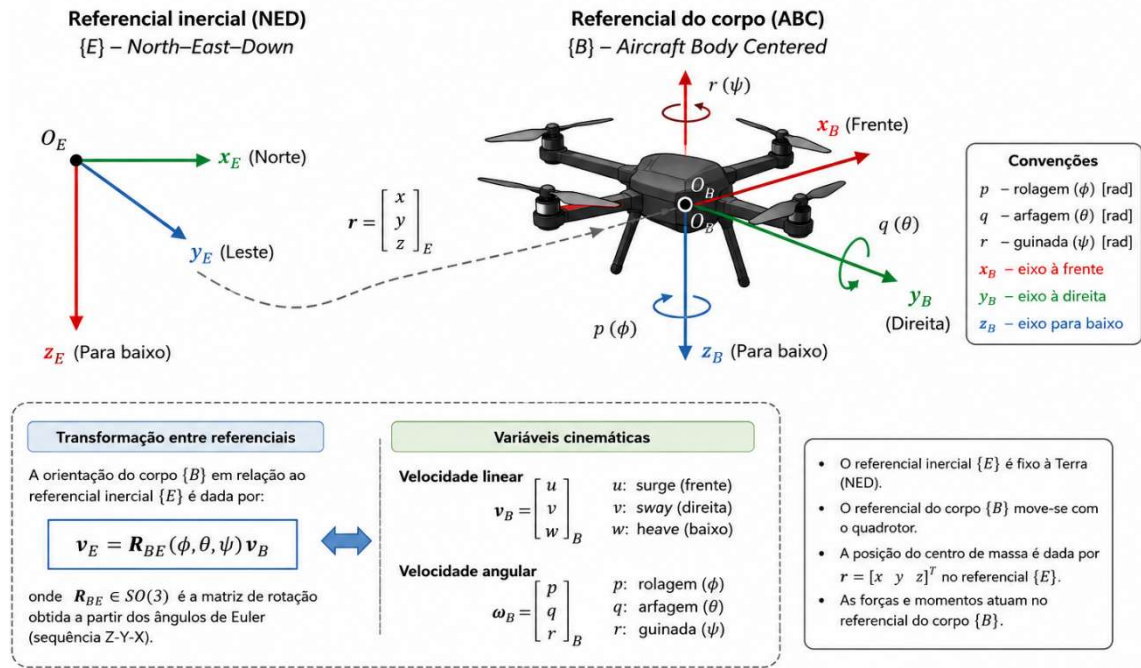
3.3.1 Referenciais e variáveis da modelagem

A modelagem matemática do quadrotor fundamenta-se na representação do movimento da aeronave em dois sistemas de coordenadas complementares: o referencial inercial, fixo à Terra, e o referencial do corpo, solidário ao veículo. A utilização desses referenciais permite descrever de forma consistente a posição, a orientação, as velocidades lineares e as velocidades angulares do quadrotor, constituindo a base para a formulação das equações cinemáticas e dinâmicas apresentadas nas seções seguintes.

No referencial inercial E , são definidas as coordenadas responsáveis por descrever a posição do centro de massa do quadrotor no espaço, enquanto o referencial do corpo B é utilizado para representar as velocidades lineares u , v , w , as velocidades angulares p , q , r e as forças e momentos atuantes sobre a aeronave. A orientação do quadrotor em relação ao referencial inercial é descrita por meio dos ângulos de Euler ϕ , θ , ψ , adotando-se a sequência de rotações compatível com a formulação desenvolvida neste trabalho.

A Figura 14 apresenta os referenciais adotados, as principais variáveis cinemáticas utilizadas na modelagem e a relação entre os sistemas de coordenadas empregados ao longo deste capítulo. Esses elementos constituem a base para a obtenção das transformações cinemáticas e, posteriormente, para a dedução das equações dinâmicas do quadrotor.

Figura 14 – Referenciais e variáveis cinemáticas do quadrotor.



Fonte: Elaborada pelo Autor (2014).

Conforme ilustrado na Figura 14, a descrição completa do movimento do quadrotor exige a utilização simultânea dos referenciais inercial e do corpo. As variáveis de posição x , y , z e os ângulos de Euler ϕ , θ , ψ definem a configuração da aeronave no espaço, enquanto as velocidades lineares u , v , w e angulares p , q , r caracterizam sua dinâmica instantânea. A relação entre esses referenciais é estabelecida por meio das transformações cinemáticas apresentadas na seção seguinte, permitindo a obtenção das equações que descrevem o comportamento do sistema.

3.3.2 Modelagem Cinemática

A modelagem cinemática descreve o movimento do quadrotor por meio das relações geométricas entre posição, orientação, velocidades lineares e velocidades angulares, independentemente das forças e dos momentos responsáveis por esse movimento. Com base nos referenciais apresentados na Seção 3.3.1, estabelecem-se as transformações necessárias entre o referencial inercial e o referencial solidário ao corpo da aeronave, permitindo representar sua configuração espacial ao longo do tempo.

A posição e a orientação da aeronave podem ser reunidas em um vetor de coordenadas generalizadas, também denominado vetor de postura do quadrotor, definido pela Equação (15).

$$q = [\xi, \eta] = (x, y, z, \psi, \theta, \phi) \in R^6, \quad (15)$$

em que x , y e z representam as coordenadas do centro de massa do quadrotor no referencial inercial, enquanto ϕ , θ e ψ representam as coordenadas angulares responsáveis por definir sua orientação espacial.

Para relacionar as grandezas expressas nos referenciais inercial e do corpo, torna-se necessária a utilização de uma matriz de rotação, construída a partir da composição sucessiva de rotações elementares em torno dos três eixos principais do sistema de coordenadas.

$$u = \sum_{i=1}^4 f_i \quad (16)$$

A matriz de rotação R do referencial $\mathbf{B}$ em relação $\mathbf{E}$ é obtida pela multiplicação das três matrizes de rotação dos três eixos x, y, z .

Rotação sobre o eixo Z_E do ângulo ψ (*yaw*).

$$R_z(\psi, z) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Rotação sobre o eixo Y_E do ângulo θ (*pitch*).

$$R_y(\theta, y) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (18)$$

Rotação sobre o eixo X_E do ângulo ϕ (*roll*).

$$R_x(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (19)$$

Multiplicando as três matrizes,

$$R = R_z(\psi, z) \cdot R_y(\theta, y) \cdot R_x(\phi, x) \quad (20)$$

$$R = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (21)$$

A composição das matrizes de rotação elementares resulta na matriz de rotação completa do quadrotor, apresentada na Equação (22).

$$R = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi & \sin\psi\sin\theta\sin\phi - \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\phi \end{bmatrix} \quad (22)$$

A matriz de rotação apresentada na Equação (22) permite transformar vetores entre os referenciais do corpo e inercial, constituindo um dos principais elementos da modelagem cinemática. Essa transformação é utilizada para descrever corretamente a orientação da aeronave ao longo do voo e fornece a base para o desenvolvimento das equações empregadas na modelagem dinâmica.

Seja R uma matriz ortonormal, dada por:

$$R^T R = I_n \quad (23)$$

e sua derivada no tempo é:

$$R^T \dot{R} + R^T \dot{R} = 0_n \quad (24)$$

Definindo S como:

$$S = R^T \dot{R} \quad (25)$$

Obtém-se a partir de (24) que:

$$S^T + S = 0_n \quad (26)$$

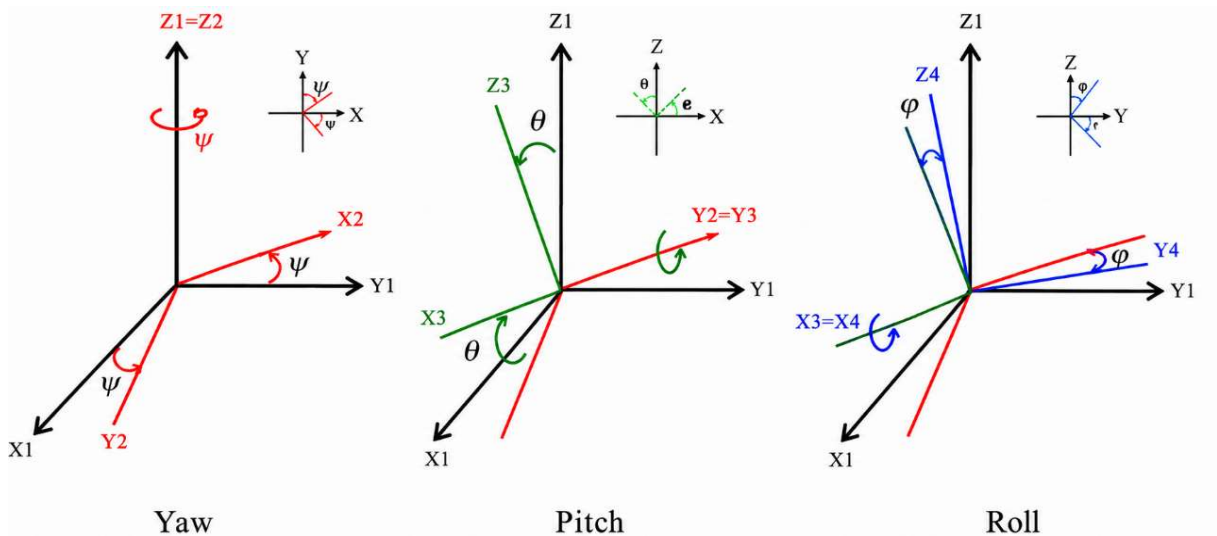
onde S é uma matriz anti-simétrica. A relação entre a derivada da matriz ortonormal e a matriz anti-simétrica é expressa da seguinte forma:

$$S = R^{-1}\dot{R} \quad (27)$$

A orientação de um corpo rígido pode ser representada por diferentes convenções de rotação. Entre elas, destacam-se os ângulos de Tait-Bryan, amplamente empregados em Mecânica de Voo e Engenharia Aeroespacial (RAFFO, 2007).

Neste trabalho, adota-se a sequência de rotações em torno dos eixos z , y e x , correspondente, respectivamente, aos movimentos de guinada ψ , arfagem θ e rolagem ϕ . Essa convenção é amplamente utilizada na modelagem de quadrotoros por representar adequadamente a orientação da aeronave em relação ao referencial inercial (MUÑOZ, 2012). A sequência dessas rotações é ilustrada na Figura 15.

Figura 15 – Rotação dos eixos para os ângulos de Tait-Bryan.



Fonte: Adaptado pelo autor a partir de Muñoz (2012).

As transformações apresentadas nesta seção estabelecem as relações cinemáticas necessárias para descrever a posição e a orientação do quadrotor ao longo do tempo, independentemente das forças responsáveis pelo seu movimento. Essas relações constituem a base para a formulação do modelo dinâmico desenvolvido na seção seguinte, na qual são incorporados os efeitos das forças externas, dos torques produzidos pelos rotores e das propriedades inerciais da aeronave.

3.3.3 Modelagem Dinâmica

Concluída a descrição cinemática do movimento do quadrotor, torna-se necessário estabelecer as relações entre as forças e os torques atuantes sobre a aeronave e o movimento resultante. Enquanto a modelagem cinemática descreve a evolução da posição e da orientação do veículo no espaço sem considerar suas causas, a modelagem dinâmica incorpora os efeitos das forças externas, dos momentos de controle e das propriedades inerciais do sistema, permitindo representar matematicamente o comportamento do quadrotor durante o voo.

No quadrotor, a sustentação é produzida pelo empuxo gerado pelos quatro rotores, enquanto as diferenças entre suas velocidades angulares originam os torques responsáveis pelos movimentos de rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*). O controle simultâneo dessas forças e momentos permite estabilizar a aeronave e realizar os movimentos descritos na Seção 3.2. Para representar esse comportamento, adota-se neste trabalho o formalismo de Euler-Lagrange, amplamente utilizado na modelagem de sistemas mecânicos multicorpos por possibilitar a obtenção sistemática das equações de movimento a partir das energias cinética e potencial do sistema.

O formalismo de Euler-Lagrange descreve a dinâmica do sistema por meio das coordenadas generalizadas definidas na modelagem cinemática, relacionando a energia cinética, a energia potencial e as forças generalizadas atuantes sobre o quadrotor. Dessa forma, as equações diferenciais de movimento podem ser obtidas a partir da expressão apresentada na Equação (28).

$$L(q, \dot{q}) = E_c - E_p \quad (28)$$

em que $L(q, \dot{q})$ representa o Lagrangeano do sistema, definido como a diferença entre a energia cinética total E_c e a energia potencial gravitacional E_p . As variáveis q e $\dot{q}$ representam, respectivamente, o vetor de coordenadas generalizadas e suas derivadas temporais.

em que L representa o Lagrangeano do sistema, definido como a diferença entre a energia cinética total e a energia potencial, q_i corresponde à i -ésima coordenada generalizada e Q_i representa a força generalizada associada à respectiva coordenada.

A energia cinética total do quadrotor é composta pelas contribuições dos movimentos translacional e rotacional da aeronave, podendo ser representada conforme a Equação (29).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (29)$$

em que Q_i , para $i = 1, \dots, n$, representa a força generalizada não conservativa associada à coordenada generalizada q_i , responsável por realizar trabalho virtual sobre o sistema. As coordenadas generalizadas q_i descrevem a configuração do quadrotor, correspondendo às variáveis de posição e orientação $(x, y, z, \phi, \theta, \psi)$, enquanto $\dot{q}_i$ representa sua derivada temporal.

A energia potencial está associada à posição da aeronave no campo gravitacional terrestre, sendo expressa pela Equação (30).

$$q(\gamma, \eta) = [x, y, z, \phi, \theta, \psi]^T \in \mathbb{R}^6 \quad (30)$$

Em que $q(\gamma, \eta) \in \mathbb{R}^6$ representa o vetor de coordenadas generalizadas do quadrotor, composto pela posição $\gamma \in \mathbb{R}^3$ do centro de massa em relação ao sistema de referência inercial e pela orientação $\eta \in \mathbb{R}^3$, descrita pelos ângulos de Euler. O vetor de posição é formado pelas coordenadas cartesianas x , y e z , enquanto o vetor de orientação é constituído pelos ângulos de guinada (ψ), rolagem (ϕ) e arfagem (θ), os quais definem a atitude do veículo durante o voo.

Substituindo as expressões da energia cinética e da energia potencial na formulação de Euler-Lagrange, obtêm-se as equações diferenciais que descrevem a dinâmica translacional e rotacional do quadrotor, apresentadas nas equações subsequentes.

$$L = E_{ct} + E_{cr} - V \quad (31)$$

Em que $E_{ct} = \frac{1}{2} m \dot{\gamma}^T \dot{\gamma}$ a energia cinética translacional do veículo, $E_{cr} = \frac{1}{2} \omega^T I \omega$ a energia cinética rotacional, em que I é o tensor de inércia e Ω o vetor de velocidades angulares, e $E_p = mgh$ a energia potencial gravitacional do quadrotor, determinada em

função de sua massa m , da aceleração da gravidade g e da altura h em relação ao referencial adotado.

Para o sistema geral, h é a altura do veículo, m a massa do quadrotor, Ω_i velocidade angular, I é a matriz de inércia e g é a aceleração da gravidade (MUÑOZ, 2012), então a equação de *Lagrange* está definida como mostra a Equação (32):

$$L = \frac{1}{2} m \dot{\gamma}^T \dot{\gamma} + \frac{1}{2} \omega^T I \omega - mgh \quad (32)$$

Como as energias potenciais e cinéticas são funções das coordenadas generalizadas e suas derivadas temporais, a equação Lagrangeana é também função das mesmas variáveis (MUÑOZ, 2012), usando equação Lagrangeana, as equações de movimento do sistema dinâmico são dadas pela Equação (33):

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \begin{bmatrix} F_i \\ \tau \end{bmatrix} \quad (33)$$

Em que $F_i = R \cdot F$ corresponde ao vetor de forças de translação nas direções x , y e z , obtido pela transformação da força total F aplicada ao veículo por meio da matriz de rotação R . O termo τ representa o vetor dos momentos (torques) associados aos movimentos de rolagem (roll), arfagem (pitch) e guinada (yaw), descritos pelos ângulos de Euler.

Portanto a orientação do quadrotor será definida pela matriz de orientação, a qual relaciona o movimento nos eixos com os três ângulos de *Euler* (MUÑOZ, 2012).

A solução da Equação (33) é dividida na dinâmica para as coordenadas γ e para η , pelo fato de que equação Lagrangeana é definida em função da energia cinética translacional e energia cinética rotacional, como é representada nas Equações (34) e (35).

$$F_i = \frac{d}{dt} \left(\frac{\partial L_T}{\partial \dot{\gamma}} \right) - \frac{\partial L_T}{\partial \gamma} \quad (34)$$

$$\tau = \frac{d}{dt} \left(\frac{\partial L_R}{\partial \dot{\eta}} \right) - \frac{\partial L_R}{\partial \eta} \quad (35)$$

As entradas do sistema são apresentadas em u_1 , u_2 , u_3 e u_4 , representam, respectivamente, o empuxo total e os torques associados aos movimentos de rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*) sendo mostradas nas equações a seguir:

$$u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (36)$$

$$u_2 = bl(\omega_4^2 - \omega_2^2) \quad (37)$$

$$u_3 = bl(\omega_3^2 - \omega_1^2) \quad (38)$$

$$u_4 = d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \quad (39)$$

Assim, o modelo dinâmico do Quadrotor em consequência das rotações de *pitch* ou *roll*, usando a matriz R juntamente com a 2ª Lei de Newton é:

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \cdot \frac{1}{m} u_1 \quad (40)$$

$$\ddot{y} = (\cos\phi \sin\theta \cos\psi - \sin\phi \sin\psi) \cdot \frac{1}{m} u_1 \quad (41)$$

$$\ddot{z} = -g + (\cos\phi \cos\theta) \cdot \frac{1}{m} u_1 \quad (42)$$

$$\ddot{\phi} = \dot{\theta} \dot{\psi} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) - \frac{I_r}{I_{xx}} \dot{\theta} \Omega + \frac{1}{I_{xx}} u_2 \quad (43)$$

$$\ddot{\theta} = \dot{\phi} \dot{\psi} \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) - \frac{I_r}{I_{yy}} \dot{\phi} \Omega + \frac{1}{I_{yy}} u_3 \quad (44)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}u_4 \quad (45)$$

onde I_r é a inércia do rotor, ϕ , θ e ψ são os ângulos de *roll*, *pitch* e *yaw*, respectivamente e I_{xx} , I_{yy} e I_{zz} os momentos de inércia nos eixos x , y e z .

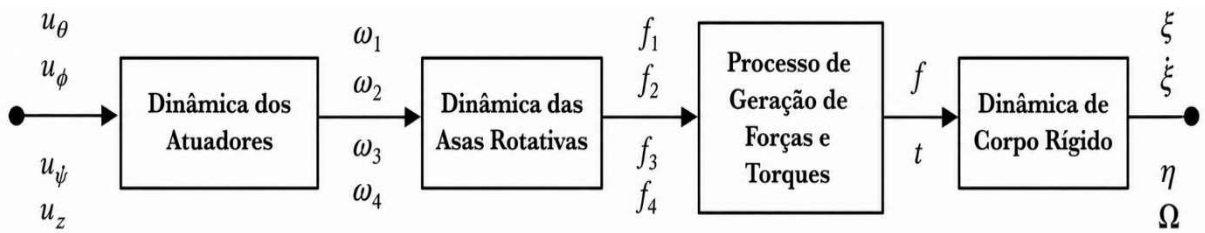
As equações obtidas representam o modelo dinâmico não linear do quadrotor, evidenciando o acoplamento existente entre os movimentos translacionais e rotacionais da aeronave. Esse modelo constitui a base para o desenvolvimento do controlador PID utilizado neste trabalho e, posteriormente, para a integração da Rede Neural Artificial na estratégia de controle híbrido proposta.

3.4 Parâmetros do quadrotor

Esta seção visa apresentar os parâmetros do quadrotor utilizado na simulação para a validação dos controladores de voo, conforme Tabela 3.

A Figura 16 ilustra o diagrama de blocos do modelo dinâmico do Quadrotor. No presente trabalho, os únicos dados disponíveis são os sinais de entrada (que são conhecidos) e as respostas de orientação η e velocidades lineares $\dot{\xi}$ e angulares $\dot{\eta}$.

Figura 16 – Diagrama de blocos do modelo dinâmico do quadrotor.



Fonte: Adaptado pelo autor a partir de BRANDÃO *et al.* (2012).

Assim, ao realizar uma estratégia de identificação de sistemas, o modelo a ser identificado estaria relacionando as entradas de comandos reais (u_i) e as variáveis de postura.

Tabela 3 – Parâmetros do modelo de quadrotor

Símbolo	Descrição	Valor
l	Distância dos rotores ao centro de massa do sistema	0,1782 [m]
m	Massa total	0,380 [kg]
g	Gravidade local	9,810 [m · s ⁻²]
I_{yy}	Momento de inércia do eixo y	0,01857 [kg · m ²]
I_{xy}	Momento de inércia do eixo x em relação ao eixo y	0,00000 [kg · m ²]
I_{yz}	Momento de inércia do eixo y em relação ao eixo z	0,00000 [kg · m ²]
I_{xx}	Momento de inércia do eixo x	0,00957 [kg · m ²]
I_{xz}	Momento de inércia do eixo x em relação ao eixo z	0,0000 [kg · m ²]
I_{zz}	Momento de inércia do eixo z	0,0255 [kg · m ²]
J_r	Momento de inércia do rotor	$2,8 \cdot 10^{-8}$ [kg · m ²]
b	Coefficiente de empuxo das hélices	$1,1429 \cdot 10^{-9}$ [N · s ²]
d	Coefficiente de arrasto da estrutura	$3,2 \cdot 10^{-11}$ [N · m · s ²]

Fonte: Adaptado pelo autor a partir de Lopes *et al.* (2014).

O modelo matemático desenvolvido neste capítulo constitui a base para implementação da estratégia híbrida apresentada no capítulo seguinte.

4 PROJETO DO CONTROLADOR PID-NEURAL

Neste capítulo são apresentados o desenvolvimento e a implementação das estratégias de controle utilizadas para a estabilização do quadrotor em voo pairado, tomando como base a modelagem matemática desenvolvida no capítulo anterior. Inicialmente, é descrito o controlador PID convencional empregado como referência para as análises comparativas. Em seguida, são apresentados o controlador PID-Neural original e a estratégia final baseada em uma RNA residual aplicada ao canal de empuxo, destacando-se sua formulação, treinamento, validação e integração à malha de controle. Também são descritos os sinais utilizados pela rede neural, os procedimentos de adaptação dos pesos sinápticos e a estrutura computacional

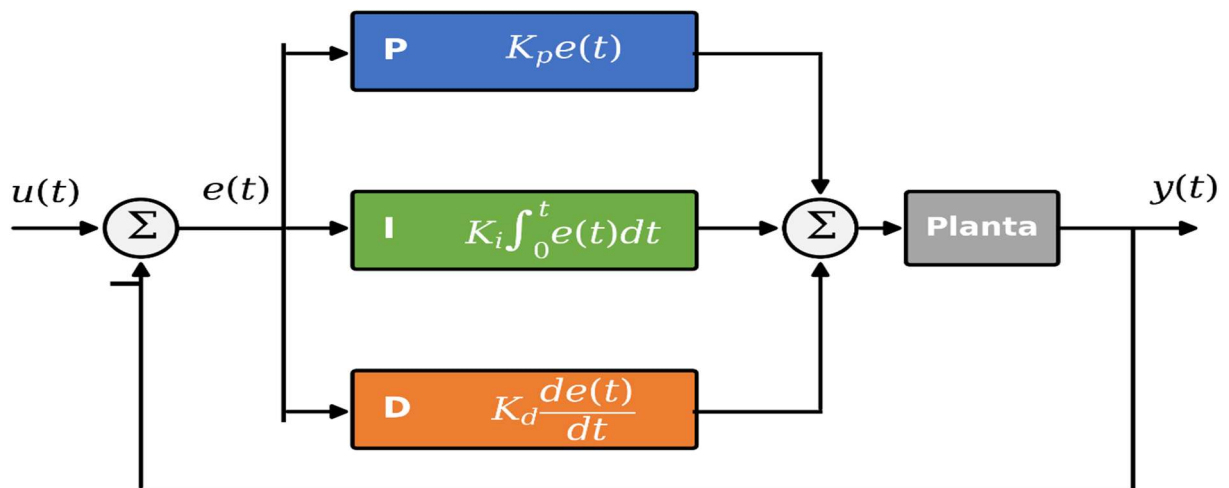
implementada no ambiente *MATLAB/Simulink*, estabelecendo a base necessária para a realização das simulações e das análises comparativas apresentadas nos capítulos seguintes.

4.1 Controlador PID

Os controladores PID são controladores com realimentação, e são muito utilizados em automação industrial. Esses controladores calculam um erro entre o valor medido na saída e o valor desejado no processo. Assim o controlador tenta diminuir o erro que foi gerado pela saída, ajustando suas entradas.

O cálculo do controlador PID envolve três parâmetros: Proporcional, Integral e Derivativo, como mostra a Figura 17.

Figura 17 – Modelo de um controlador Proporcional Integral Derivativo.



Fonte: Elaborada pelo Autor (2014).

Função Proporcional:

$$P_{out} = K_p e(t) \quad (46)$$

K_p é a constante proporcional;

$e(t)$ é o sinal de erro atuante;

Essa função proporcional do controlador PID, representada pela Equação (46), produz um valor na saída proporcional ao erro obtido na Realimentação. A resposta proporcional pode ser ajustada a partir da constante de ganho K_p . O aumento do ganho proporcional tende a

tornar a resposta mais rápida e a reduzir o erro em regime transitório; entretanto, valores excessivos podem elevar o sobressinal, intensificar as oscilações e comprometer a estabilidade. Por outro lado, valores muito baixos de K_p geralmente produzem respostas mais lentas e menor capacidade de correção do erro.

Função integral:

$$I_{out} = K_i \int_0^t e(t) dt \quad (47)$$

K_i é a constante integral;

A função integral do controlador PID, representada pela Equação (47), soma todos os erros instantâneos e a somatória é multiplicada pela constante K_i . A função integral do controlador PID acelera o movimento do processo até o ponto desejado e elimina o erro que ocorre na função anterior. Como a função soma dados instantâneos, o resultado do processo pode ultrapassar o ponto desejado. Essa consequência se chama "overshoot".

Função derivativa:

$$D_{out} = K_d \frac{d}{dt} e(t) \quad (48)$$

K_d é a constante derivativa.

A função derivativa do controlador PID, representada pela Equação (48), retarda a taxa de variação de saída do controlador. Essa função diminui o "overshoot" da função anterior e melhora a estabilidade do controlador. Por outro lado, a função derivativa causa um retardo na resposta e é muito suscetível a ruídos. Isto acontece porque essa função amplifica o ruído. No caso que o ruído e o ganho K_d sejam grandes, podem causar instabilidade no controlador.

A fórmula do controlador PID é dada pela Equação (49):

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (49)$$

Onde: $u(t)$ é a entrada em relação ao tempo;

A Figura 16 apresenta a estrutura clássica de um controlador PID. Observa-se que o sinal de controle resulta da soma das ações proporcional, integral e derivativa, permitindo reduzir o erro em regime permanente, aumentar a rapidez da resposta e melhorar o amortecimento do sistema.

Portanto, o desempenho do sistema depende das constantes que podem melhorar ou piorar a estabilidade do sistema a ser controlado.

4.2 Controlador PID em tempo discreto

O controlador PID-Neural é baseado no modelo da equação discreta da função de transferência do controlador PID. A Equação (50) representa a função genérica do controlador PID no domínio s .

$$\frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (50)$$

Em seguida a representação da função do controlador PID no domínio do tempo discreto é obtida por mapeamento da função transferência do domínio s para o domínio z usando aproximações discretas (FERRARI, 2011).

Método *forward*:

$$s = \frac{z - 1}{\Delta t} \quad (51)$$

Método *backward*:

$$s = \frac{z - 1}{z\Delta t} \quad (52)$$

A parte integral é aproximada usando o método *forward*, conforme Equação (51) e a parte derivativa com *backward*, conforme Equação (52), que substituindo na Equação (50) resulta na Equação (53):

$$C(z) = K_p + K_i \frac{\Delta t}{z-1} + K_d \frac{z-1}{z\Delta t} \quad (53)$$

O sinal de controle de tempo discreto $u(n)$, conforme Equação (54) é derivado, do sinal de erro $e(n)$:

$$u(n) = K_p e(n) + K_i v(n) + K_d w(n) \quad (54)$$

Onde, $v(n)$ é o termo integral e $w(n)$ é o termo derivativo.

$$\frac{V(s)}{E(s)} = \frac{1}{s} \quad (55)$$

$$\frac{W(s)}{E(s)} = s \quad (56)$$

Representando as Equações (55) e (56) no domínio z , obtemos as Equações (57) e (58):

$$\frac{V(z)}{E(z)} = \frac{\Delta t}{z-1} \quad (57)$$

$$\frac{W(z)}{E(z)} = \frac{z-1}{z\Delta t} \quad (58)$$

As Equações (59) e (60) representam o termo integral $v(n)$ e o termo derivativo $w(n)$ em tempo discreto respectivamente (FERRARI, 2011).

$$v(n) = v(n-1) + \Delta t \cdot e(n) \quad (59)$$

$$w(n) = \frac{1}{\Delta t} (e(n) - e(n-1)) \quad (60)$$

4.3 Projeto do controlador PID-Neural.

O controlador desenvolvido neste trabalho foi concebido a partir da integração entre um controlador PID convencional e uma Rede Neural Artificial (RNA), buscando explorar simultaneamente a robustez do controle clássico e a capacidade de aproximação não linear proporcionada pelas redes neurais. A estratégia proposta utiliza o controlador PID como responsável pela estabilização principal da aeronave, enquanto a RNA atua como um mecanismo complementar capaz de estimar correções adicionais ao sinal de controle em situações nas quais o modelo matemático apresenta incertezas ou variações paramétricas.

A concepção do controlador baseou-se na hipótese de que o PID convencional é capaz de estabilizar o quadrotor em condições nominais de voo pairado, porém seu desempenho pode ser degradado diante de alterações de massa, variações de parâmetros físicos ou simplificações introduzidas na modelagem matemática. Nesse contexto, a RNA foi incorporada como um compensador não linear destinado a reduzir os erros residuais produzidos pelo controlador clássico.

A arquitetura proposta utiliza como entradas os estados do sistema e os sinais de erro associados à altitude e à atitude do quadrotor. Esses sinais são processados pela rede neural, que estima uma correção residual adicionada ao sinal de controle produzido pelo controlador PID. Inicialmente foi considerada a aplicação da correção neural sobre todos os canais de controle (empuxo e torques). Entretanto, os resultados obtidos durante a etapa de treinamento mostraram que essa configuração poderia introduzir oscilações adicionais na dinâmica angular do veículo.

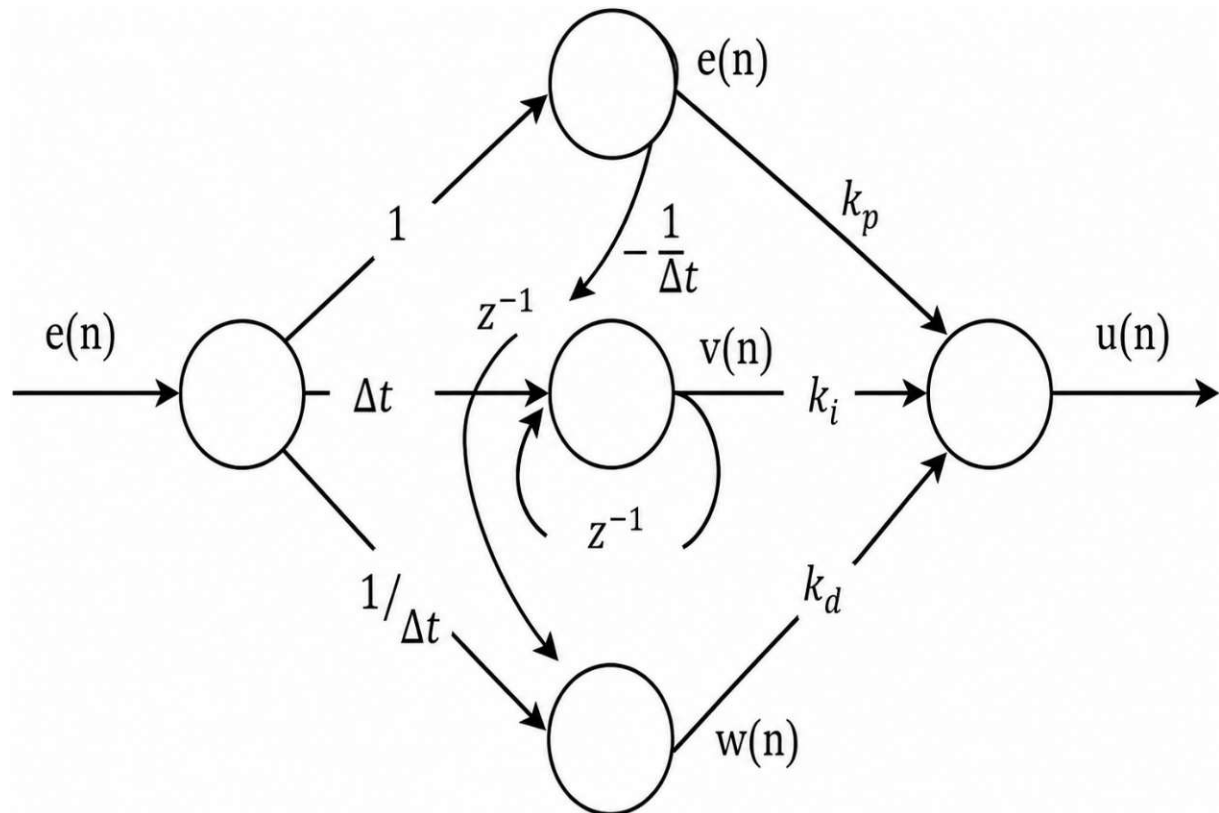
Em razão desse comportamento, adotou-se como configuração final a aplicação da correção neural exclusivamente ao canal correspondente ao empuxo total, preservando os torques de rolagem, arfagem e guinada sob responsabilidade exclusiva do controlador PID. Essa estratégia permitiu melhorar a resposta vertical sem comprometer a estabilidade angular da aeronave, constituindo a principal contribuição desta pesquisa.

Durante o desenvolvimento desta pesquisa foram avaliadas diferentes possibilidades de integração entre o controlador PID e a Rede Neural Artificial. Inicialmente considerou-se a atuação neural sobre os quatro canais de controle do quadrotor. Entretanto, as análises realizadas demonstraram que essa configuração aumentava significativamente a complexidade do processo de ajuste sem produzir ganhos consistentes de desempenho.

Em razão disso, optou-se por restringir a contribuição neural exclusivamente ao canal correspondente ao empuxo total (u_1), mantendo os torques de rolagem, arfagem e guinada sob responsabilidade do controlador PID convencional. Essa estratégia reduziu o acoplamento entre os canais de controle e apresentou maior robustez frente às incertezas paramétricas

consideradas nesta pesquisa, sendo adotada como arquitetura definitiva do controlador PID-Neural.

Figura 18 – Estrutura do controlador PID-Neural.



Fonte: Elaborada pelo autor (2014).

A entrada da rede é o sinal de erro $e(n)$ e a saída é o sinal de controle $u(n)$. A função de ativação dos neurônios é linear (FERRARI, 2011).

Conforme ilustrado na Figura 18, o controlador desenvolvido neste trabalho é composto por dois módulos principais: o controlador PID convencional e a Rede Neural Artificial. Inicialmente, as referências de voo pairado são comparadas com os estados estimados do quadrotor, produzindo os sinais de erro utilizados pelo controlador PID. Em seguida, esses sinais são também disponibilizados à RNA, responsável por estimar uma parcela residual destinada a compensar efeitos não modelados da dinâmica da planta.

A estratégia proposta preserva o controlador PID como responsável pela estabilização principal do sistema, enquanto a RNA atua apenas como mecanismo de compensação não linear. Essa abordagem permite combinar a robustez do controle clássico com a capacidade de

aproximação funcional das redes neurais, reduzindo os efeitos decorrentes de simplificações da modelagem matemática e de variações paramétricas da planta.

4.4 Equações de adaptação do PID-Neural

As equações apresentadas nesta seção descrevem a formulação original do controlador PID-Neural, baseada na adaptação dos ganhos proporcional, integral e derivativo. Durante o desenvolvimento da pesquisa, essa configuração foi avaliada como etapa inicial. Entretanto, os resultados obtidos motivaram a adoção da arquitetura residual apresentada nas seções seguintes, na qual a RNA não modifica diretamente os ganhos do PID, mas estima uma correção complementar aplicada ao sinal de controle. Na configuração final, essa correção foi restrita ao canal de empuxo u_1 , mantendo-se os torques u_2 , u_3 e u_4 sob atuação exclusiva do controlador PID convencional.

Os ganhos K_p , K_i e K_d da estrutura neural do controlador PID, como mostra a Figura 17, serão os pesos a serem sintonizados e o restante dos pesos possuem valores fixos. A saída da rede pode ser computada seguindo as Equações (61), (62) e (63):

$$e(n) = 0_1(n) \quad (61)$$

$$v(n) = 0_2(n) = 0_2(n-1) + \Delta t \cdot e(n) \quad (62)$$

$$w(n) = 0_3(n) + \frac{1}{\Delta t} (e(n) - e(n-1)) \quad (63)$$

O algoritmo de treinamento do controlador PID-Neural é baseado no método do gradiente descendente e utiliza o sinal de erro $e(n)$ que é a diferença entre o sinal desejado $d(n)$ e resposta de saída da rede neural, conforme Equação (64).

$$\varepsilon(n) = d(n) - u(n)w(n) = 0_3(n) + \frac{1}{\Delta t} (e(n) - e(n-1)) \quad (64)$$

Para o treinamento off-line, os dados são gerados de um sistema PID simulado e usado para treinar a rede neural. O desempenho desejado é minimizado pela soma dos erros E_{sq} do treinamento dos dados, conforme Equação (65).

$$E_{sq}(n) = \frac{1}{2} \sum_{n=1}^N \varepsilon(n)^2 \quad (65)$$

em que $E_{sq}(n)$ representa a soma quadrática dos erros acumulados durante o treinamento, N corresponde ao número total de iterações (ou amostras) consideradas no processo de treinamento, e $\varepsilon(n)$ representa o erro na n -ésima amostra, definido como a diferença entre a saída desejada e a saída produzida pela rede neural.

De acordo com método do gradiente descendente, a atualização dos pesos segue as seguintes derivações, conforme Equações (66), (67) e (68).

$$k_p(n+1) = k_p(n) - \eta \frac{\partial E_{sq}(n)}{\partial k_p} \quad (66)$$

$$k_i(n+1) = k_i(n) - \eta \frac{\partial E_{sq}(n)}{\partial k_i} \quad (67)$$

$$k_d(n+1) = k_d(n) - \eta \frac{\partial E_{sq}(n)}{\partial k_d} \quad (68)$$

Em que $K_d(n+1)$ representa o ganho derivativo na iteração subsequente, calculado a partir do ganho derivativo da iteração anterior $K_d(n)$, da taxa de aprendizagem η e do gradiente da função erro $\frac{\partial E_{sq}(n)}{\partial K_d}$. O parâmetro η corresponde à taxa de aprendizagem responsável por controlar a intensidade da atualização do ganho, enquanto $E_{sq}(n)$ representa a soma dos erros na n -ésima iteração. De forma análoga, $K_p(n)$ e $K_i(n)$ correspondem, respectivamente, aos ganhos proporcional e integral na iteração n , ao passo que $K_p(n+1)$ e $K_i(n+1)$ representam os respectivos ganhos atualizados na iteração subsequente.

A regra da cadeia da derivada é usada na retropropagação do erro para o ajuste dos pesos e na minimização da performance do treinamento. A derivada da soma dos erros quadráticos em relação aos seus respectivos parâmetros pode ser escrita conforme Equações (69), (70) e (71).

$$\frac{\partial E_{sq}(n)}{\partial k_p} = -\varepsilon(n) \frac{\partial u(n)}{\partial k_p} = -\varepsilon(n) 0_1(n) \quad (69)$$

$$\frac{\partial E_{sq}(n)}{\partial k_i} = -\varepsilon(n) \frac{\partial u(n)}{\partial k_i} = -\varepsilon(n) 0_2(n) \quad (70)$$

$$\frac{\partial E_{sq}(n)}{\partial k_d} = -\varepsilon(n) \frac{\partial u(n)}{\partial k_d} = -\varepsilon(n) 0_3(n) \quad (71)$$

4.5 Estrutura interna do controlador PID-Neural

Após a definição da arquitetura geral do controlador PID-Neural apresentada na seção anterior, torna-se necessário detalhar sua estrutura interna e o modo como a Rede Neural Artificial (RNA) é integrada ao controlador PID. Enquanto a Figura 17 apresenta a concepção da estratégia de controle proposta, esta seção descreve os componentes responsáveis pelo funcionamento interno do controlador, incluindo as variáveis utilizadas pela RNA, sua arquitetura, o processo de estimação da correção residual e a forma como essa contribuição é incorporada ao sinal de controle.

A organização adotada procura evidenciar a interação entre o controlador clássico e a RNA, demonstrando que a rede neural não substitui o controlador PID, mas atua como um mecanismo complementar destinado a compensar efeitos não modelados da dinâmica do quadrotor. Essa abordagem preserva a robustez do controle clássico e, simultaneamente, explora a capacidade de aproximação não linear característica das redes neurais artificiais.

4.5.1 Variáveis de entrada da rede neural

A seleção das variáveis de entrada representa uma etapa fundamental no desenvolvimento da rede neural, uma vez que essas informações determinam a capacidade da RNA em identificar padrões associados à dinâmica do quadrotor durante o voo pairado. Dessa forma, foram selecionadas variáveis capazes de representar tanto o comportamento dinâmico da planta quanto o desempenho do controlador PID.

O vetor de entrada é constituído pelos erros de altitude, rolagem, arfagem e guinada, pelos termos integrais desses erros e pelas velocidades linear e angulares associadas ao movimento da aeronave. Em conjunto, essas variáveis fornecem à RNA informações

suficientes para estimar a parcela residual do sinal de controle, permitindo compensar pequenas diferenças entre o modelo matemático e o comportamento observado durante as simulações.

Durante a fase de desenvolvimento também foram avaliadas diferentes combinações de variáveis de entrada. Entretanto, verificou-se que a utilização apenas das variáveis diretamente relacionadas ao estado do sistema proporcionou melhor capacidade de generalização da rede, reduzindo a complexidade do treinamento sem comprometer o desempenho do controlador.

Quadro 4 – Características da Rede Neural Artificial utilizada no controlador PID-Neural

Característica	Especificação
Finalidade	Estimar uma correção residual para complementar a ação do controlador PID durante a estabilização em voo pairado.
Tipo de rede	Rede Neural Artificial do tipo <i>feedforward</i> multicamada (MLP).
Variáveis de entrada	Erros de altitude, rolagem, arfagem e guinada; integrais dos erros; velocidade vertical; velocidades angulares (p , q e r); variáveis auxiliares representativas da dinâmica da planta.
Variável de saída	Correção residual do sinal de controle aplicada ao empuxo total (Δu_1).
Arquitetura	Uma camada de entrada, duas camadas ocultas e uma camada de saída.
Número de neurônios	24 neurônios na primeira camada oculta e 16 neurônios na segunda camada oculta.
Método de treinamento	Aprendizado supervisionado utilizando o algoritmo de Levenberg–Marquardt (<i>trainlm</i>).
Critério de otimização	Minimização do erro quadrático médio (MSE) entre a saída desejada e a saída estimada pela RNA.
Estratégia de integração	A RNA atua de forma residual sobre o controlador PID, fornecendo uma correção adicional apenas ao canal de empuxo, enquanto os torques permanecem sob responsabilidade do controlador PID.

Fonte: Elaborada pelo autor (2014).

As características apresentadas no Quadro 4 sintetizam a configuração geral da RNA empregada nesta pesquisa. A partir dessas informações, a subseção seguinte descreve a arquitetura adotada para implementação da rede neural no controlador PID-Neural.

4.5.2 Arquitetura da rede neural artificial

A rede neural empregada neste trabalho possui arquitetura do tipo *feedforward* multicamada (MLP), cuja fundamentação teórica foi apresentada no Capítulo 2. Essa arquitetura foi escolhida devido à sua elevada capacidade de aproximação de funções não

lineares, característica particularmente importante para compensar incertezas presentes na dinâmica do quadrotor.

A configuração adotada é composta por uma camada de entrada, duas camadas ocultas contendo 24 e 16 neurônios, respectivamente, e uma camada de saída responsável pela estimação da correção residual aplicada ao sinal de controle. O treinamento supervisionado foi realizado por meio do algoritmo de Levenberg–Marquardt, amplamente empregado em problemas de aproximação funcional devido à sua rápida convergência e estabilidade numérica.

As funções de ativação utilizadas em cada camada seguem os conceitos apresentados no Capítulo 2, não sendo reproduzidas nesta seção para evitar repetições desnecessárias. O objetivo desta subseção é destacar apenas a arquitetura efetivamente utilizada na implementação do controlador.

A arquitetura implementada fundamenta-se nos conceitos apresentados no Capítulo 2, compreendendo o modelo do neurônio artificial (Figura 6), a arquitetura das redes neurais do tipo *feedforward* multicamada (Figura 7) e as funções de ativação empregadas durante o treinamento da RNA. Dessa forma, esta subseção concentra-se exclusivamente na configuração da rede neural utilizada no controlador PID-Neural, evitando a repetição dos fundamentos teóricos já discutidos.

Conforme ilustrado na Figura 17, o controlador recebe continuamente os estados estimados da planta e as referências de voo pairado. A partir dessas informações são calculados os erros utilizados pelo PID e pela RNA, permitindo que ambos contribuam para a geração do sinal de controle aplicado ao sistema dinâmico.

A arquitetura geral das redes neurais *feedforward* utilizadas nesta pesquisa foi apresentada nas Figuras 5 a 8 do Capítulo 2.

4.5.3 Cálculo da saída da rede neural

Durante a operação do controlador, as variáveis de entrada são processadas pela RNA por meio do mecanismo de propagação direta (*forward propagation*). Nesse processo, os sinais percorrem sucessivamente as camadas da rede, sendo ponderados pelos respectivos pesos sinápticos e transformados pelas funções de ativação até a obtenção da saída da RNA.

A saída produzida pela rede corresponde à estimativa da parcela residual do sinal de controle. Diferentemente do controlador PID convencional, cuja ação depende exclusivamente dos erros de rastreamento, a RNA utiliza informações provenientes do estado

da planta para estimar uma correção adicional destinada a compensar efeitos decorrentes de simplificações do modelo matemático e pequenas variações paramétricas.

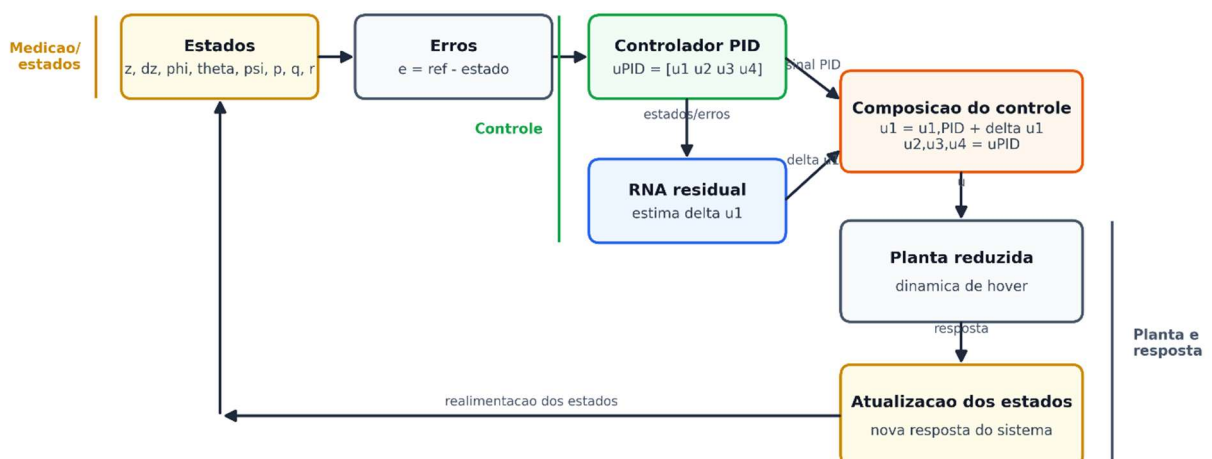
Essa estratégia permite combinar a robustez do controlador PID com a capacidade de aproximação funcional da rede neural, produzindo um controlador híbrido capaz de preservar a estabilidade do sistema e reduzir os erros residuais observados durante o voo pairado.

4.5.4 Integração entre a RNA e o controlador PID

A integração entre a Rede Neural Artificial e o controlador PID constitui a principal estratégia de controle desenvolvida nesta pesquisa. Inicialmente, o controlador PID calcula os sinais de controle correspondentes ao empuxo total e aos torques de rolagem, arfagem e guinada a partir dos erros de rastreamento obtidos na malha de controle. Em seguida, esses mesmos estados são fornecidos à RNA, responsável por estimar uma correção residual aplicada ao sinal de empuxo.

Durante a fase inicial do desenvolvimento foram avaliadas arquiteturas nas quais a RNA estimava correções para os quatro canais de controle do quadrotor. Entretanto, os resultados das simulações mostraram que essa configuração introduzia pequenas oscilações adicionais na dinâmica angular da aeronave. Em razão desse comportamento, adotou-se como configuração final a atuação da RNA exclusivamente sobre o canal de empuxo total, mantendo os torques sob responsabilidade exclusiva do controlador PID. A integração entre o controlador PID e a Rede Neural Artificial, bem como o fluxo de processamento dos sinais durante a execução do controlador, são apresentados na Figura 19.

Figura 19 – Fluxo operacional do controlador PID-Neural



Conforme ilustrado na Figura 18, o controlador PID permanece responsável pela estabilização principal do quadrotor, enquanto a RNA atua como um compensador residual aplicado apenas ao empuxo total. Os estados atualizados da planta são continuamente realimentados ao controlador, caracterizando uma estrutura de controle em malha fechada. Essa estratégia permitiu melhorar a resposta vertical do sistema sem comprometer a estabilidade dos movimentos de atitude, constituindo a configuração final utilizada nas simulações apresentadas no Capítulo 6. A formulação matemática correspondente à integração entre a RNA e o controlador PID é apresentada pelas Equações (101) e (102).

4.6 Treinamento e validação da RNA

Após a definição da estrutura interna do controlador PID-Neural, foi necessário estabelecer um procedimento sistemático para o treinamento da Rede Neural Artificial. Essa etapa teve como objetivo ajustar os pesos sinápticos da rede de forma que sua saída representasse adequadamente a parcela residual do sinal de controle produzida durante a estabilização do quadrotor em voo pairado.

O processo de treinamento foi dividido em quatro etapas principais: geração da base de dados, definição das configurações de treinamento, ajuste iterativo dos coeficientes de ponderação e validação da rede treinada. Essa organização permitiu avaliar o desempenho da RNA em diferentes cenários e selecionar a configuração que apresentou maior capacidade de generalização antes de sua integração definitiva ao controlador PID.

4.6.1 Geração da base de dados

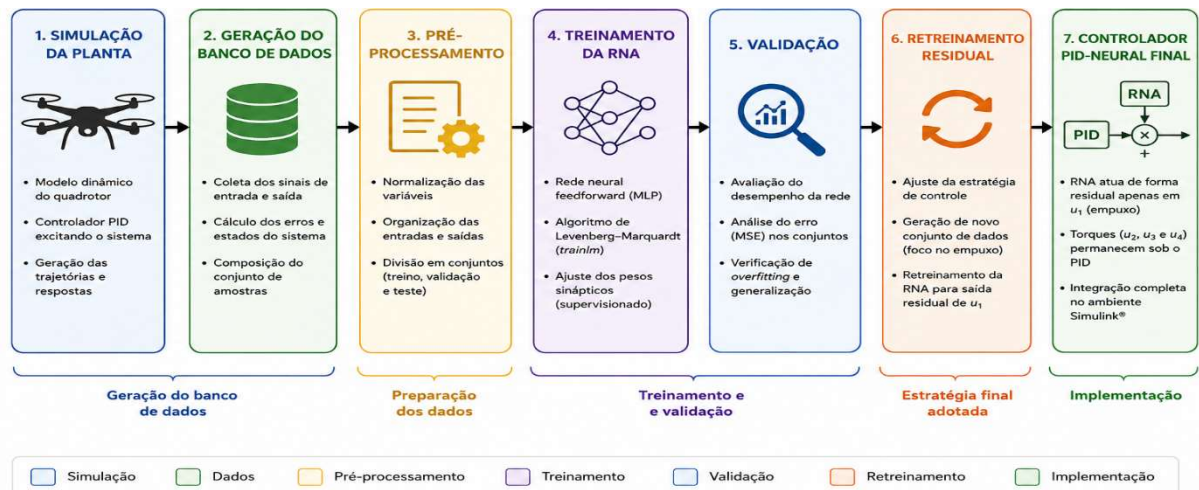
O desempenho de uma rede neural supervisionada depende diretamente da qualidade e da representatividade dos dados utilizados durante o treinamento. Dessa forma, foi desenvolvida uma metodologia específica para geração automática da base de dados empregada nesta pesquisa, utilizando o modelo matemático do quadrotor implementado no ambiente *MATLAB®/Simulink®*.

Inicialmente, foram realizadas diversas simulações do sistema em voo pairado, considerando diferentes condições iniciais, perturbações e variações dos parâmetros da planta. Durante cada simulação foram registrados os estados do quadrotor, os sinais de erro

calculados pelo controlador PID e as respectivas ações de controle necessárias para estabilização da aeronave.

Os dados obtidos foram posteriormente organizados em matrizes de entrada e saída compatíveis com o processo de treinamento supervisionado da RNA. Antes do treinamento, realizou-se uma etapa de pré-processamento dos dados, eliminando amostras inconsistentes e padronizando as variáveis utilizadas pela rede neural. Esse procedimento contribuiu para aumentar a estabilidade do treinamento e reduzir a influência de valores discrepantes (*outliers*). A Figura 20 apresenta o fluxo das etapas empregadas na geração e preparação da base de dados utilizada no treinamento da rede neural.

Figura 20 – Fluxograma do processo de geração e preparação da base de dados para treinamento da Rede Neural Artificial.



Fonte: Elaborada pelo autor (2014).

Conforme ilustrado na Figura 20, a construção da base de dados inicia-se com a execução das simulações do modelo matemático do quadrotor, seguida da aquisição dos estados da planta e dos sinais de controle gerados pelo controlador PID. Em seguida, os dados são organizados, pré-processados e divididos em conjuntos de treinamento, validação e teste, constituindo a base utilizada no processo de aprendizagem supervisionada da rede neural.

4.6.2 Configuração do treinamento

Após a geração da base de dados, foram definidos os parâmetros utilizados durante o processo de treinamento supervisionado da rede neural. Esses parâmetros estabelecem a

arquitetura da RNA, o algoritmo de otimização, os critérios de convergência e a estratégia empregada para seleção da configuração final do controlador.

A escolha dessas configurações buscou conciliar capacidade de aproximação não linear, estabilidade do processo de treinamento e custo computacional, possibilitando a obtenção de uma rede neural com boa capacidade de generalização. O Quadro 5 sintetiza os principais parâmetros empregados no treinamento e na validação da RNA, constituindo a configuração utilizada para obtenção do controlador PID-Neural proposto nesta pesquisa.

Quadro 5 – Configurações utilizadas no treinamento e retreinamento da Rede Neural Artificial

Parâmetro	Configuração adotada
Objetivo do treinamento	Ajustar os pesos sinápticos da RNA para estimar a correção residual aplicada ao controlador PID durante a estabilização do quadrotor em voo pairado.
Ambiente computacional	<i>MATLAB®</i> e <i>Simulink®</i> .
Tipo de rede	Rede Neural Artificial <i>feedforward</i> multicamada (MLP).
Tipo de aprendizado	Supervisionado.
Algoritmo de treinamento	<i>Levenberg–Marquardt (trainlm)</i> .
Base de dados	Obtida a partir de simulações do modelo matemático do quadrotor em diferentes condições de operação.
Variáveis de entrada	Estados da planta, erros de rastreamento, termos integrais e velocidades lineares e angulares selecionadas para representar a dinâmica do sistema.
Divisão da base de dados	Correção residual aplicada ao canal de empuxo do controlador PID.
Função de desempenho	70% para treinamento, 15% para validação e 15% para teste.
Crítérios de parada	Erro Quadrático Médio (<i>Mean Squared Error – MSE</i>).
Critério de seleção da rede	Atingimento da meta de desempenho, número máximo de épocas ou interrupção por validação (<i>early stopping</i>).
Configuração final adotada	A RNA atua de forma residual sobre o controlador PID, fornecendo uma correção adicional apenas ao canal de empuxo, enquanto os torques permanecem sob responsabilidade do controlador PID.

Fonte: Elaborada pelo autor (2014).

Conforme apresentado no Quadro 5, o treinamento da RNA foi realizado utilizando aprendizado supervisionado com o algoritmo de Levenberg–Marquardt, adotando critérios de convergência baseados no erro quadrático médio e na capacidade de generalização da rede. Essas configurações foram mantidas durante as etapas de treinamento e retreinamento,

permitindo avaliar de forma consistente o desempenho da RNA em diferentes cenários de simulação.

4.6.3 Processo de treinamento

O treinamento da Rede Neural Artificial foi realizado por meio de aprendizado supervisionado, utilizando como base os dados obtidos durante as simulações do controlador PID. O objetivo dessa etapa consiste em ajustar iterativamente os coeficientes de ponderação da rede neural, de modo que sua saída represente adequadamente a correção residual aplicada ao controlador durante a estabilização do quadrotor em voo pairado.

Inicialmente, os sinais de entrada são propagados pela RNA por meio do processo de propagação direta (*forward propagation*). Nessa etapa, calcula-se a saída dos neurônios da camada escondida e, posteriormente, a saída do neurônio da camada de saída, conforme as Equações (72) a (75).

$$h_j = \frac{1}{1 + e^{-s_j}} \quad (72)$$

$$s_j = \sum_{i=1}^3 w_{ji} x_i \quad (73)$$

$$u(t) = \frac{1}{1 + e^{-r}} \quad (74)$$

$$r = \sum_{j=1}^3 v_j h_j \quad (75)$$

Após a propagação direta, define-se a função custo utilizada durante o treinamento. Nesta pesquisa foi adotado o erro quadrático acumulado, apresentado na Equação (76), cuja finalidade é quantificar a diferença entre a saída desejada e a saída produzida pela rede neural.

$$E(t) = \frac{1}{2} \sum_{k=1}^t e_y(k)^2 \quad (76)$$

A minimização da função custo é realizada utilizando o método do gradiente descendente, no qual a direção de atualização dos coeficientes de ponderação é determinada pelo vetor gradiente da função erro em relação aos pesos da rede. O gradiente correspondente é definido pela Equação (77).

$$\nabla E(t) = \begin{bmatrix} \frac{\partial E(t)}{\partial v_j} \\ \frac{\partial E(t)}{\partial w_{ji}} \end{bmatrix} \quad (77)$$

A partir dessa definição, inicia-se o cálculo das derivadas parciais associadas aos coeficientes da camada de saída. Aplicando a regra da cadeia, obtêm-se as Equações (78) e (79), que relacionam a função erro aos coeficientes de ponderação v_j .

$$\frac{\partial E(t)}{\partial v_j} = \frac{\partial E(t)}{\partial e_y} \cdot \frac{\partial e_y}{\partial e_u} \cdot \frac{\partial e_u}{\partial u(t)} \cdot \frac{\partial u(t)}{\partial r} \cdot \frac{\partial r}{\partial v_j} \quad (78)$$

$$\frac{\partial E(t)}{\partial v_j} = e_y \cdot \frac{\partial e_y}{\partial e_u} \cdot (-1) \cdot u(t) \cdot (1 - u(t)) \cdot h_j \quad (79)$$

A derivada da função logística apresentada na Equação (80) permite simplificar a expressão anterior e introduzir o termo auxiliar σ^1 , definido na Equação (81). Com essa substituição, a derivada parcial em relação aos coeficientes da camada de saída passa a assumir a forma apresentada na Equação (82).

$$\frac{\partial u(t)}{\partial r} = \frac{e^{-r}}{1 + e^{-r}} \cdot \frac{1}{1 + e^{-r}} = u(t) \cdot (1 - u(t)) \quad (80)$$

$$\sigma^1 = e_y \cdot u(t) \cdot (1 - u(t)) \quad (81)$$

$$\frac{\partial E(t)}{\partial v_j} = \sigma^1 \cdot h_j \cdot \frac{\partial e_y}{\partial e_u} \quad (82)$$

Em seguida, aplica-se novamente a regra da cadeia para determinar a derivada parcial da função custo em relação aos coeficientes de ponderação da camada escondida. Esse desenvolvimento conduz às Equações (83) e (84), nas quais é introduzido o termo auxiliar σ_j^2 , definido na Equação (85). A expressão final para o gradiente associado aos coeficientes w_{ji} é apresentada na Equação (86).

$$\frac{\partial E(t)}{\partial w_{ji}} = \frac{\partial E(t)}{\partial e_y} \cdot \frac{\partial e_y}{\partial e_u} \cdot \frac{\partial e_u}{\partial u(t)} \cdot \frac{\partial u(t)}{\partial r} \cdot \frac{\partial r}{\partial h_j} \cdot \frac{\partial h_j}{\partial S_j} \cdot \frac{\partial S_j}{\partial w_{ji}} \quad (83)$$

$$\frac{\partial E(t)}{\partial w_{ji}} = e_y \cdot \frac{\partial e_y}{\partial e_u} \cdot u(t) \cdot (1 - u(t)) \cdot v_j h_j \cdot (1 - h_j) \cdot x_i$$

$$\frac{\partial E(t)}{\partial w_{ji}} = \sigma^1 \cdot v_j \cdot h_j \cdot (1 - h_j) \cdot x_i \cdot \frac{\partial e_y}{\partial e_u} \quad (84)$$

$$\sigma_j^2 = \sigma^1 \cdot v_j \cdot h_j \cdot (1 - h_j) \quad (85)$$

$$\frac{\partial E(t)}{\partial w_{ji}} = \sigma_j^2 \cdot x_i \cdot \frac{\partial e_y}{\partial e_u} \quad (86)$$

Obtidos os gradientes da função erro, torna-se possível estabelecer as equações de atualização dos coeficientes de ponderação da RNA. Inicialmente, os ajustes dos pesos das camadas de saída e escondida são realizados pelas Equações (87) e (88), nas quais o incremento aplicado aos coeficientes depende da taxa de aprendizagem η e do ganho equivalente da planta.

$$v_j(t+1) = v_j(t) + \left(\eta \frac{\partial e_y}{\partial e_u} \right) \sigma^1 h_j \quad (87)$$

$$w_{ji}(t + 1) = w_{ji}(t) + \left(\eta \frac{\partial e_y}{\partial e_u} \right) \sigma_j^2 x_i \quad (88)$$

Entretanto, conforme discutido por Cui e Shin (1992), a magnitude do ganho equivalente da planta geralmente não é conhecida. Nessas condições, demonstra-se que a convergência do algoritmo depende apenas do conhecimento do sinal desse ganho, podendo sua magnitude ser incorporada ao coeficiente de aprendizagem sem comprometer a estabilidade do processo de adaptação.

Considerando essa hipótese, as equações de atualização dos coeficientes de ponderação podem ser reescritas conforme as Equações (89) e (90), que constituem a formulação utilizada nesta pesquisa para adaptação dos pesos da RNA.

$$v_j(t + 1) = v_j(t) + \eta \text{sign} \left(\frac{\partial e_y}{\partial e_u} \right) \sigma^1 h_j \quad (89)$$

$$w_{ji}(t + 1) = w_{ji}(t) + \eta \text{sign} \left(\frac{\partial e_y}{\partial e_u} \right) \sigma_j^2 x_i \quad (90)$$

As Equações (89) e (90) representam o mecanismo de adaptação empregado durante o treinamento da rede neural. Em vez de caracterizarem apenas um procedimento convencional de treinamento, essas expressões passam a representar as equações de adaptação do controlador PID-Neural, permitindo que os coeficientes de ponderação sejam continuamente ajustados em função do erro de regulação observado durante a operação do sistema.

As análises realizadas ao longo do desenvolvimento deste trabalho evidenciaram que a utilização dessas equações proporciona elevado grau de robustez ao controlador. Observou-se que, em uma estrutura de controle em malha fechada, não é necessariamente requerido um treinamento prévio da RNA, uma vez que os próprios coeficientes de ponderação podem ser ajustados continuamente durante a operação do controlador, conduzindo o erro de regulação a valores próximos de zero.

Além disso, verificou-se que o desempenho dinâmico do sistema depende diretamente do valor adotado para o coeficiente de aprendizagem. Valores elevados de η tendem a produzir respostas mais rápidas, embora possam aumentar o sobresinal da resposta transitória.

Em contrapartida, valores menores resultam em respostas mais amortecidas, favorecendo a estabilidade do sistema.

Com o objetivo de conciliar rapidez de resposta e estabilidade, adotou-se um coeficiente de aprendizagem variável, definido pela Equação (91).

$$\eta' = \eta + \alpha \cdot \text{abs}(e_y) \quad (91)$$

Nessa estratégia, um valor básico para o coeficiente de aprendizagem é acrescido de um termo proporcional à magnitude do erro de regulação. Dessa forma, quando o sistema apresenta erros elevados, o valor efetivo de η aumenta, acelerando o processo de adaptação da rede neural. À medida que o erro diminui, o coeficiente de aprendizagem retorna gradativamente ao seu valor nominal, proporcionando uma convergência mais suave e reduzindo oscilações próximas ao ponto de operação.

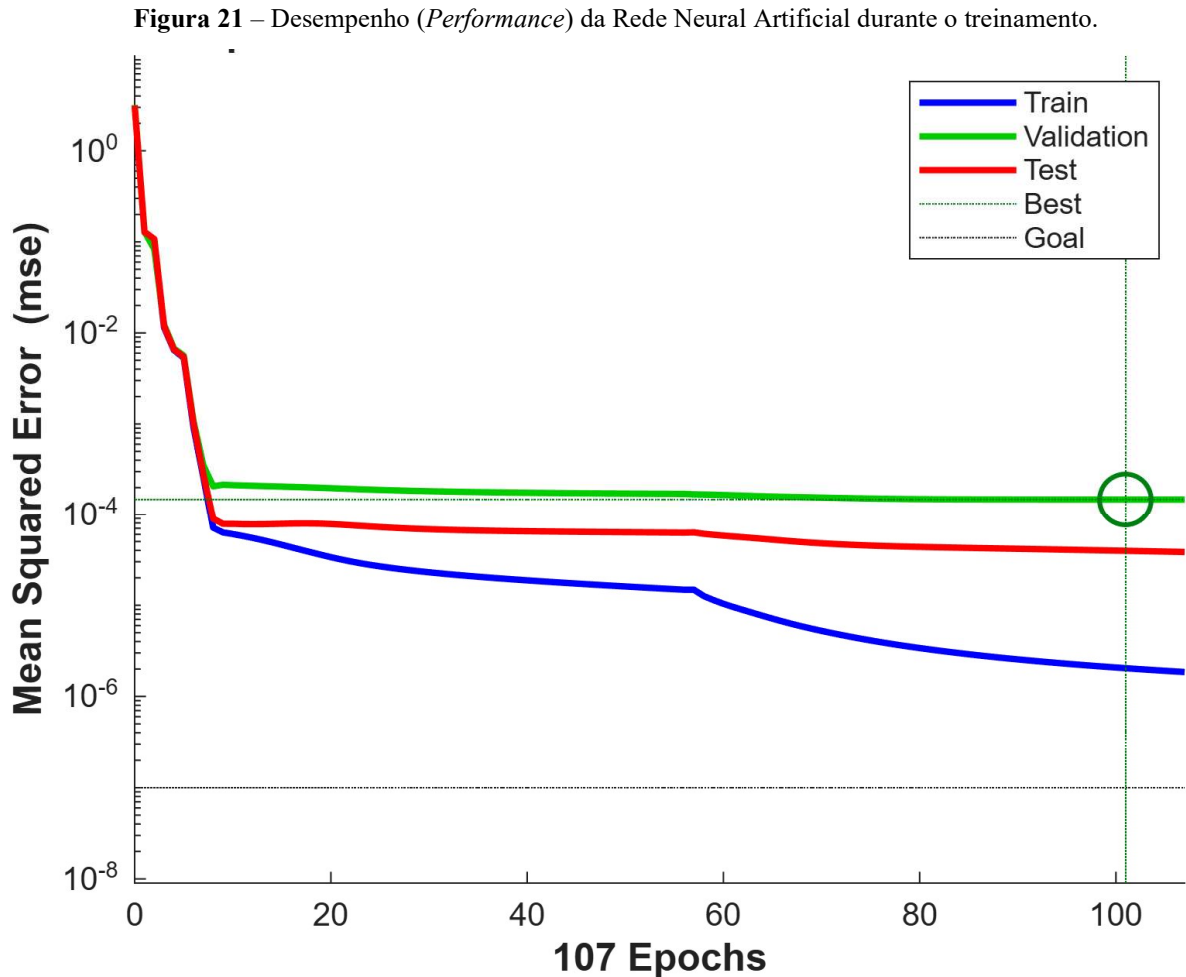
Concluído o processo de treinamento, a rede neural passou a representar adequadamente a relação não linear entre os estados do quadrotor e a parcela residual do sinal de controle. A configuração obtida foi posteriormente submetida à etapa de validação apresentada na Seção 4.6.4 e, após sua integração ao controlador PID, utilizada nas simulações comparativas discutidas no Capítulo 6.

4.6.4 Validação da rede neural

Concluído o processo de treinamento, a Rede Neural Artificial foi submetida à etapa de validação com o objetivo de verificar sua capacidade de generalização para dados não utilizados durante o aprendizado. Para isso, a base de dados foi dividida em conjuntos independentes de treinamento, validação e teste, permitindo avaliar simultaneamente a convergência do algoritmo de otimização, a qualidade da aproximação funcional e a capacidade da RNA em representar a dinâmica não linear do quadrotor.

A validação foi realizada por meio da análise do erro quadrático médio (*Mean Squared Error – MSE*), do acompanhamento do estado do treinamento, da regressão entre os valores estimados e desejados e da distribuição estatística dos erros produzidos pela rede neural. Em conjunto, esses indicadores forneceram evidências quantitativas sobre a qualidade do treinamento e subsidiaram a seleção da configuração final incorporada ao controlador PID-

Neural. A Figura 21 apresenta a evolução do erro quadrático médio (*Performance*) durante o treinamento da Rede Neural Artificial para os conjuntos de treinamento, validação e teste.

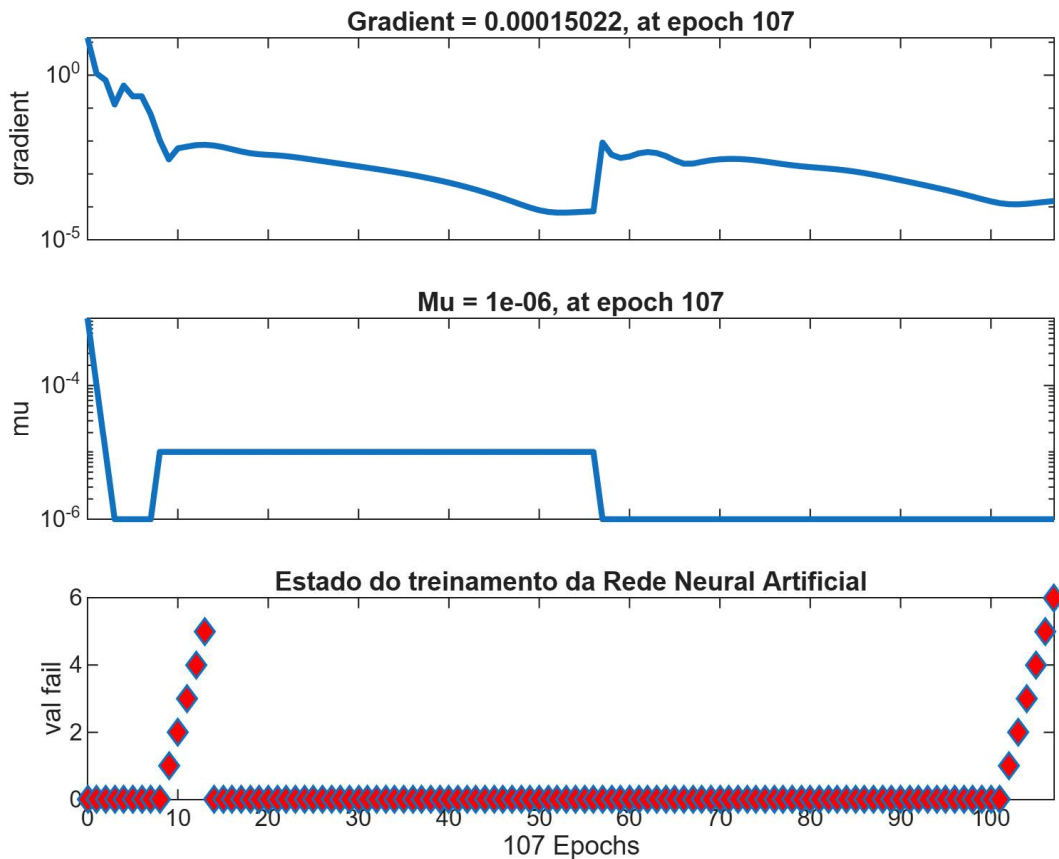


Fonte: Elaborada pelo autor (2014).

Conforme apresentado na Figura 21, observa-se uma redução acentuada do erro nas primeiras épocas de treinamento, seguida por uma convergência gradual até a estabilização das curvas. O treinamento foi encerrado na época 107, sendo identificado o melhor desempenho de validação aproximadamente na época 100, quando o erro quadrático médio atingiu cerca de $(1,4 \cdot 10^{-4})$. A curva de treinamento continuou reduzindo o erro até valores próximos de $2,0 \cdot 10^{-6}$ enquanto as curvas de validação e teste permaneceram praticamente constantes, indicando que a rede preservou sua capacidade de generalização durante o processo de aprendizagem. Esse comportamento demonstra que o algoritmo convergiu adequadamente sem apresentar evidências relevantes de sobreajuste (*overfitting*), produzindo uma solução com bom compromisso entre precisão e capacidade de generalização.

A Figura 22 apresenta o estado do treinamento (*Training State*), incluindo a evolução do gradiente da função custo, do parâmetro de amortecimento (μ) do algoritmo de *Levenberg–Marquardt* e do número de falhas consecutivas de validação (*Validation Checks*).

Figura 22 – Estado do treinamento (*Training State*) da Rede Neural Artificial.



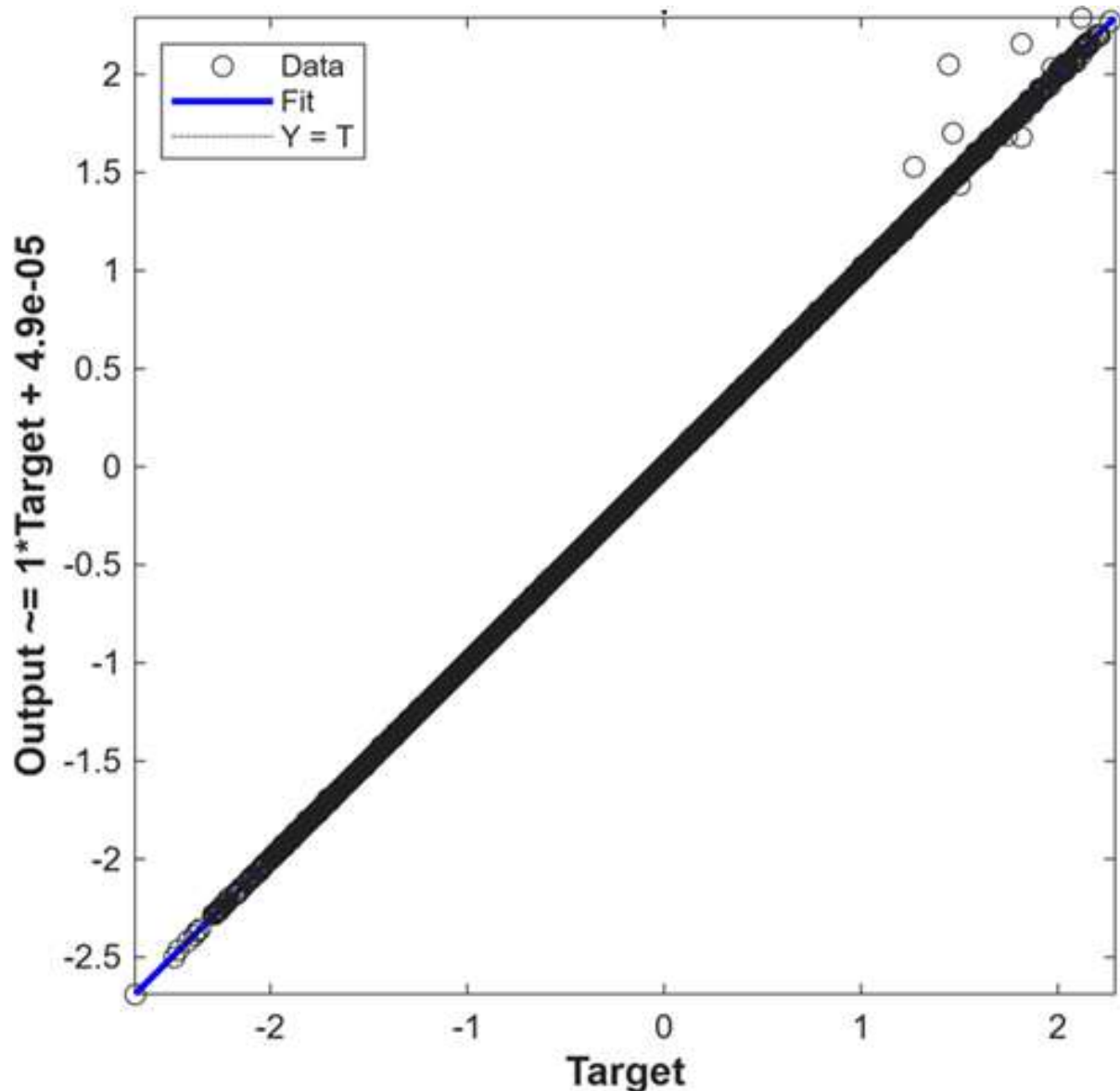
Fonte: Elaborada pelo autor (2014).

Os resultados apresentados na Figura 22 evidenciam a estabilidade do processo de treinamento ao longo das 107 épocas executadas. Observa-se que o gradiente da função custo foi progressivamente reduzido, atingindo aproximadamente $(1,5022 \cdot 10^{-4})$ na última época, indicando que o algoritmo convergiu para uma região de mínimo da função objetivo. Verifica-se ainda que o parâmetro de amortecimento do algoritmo de *Levenberg–Marquardt* permaneceu reduzido, estabilizando-se em $(\mu = 10^{-6})$ durante a fase final do treinamento, comportamento característico de uma convergência satisfatória.

Outro aspecto importante refere-se ao critério de parada adotado. O treinamento foi interrompido após atingir seis falhas consecutivas de validação (*Validation Checks* = 6), caracterizando o mecanismo de *early stopping*. Esse procedimento impede que a rede continue ajustando seus parâmetros exclusivamente aos dados de treinamento, preservando sua capacidade de generalização e reduzindo a possibilidade de sobreajuste.

A Figura 23 apresenta a regressão entre os valores estimados pela Rede Neural Artificial e os respectivos valores desejados utilizados durante o treinamento.

Figura 23 – Regressão (*Regression*) entre as saídas desejadas e estimadas pela Rede Neural Artificial.



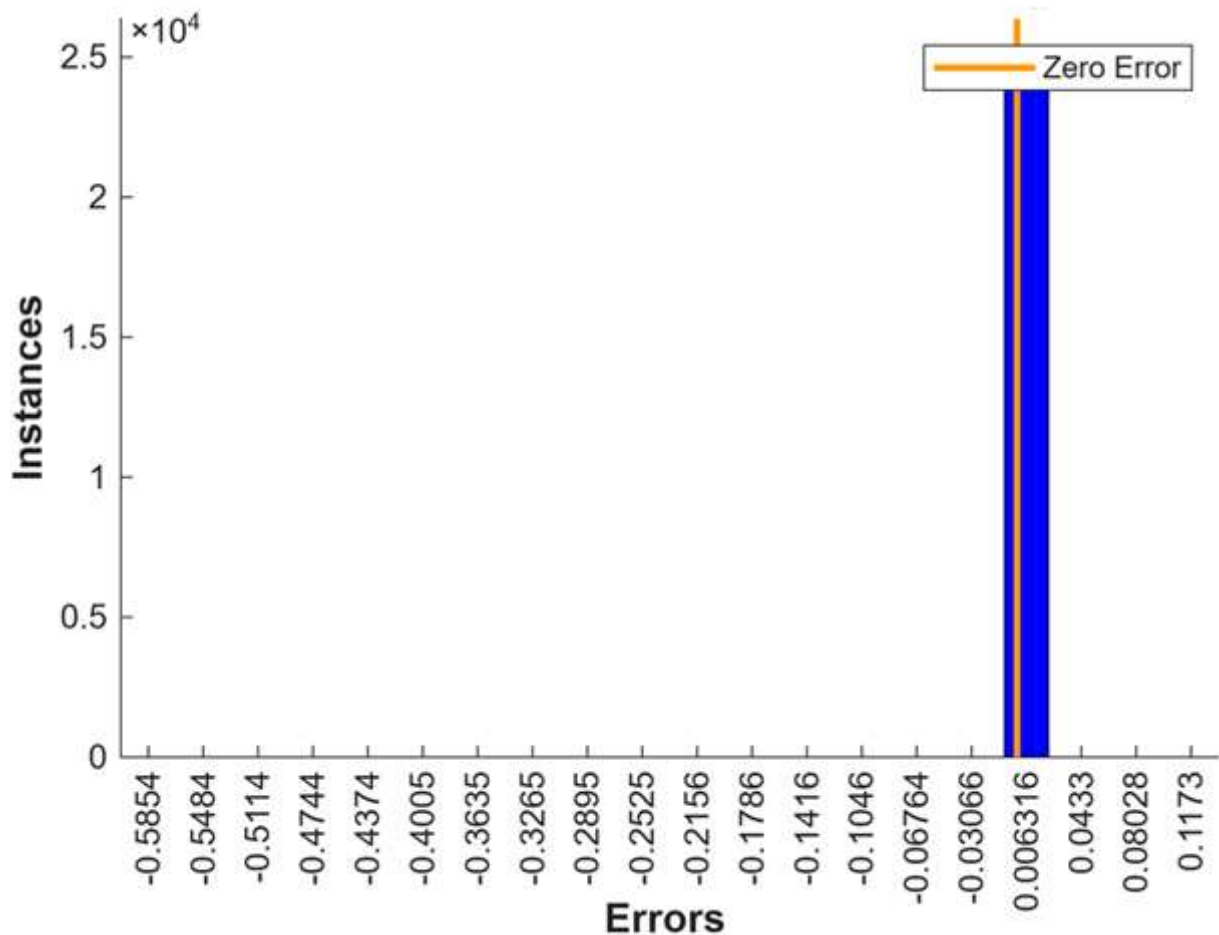
Fonte: Elaborada pelo autor (2014).

A análise da Figura 23 demonstra elevada concordância entre as saídas produzidas pela RNA e os valores de referência utilizados durante o treinamento. Observa-se que praticamente todas as amostras encontram-se distribuídas ao longo da reta identidade ($Y = T$), enquanto a reta de regressão apresenta inclinação praticamente unitária e intercepto reduzido, da ordem de $4,9 \cdot 10^{-5}$. A pequena dispersão observada em torno da reta de referência evidencia a elevada capacidade de aproximação funcional da rede neural, indicando que a arquitetura

adotada foi capaz de representar adequadamente a relação não linear entre os estados do quadrotor e a correção residual aplicada ao controlador PID.

A Figura 24 apresenta o histograma dos erros produzidos pela Rede Neural Artificial após a conclusão do treinamento supervisionado.

Figura 23 – Histograma dos erros (*Error Histogram*) da Rede Neural Artificial.



Fonte: Elaborada pelo autor (2014).

Conforme observado na Figura 24, a distribuição dos erros encontra-se fortemente concentrada nas proximidades do valor zero, evidenciando que a maior parte das estimativas produzidas pela RNA apresenta reduzida diferença em relação aos valores desejados. Embora o histograma, por si só, não permita afirmar que os erros seguem uma distribuição estatística específica, a elevada concentração das amostras em torno da origem indica baixa dispersão e pequena magnitude dos erros residuais. Esse resultado complementa as análises obtidas por

meio do erro quadrático médio e da regressão linear, reforçando a consistência do treinamento realizado.

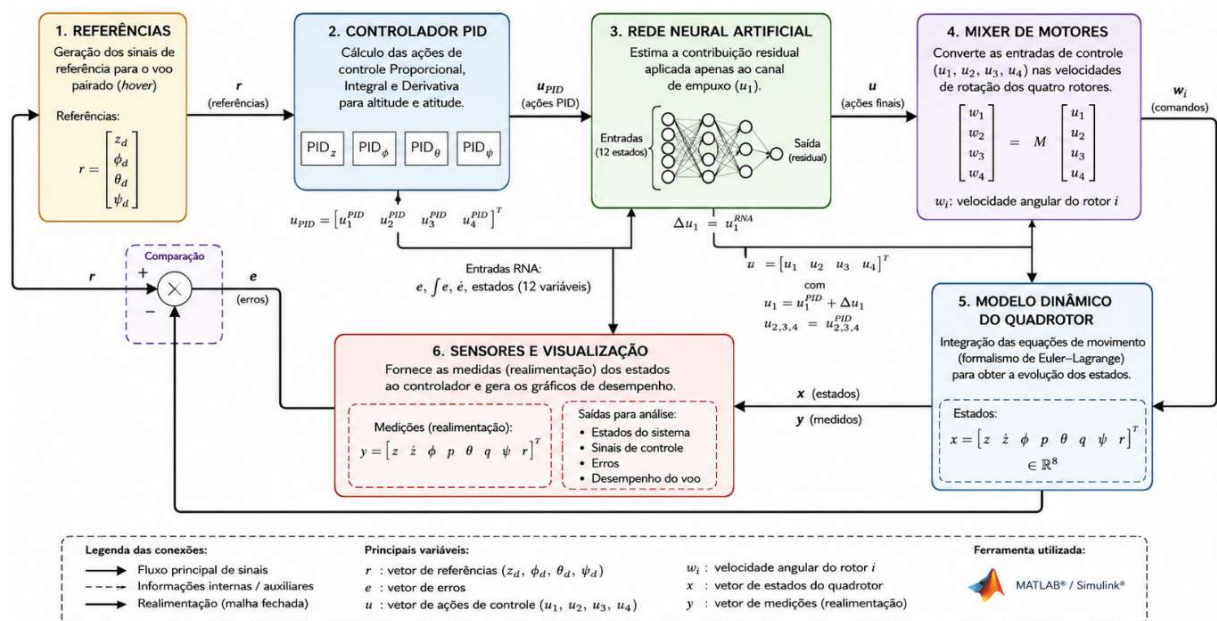
4.7 Estrutura computacional do sistema de controle

Após o desenvolvimento da arquitetura do controlador PID-Neural e da validação da Rede Neural Artificial, tornou-se necessária sua implementação em ambiente computacional para realização das simulações de voo pairado. Nesta pesquisa foi utilizado o ambiente *MATLAB/Simulink* devido à disponibilidade de ferramentas específicas para modelagem dinâmica, implementação de algoritmos de controle e simulação de sistemas não lineares.

A implementação computacional foi concebida de forma a representar, de maneira organizada, todas as etapas envolvidas no processo de estabilização do quadrotor. Para isso, a arquitetura foi subdividida em módulos funcionais responsáveis pela geração das referências, processamento das leis de controle, distribuição dos sinais para os atuadores, modelagem da dinâmica da aeronave, aquisição das variáveis de estado e visualização dos resultados.

Essa organização favoreceu a reutilização dos componentes desenvolvidos, simplificou a validação individual de cada módulo e permitiu a realização de diferentes experimentos computacionais com alterações mínimas na estrutura geral do modelo. A Figura 25 ilustra a organização geral dessa arquitetura computacional.

Figura 25 – Organização computacional do controlador PID-Neural implementado em *MATLAB/Simulink*.



A implementação computacional foi organizada em seis módulos responsáveis pela geração das referências, cálculo das ações de controle, estimação neural, distribuição dos comandos aos motores, modelagem dinâmica do quadrotor e aquisição dos estados do sistema. O Quadro 6 apresenta a organização desses módulos e suas respectivas funções. Essa estrutura modular facilita a integração dos componentes, simplifica a manutenção do modelo computacional e permite a validação individual de cada módulo antes da realização das simulações em malha fechada.

Quadro 6: Organização dos módulos computacionais

Módulo	Função
Referências	Geração dos sinais de referência para altitude e atitude.
Controlador PID	Cálculo das ações proporcional, integral e derivativa.
Rede Neural Artificial	Estimativa da contribuição residual aplicada ao canal de empuxo.
Mixer de motores	Conversão das entradas de controle em velocidades dos quatro rotores.
Modelo Dinâmico	Integração das equações diferenciais do quadrotor utilizando Euler-Lagrange.
Sensores e Visualização	Realimentação dos estados do sistema e geração dos gráficos de desempenho.

Fonte: Elaborada pelo autor (2014).

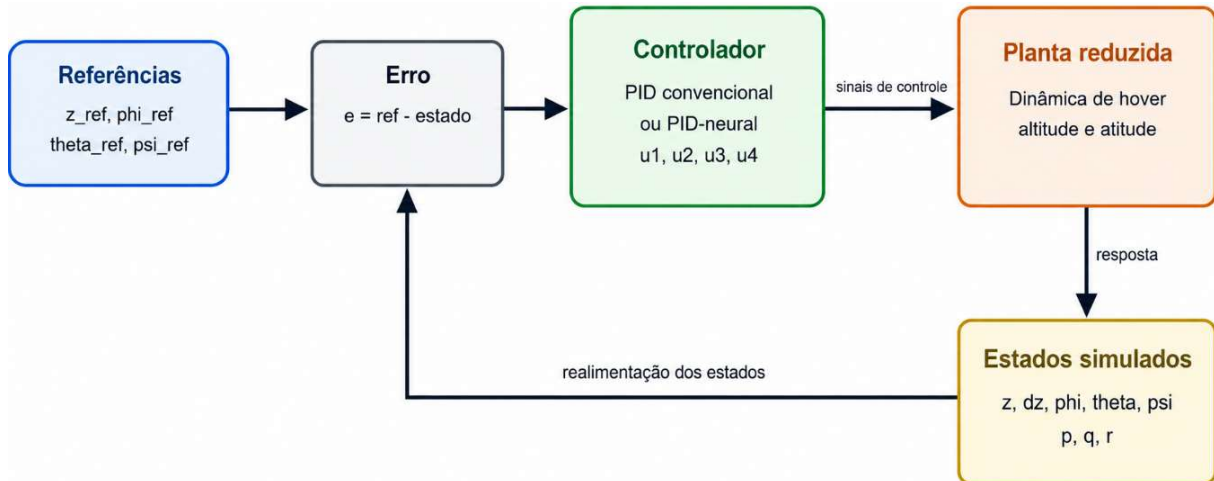
Conforme apresentado no Quadro 6, a implementação computacional foi organizada em módulos independentes. Essa estratégia permitiu que alterações em um componente fossem realizadas sem modificar a estrutura dos demais blocos, simplificando o desenvolvimento e os testes realizados durante a pesquisa.

4.7.1 Organização da malha fechada

A Figura 26 apresenta a arquitetura da simulação em malha fechada adotada neste trabalho para a estabilização de um quadrotor em voo pairado. Nessa estrutura, os estados estimados da aeronave são continuamente realimentados ao controlador, que calcula, a cada passo de integração, os sinais de controle necessários para minimizar o erro entre os estados medidos e as referências de voo. Esse mecanismo de realimentação permite compensar

continuamente as variações da dinâmica do sistema, contribuindo para a estabilidade, o acompanhamento da trajetória de referência e o desempenho do controlador ao longo da simulação.

Figura 26 – Estrutura da simulação em malha fechada para voo pairado.



Fonte: Elaborado pelo autor (2014).

Inicialmente, o bloco de referências fornece os valores desejados para altitude, rolagem, arfagem e guinada. Esses sinais são comparados com os estados atuais do veículo, produzindo os sinais de erro utilizados pelo controlador PID.

Posteriormente, a Rede Neural Artificial calcula uma contribuição residual destinada exclusivamente ao canal de empuxo u_1 , preservando os torques u_2 , u_3 e u_4 sob responsabilidade exclusiva do controlador PID. Essa configuração foi definida após os estudos apresentados na Seção 4.5 e mostrou-se mais adequada para manter a estabilidade do sistema.

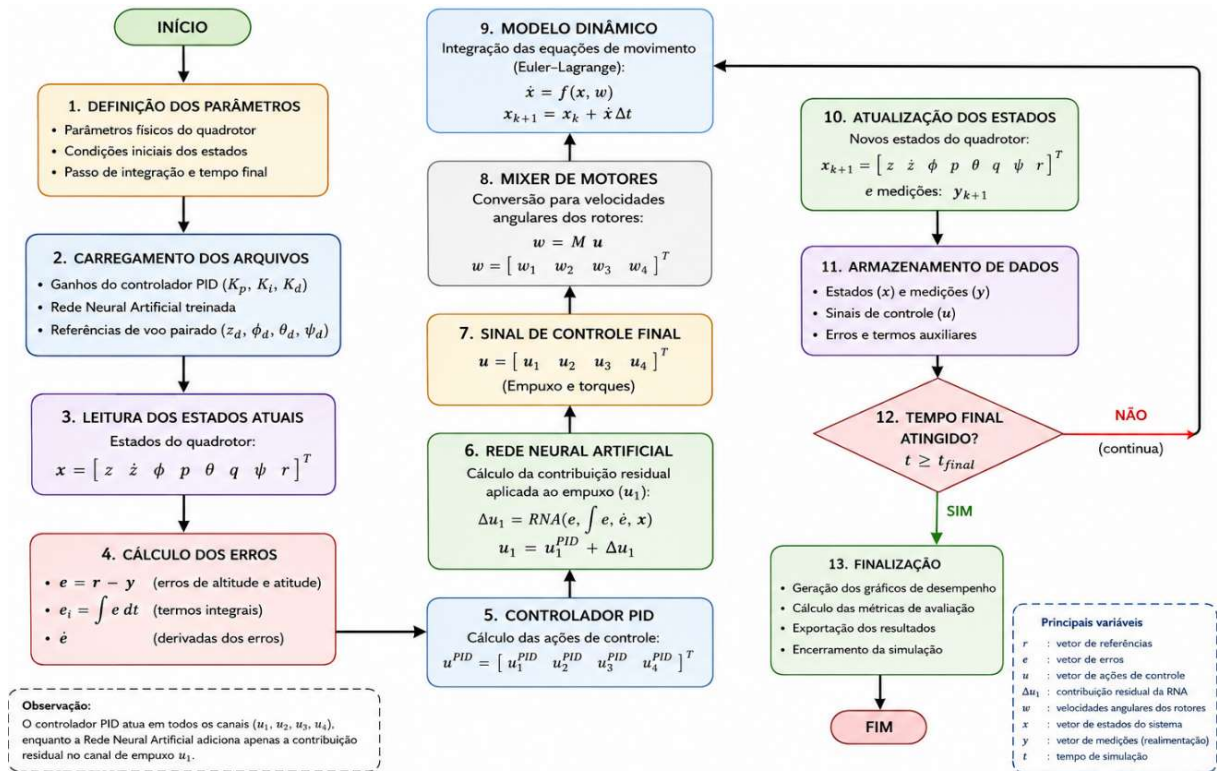
4.7.2 Fluxo operacional da simulação

A implementação computacional desenvolvida para a realização das simulações segue uma sequência lógica de processamento que integra os módulos de inicialização, controle, dinâmica do quadrotor e análise dos resultados. Com o objetivo de sintetizar essa sequência de operações, a Figura 27 apresenta o fluxograma operacional da simulação, destacando as principais etapas executadas durante cada ciclo de integração numérica. Inicialmente são carregados os parâmetros físicos do modelo, as condições iniciais e as referências de voo; em

seguida, são calculadas as ações de controle, atualizados os estados do quadrotor e armazenados os dados necessários para posterior análise.

Essa representação facilita a compreensão do funcionamento global da implementação e evidencia a interação entre os diferentes componentes que constituem o ambiente de simulação utilizado nesta pesquisa.

Figura 27 – Fluxo operacional da simulação do controlador PID-Neural.



Fonte: Elaborada pelo autor (2014).

O processo inicia com a definição das condições iniciais da simulação e dos parâmetros físicos do quadrotor. Em seguida, são carregados os ganhos do controlador PID e a Rede Neural Artificial previamente treinada.

Durante cada passo de integração, os estados do quadrotor são atualizados pelo modelo dinâmico, enquanto o controlador calcula os sinais de controle necessários para reduzir os erros de altitude e atitude. A RNA estima continuamente a contribuição residual aplicada ao empuxo, produzindo o sinal final enviado ao mixer dos motores.

Esse procedimento é repetido até o término da simulação, quando são armazenados os estados do sistema, os sinais de controle e as métricas utilizadas na análise comparativa entre os controladores.

4.7.3 Organização dos arquivos *MATLAB*

Para facilitar o desenvolvimento e permitir a reprodução dos resultados obtidos nesta pesquisa, os scripts *MATLAB* foram organizados conforme apresentado no Quadro 7.

Quadro 7: Organização dos principais arquivos do projeto

Arquivo	Finalidade
parametrosDrone.m	Define os parâmetros físicos do quadrotor.
referencias.m	Define as referências do voo pairado.
controladorPID.m	Implementa o controlador PID convencional.
controladorPIDNeuralResidual.m	Implementa o controlador PID-Neural residual.
dinamicaQuadrotor.m	Resolve o modelo dinâmico do quadrotor.
treinarRNA.m	Treinamento da Rede Neural Artificial.
gerarDataset.m	Construção da base de dados para treinamento.
main_PID.m	Simulação utilizando apenas PID.
main_PIDNeural.m	Simulação utilizando o controlador PID-Neural.

Fonte: Elaborada pelo autor (2014).

Essa organização modular permitiu separar claramente as etapas de modelagem, treinamento, controle e análise dos resultados, facilitando a manutenção do código e a reprodução dos experimentos computacionais desenvolvidos nesta pesquisa.

A implementação computacional desenvolvida neste capítulo foi estruturada de forma modular, permitindo substituir facilmente diferentes estratégias de controle sem alterações significativas na dinâmica da planta. Essa característica possibilitou comparar diretamente o controlador PID convencional e a estratégia PID-Neural proposta nos experimentos apresentados no capítulo seguinte.

5 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados o desenvolvimento e a implementação dos controladores utilizados para a estabilização do quadrotor em voo pairado, com base na

modelagem matemática e nos fundamentos de controle discutidos anteriormente. São descritas as estratégias de controle PID clássico, PID adaptativo e RNA híbrida associada ao PID, considerando os estados relevantes da planta, os sinais de erro, as ações de controle e as restrições físicas de empuxo e torque. Além disso, são descritos os procedimentos de treinamento e validação da rede neural, bem como sua integração à malha de controle, estabelecendo a base computacional necessária para as simulações e análises comparativas apresentadas no capítulo seguinte.

5.1 Considerações iniciais

Este capítulo apresenta a análise dos resultados obtidos por meio de simulações computacionais aplicadas à estabilização em voo pairado de um quadrotor utilizando o controlador PID-Neural proposto. O objetivo principal é avaliar o comportamento dinâmico do sistema diante da ação de controle, observando sua capacidade de manter a estabilidade em torno da condição de *hover* e comparando seu desempenho com a estratégia PID convencional.

Embora o modelo geral do quadrotor tenha sido desenvolvido considerando seus seis graus de liberdade, as simulações realizadas neste capítulo utilizam uma representação reduzida da dinâmica, concentrada nos estados de altitude e atitude. Essa escolha é adequada ao escopo do trabalho, uma vez que o foco está na estabilização em voo pairado, situação na qual as dinâmicas verticais e rotacionais são as mais relevantes para a análise do desempenho do controlador.

A partir dessa configuração, são apresentados os cenários de simulação, os parâmetros adotados, as respostas temporais do sistema e os indicadores quantitativos utilizados para comparar o desempenho dos controladores. Dessa forma, busca-se verificar se a inclusão da estrutura neural contribui para melhorar a resposta transitória, reduzir oscilações e aumentar a capacidade de estabilização do quadrotor em torno da condição de equilíbrio.

Além da comparação inicial entre o controlador PID e o controlador PID-Neural originalmente implementado, este capítulo também contempla uma avaliação complementar baseada em uma estratégia PID-Neural residual. Essa etapa foi incluída após a verificação de que a configuração inicial da rede neural não produzia alteração significativa no desempenho em relação ao PID. Posteriormente, a contribuição residual foi restringida ao canal de empuxo u_1 , preservando os torques de atitude sob ação PID, com o objetivo de melhorar a resposta vertical sem comprometer a estabilização angular.

5.2 Ambiente de simulação

As simulações deste trabalho foram desenvolvidas com o objetivo de avaliar o desempenho do controlador PID-Neural proposto na estabilização em voo pairado de um quadrotor. Para isso, foi utilizado um ambiente computacional capaz de representar a dinâmica da planta, aplicar as estratégias de controle e registrar as respostas temporais das variáveis de interesse.

Embora a modelagem geral do quadrotor tenha sido apresentada considerando seis graus de liberdade, neste capítulo adota-se uma representação reduzida da dinâmica, direcionada à condição de *hover*. Essa simplificação é adequada ao escopo do trabalho, pois a análise concentra-se na capacidade do controlador de manter o quadrotor estabilizado em torno de uma condição de equilíbrio.

No modelo reduzido, são consideradas principalmente as variáveis associadas à altitude e à atitude do veículo. Assim, a dinâmica vertical, relacionada ao eixo z , e a dinâmica rotacional, representada pelos ângulos de Euler ϕ , θ e ψ , são preservadas por serem essenciais à manutenção do voo pairado. As componentes translacionais horizontais, associadas aos eixos x e y , não constituem o foco da análise, uma vez que o objetivo não é o rastreamento de trajetórias, mas sim a estabilização em *hover*.

O ambiente de simulação contempla a planta dinâmica do quadrotor, os sinais de referência e os controladores avaliados. O controlador PID convencional é utilizado como base comparativa, enquanto o controlador PID-Neural corresponde à estratégia proposta. A partir dos erros entre as referências e as variáveis simuladas, os controladores geram os sinais de controle necessários para corrigir desvios de altitude e atitude.

Além das respostas temporais dos estados controlados, também são analisados os sinais de controle, permitindo verificar não apenas a capacidade de estabilização, mas também a suavidade e a coerência dos comandos aplicados. Dessa forma, o ambiente adotado fornece uma base adequada para comparar o desempenho das estratégias de controle e avaliar a viabilidade do PID-Neural na estabilização em voo pairado.

O ambiente de simulação foi implementado em *MATLAB/Simulink*, utilizando funções específicas para a definição dos parâmetros físicos do quadrotor, das referências de voo pairado, da planta reduzida e dos controladores avaliados. As simulações foram conduzidas em malha fechada, permitindo analisar a resposta temporal dos estados de altitude e atitude, bem como os sinais de controle aplicados à planta.

O ambiente de simulação foi estruturado para permitir tanto a avaliação da configuração inicial do controlador PID-Neural quanto os testes complementares com a rede neural residual aplicada ao canal de empuxo. Em todos os casos, foram preservadas as mesmas condições gerais de simulação, de modo a permitir comparação coerente entre as estratégias de controle.

5.2.1 Modelo reduzido para simulação em voo pairado

Embora a modelagem geral do quadrotor tenha sido apresentada em termos de seis graus de liberdade, as simulações deste capítulo consideram uma versão reduzida da dinâmica de voo pairado. Essa simplificação é pertinente ao objetivo do trabalho, pois a análise está concentrada na estabilização do veículo em torno da condição de *hover*, preservando as dinâmicas verticais e rotacionais de maior interesse para o controle.

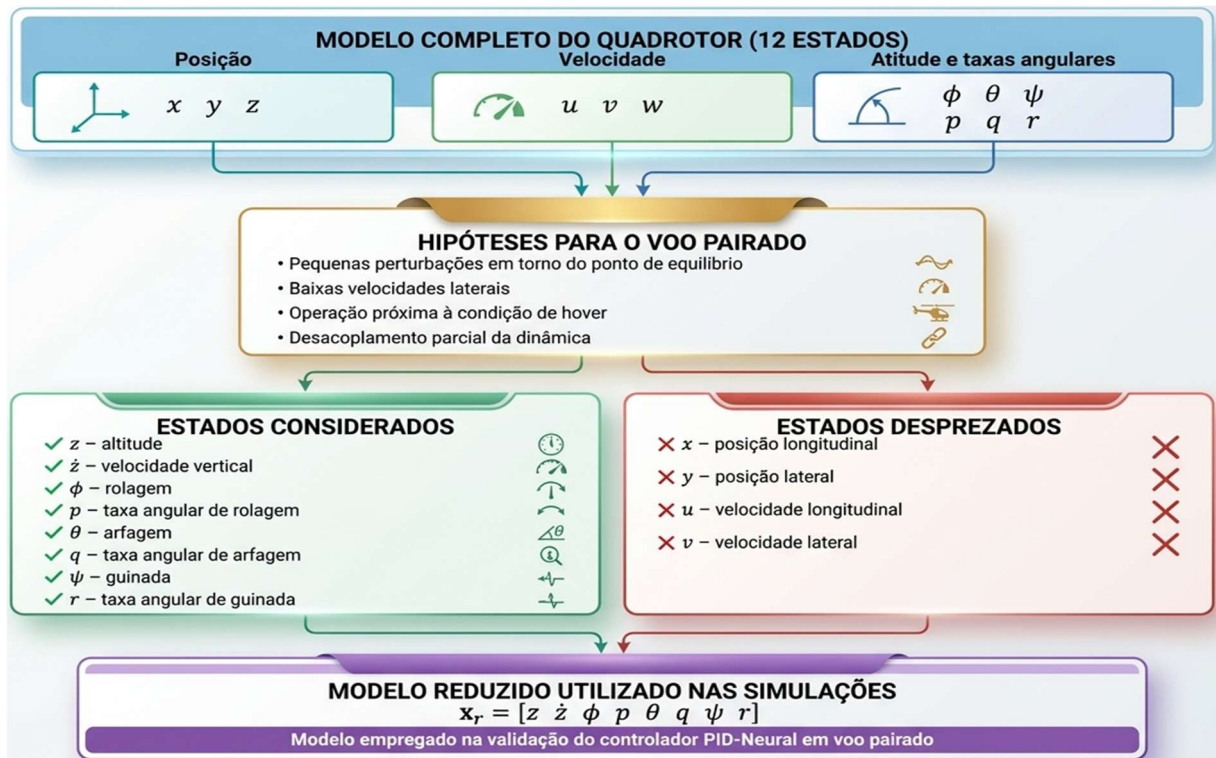
No modelo reduzido, são considerados os estados associados à altitude e à atitude do quadrotor. A dinâmica vertical é representada pelo deslocamento no eixo z e por sua respectiva velocidade, enquanto a dinâmica de atitude é descrita pelos ângulos de Euler ϕ , θ e ψ , juntamente com as velocidades angulares correspondentes. Esses estados são essenciais para avaliar a capacidade do controlador de manter o veículo estabilizado, com altitude próxima ao valor de referência e ângulos de atitude próximos de zero.

Nos testes complementares, foi utilizada a mesma estrutura reduzida, porém com variações paramétricas controladas da planta, permitindo avaliar a sensibilidade dos controladores frente a alterações de massa, inércia e amortecimento.

A utilização do modelo reduzido também favoreceu a implementação da estratégia de controle proposta, permitindo que o controlador PID-Neural atuasse diretamente sobre as variáveis críticas para a estabilização em voo pairado. Em cada passo de simulação, os estados do quadrotor são comparados às referências estabelecidas, produzindo os sinais de controle que alimentam a planta e fecham a malha de realimentação. Essa estrutura reproduz o funcionamento do sistema de controle desenvolvido nesta pesquisa e constitui a base para todos os experimentos computacionais apresentados neste capítulo.

A Figura 28 apresenta a organização da simulação em malha fechada utilizada durante as avaliações.

Figura 28 – Modelo reduzido adotado nas simulações em voo pairado.



Fonte: Elaborado pelo autor (2014).

As dinâmicas translacionais horizontais, associadas aos eixos x e y , não são o foco das simulações deste capítulo, uma vez que o objetivo não é analisar o deslocamento lateral ou o rastreamento de trajetória, mas sim a permanência do quadrotor em voo pairado. Dessa forma, o modelo reduzido permite uma avaliação mais direta da estratégia de controle proposta, reduzindo a complexidade da simulação sem eliminar os principais efeitos dinâmicos envolvidos na estabilização em *hover*.

Assim, a utilização do modelo reduzido fornece uma base adequada para comparar o desempenho do controlador PID convencional e do controlador PID-Neural, permitindo observar a resposta do sistema em relação à altitude, à atitude e aos sinais de controle gerados durante a estabilização.

A Figura 28 apresenta a arquitetura completa empregada nas simulações realizadas neste trabalho. Nela observam-se a planta dinâmica do quadrotor, o controlador, o gerador de referências e os blocos responsáveis pela aquisição dos sinais utilizados na avaliação do desempenho do sistema em voo pairado.

5.2.2 Parâmetros físicos e condições iniciais

Os parâmetros físicos adotados nas simulações foram definidos com base no modelo do quadrotor utilizado no projeto, correspondente ao Parrot AR.Drone 2.0. Esses parâmetros contemplam as grandezas necessárias para representar a dinâmica vertical e rotacional do veículo em voo pairado, incluindo massa, aceleração da gravidade, momentos de inércia, limites de empuxo e torque, além dos coeficientes de amortecimento considerados no modelo simplificado. Os valores utilizados são apresentados na Tabela 4.

Tabela 4: Parâmetros físicos utilizados na simulação.

Parâmetro	Símbolo	Valor	Unidade
Massa do quadrotor	m	0,420	kg
Aceleração da gravidade	g	9,81	m/s ²
Momento de inércia em x	I _{xx}	$2,24 \times 10^{-3}$	kg.m ²
Momento de inércia em y	I _{yy}	$2,98 \times 10^{-3}$	kg.m ²
Momento de inércia em z	I _{zz}	$4,80 \times 10^{-3}$	kg.m ²
Empuxo máximo	maxThrust	12	N
Empuxo mínimo	minThrust	0	N
Torque máximo	maxTorque	2	N.m
Amortecimento vertical	k _z	0,25	-
Amortecimento em rolagem	k _{phi}	0,08	-
Amortecimento em arfagem	k _{theta}	0,08	-
Amortecimento em guinada	k _{psi}	0,05	-

Fonte: Elaborado pelo autor (2014).

Esses parâmetros são fundamentais para determinar a resposta dinâmica da planta reduzida, pois influenciam diretamente a relação entre os sinais de controle aplicados e o movimento resultante do quadrotor. A massa e a gravidade estão associadas ao equilíbrio vertical e à manutenção da altitude, enquanto os momentos de inércia determinam a sensibilidade do veículo aos torques de rolagem, arfagem e guinada. Já os limites de empuxo e torque representam restrições impostas aos sinais de controle, e os coeficientes de amortecimento introduzem efeitos dissipativos na dinâmica simplificada de *hover*.

Para os testes complementares com a estratégia PID-Neural residual, alguns parâmetros físicos foram alterados de forma controlada em relação aos valores nominais, especialmente a

massa, os momentos de inércia e os coeficientes de amortecimento. Essas variações não substituem o modelo nominal adotado no trabalho, mas permitem avaliar o comportamento dos controladores diante de incertezas paramétricas, conforme apresentado na Tabela 5.

Tabela 5: Variações paramétricas utilizadas nos testes complementares.

Cenário	Descrição	Fator massa	Massa (kg)	Fator inércia	Fator amortecimento
massa_menor_20	Massa reduzida em 20%	0,80	0,336	1,00	1,00
massa_maior_20	Massa aumentada em 20%	1,20	0,504	1,00	1,00
inerCIAS_maiores	Inércias aumentadas em 20%	1,00	0,420	1,20	1,00
amortecimento_menor	Amortecimento reduzido em 30%	1,00	0,420	1,00	0,70
combinado	Massa menor, inércias maiores e amortecimento menor	0,80	0,336	1,20	0,70

Fonte: Elaborado pelo autor (2014).

As condições iniciais utilizadas na simulação padrão foram definidas de modo a representar o quadrotor próximo da condição de voo pairado, porém sujeito a desvios iniciais de atitude. Nesse caso, considera-se altitude inicial nula, velocidades iniciais nulas e ângulos iniciais diferentes de zero para rolagem, arfagem e guinada. Essa configuração permite avaliar a capacidade dos controladores de corrigir perturbações iniciais e conduzir o sistema à condição de equilíbrio desejada. Os valores adotados para as condições iniciais são apresentados na Tabela 6.

Tabela 6: Condições iniciais padrão da simulação.

Estado inicial	Símbolo	Valor	Unidade
Altitude inicial	$z(0)$	0	m
Velocidade vertical inicial	$\dot{z}(0)$	0	m/s
Rolagem inicial	$\phi(0)$	8	graus
Velocidade angular em x	$p(0)$	0	rad/s
Arfagem inicial	$\theta(0)$	-5	graus
Velocidade angular em y	$q(0)$	0	rad/s
Guinada inicial	$\psi(0)$	10	graus
Velocidade angular em z	$r(0)$	0	rad/s

Fonte: Elaborado pelo autor (2014).

A condição de referência considerada para o voo pairado corresponde à manutenção da altitude em 1,0 m, com ângulos de atitude nulos, conforme indicado na Tabela 7. Assim, o controlador deve atuar para elevar o quadrotor até a altitude desejada e, simultaneamente, estabilizar os ângulos de rolagem, arfagem e guinada em torno de zero.

Tabela 7: Referências de *hover* usadas na simulação.

Referência	Símbolo	Valor	Unidade
Altitude de referência	z_{ref}	1,0	m
Rolagem de referência	ϕ_{ref}	0	graus/rad
Arfagem de referência	θ_{ref}	0	graus/rad
Guinada de referência	ψ_{ref}	0	graus/rad

Fonte: Elaborado pelo autor (2014).

A definição das mesmas condições iniciais e referências para todos os controladores é essencial para garantir uma comparação coerente entre o PID convencional e o PID-Neural. Dessa forma, as diferenças observadas nas respostas temporais, no erro em regime permanente e no esforço de controle podem ser atribuídas às estratégias de controle avaliadas, e não a variações nas condições de partida da simulação.

Ressalta-se ainda que o projeto possui uma função específica para geração de condições iniciais aleatórias, utilizada em testes complementares e na formação de dados para treinamento ou avaliação. Entretanto, nas simulações padrão executadas pela função *simuladorHover*, foram adotadas as condições iniciais fixas apresentadas na Tabela 4, garantindo repetibilidade e comparação direta entre os controladores.

5.2.3 Configurações numéricas da simulação

As simulações foram realizadas em tempo discreto, adotando-se um passo de amostragem fixo de $T_s = 0,01$ s e tempo total de simulação de $T_f = 20$ s, conforme apresentado na Tabela 6. Com essa configuração, cada ensaio computacional é composto por aproximadamente 2000 iterações, quantidade suficiente para observar o comportamento transitório do sistema e sua tendência de estabilização em torno da condição de voo pairado.

A atualização dos estados da planta reduzida foi realizada por meio de integração numérica discreta, utilizando o método de Euler explícito. Nesse procedimento, a derivada dos estados é calculada a partir da função dinâmica da planta, considerando o estado atual, os sinais de controle gerados pelo controlador e os parâmetros físicos do quadrotor. Em seguida, os estados são atualizados a cada passo de amostragem, permitindo a evolução temporal da simulação.

Em cada instante de tempo, o controlador recebe os estados atuais do quadrotor e as referências de *hover*, calcula os erros de altitude e atitude e gera os sinais de controle correspondentes. Esses sinais são aplicados à planta reduzida, resultando na nova resposta dinâmica do sistema. Esse processo é repetido durante todo o intervalo de simulação, caracterizando a estrutura de controle em malha fechada descrita anteriormente.

As configurações numéricas resumidas na Tabela 8 foram mantidas iguais para todos os controladores avaliados. Dessa forma, a comparação entre o PID convencional e o PID-Neural ocorre sob as mesmas condições de simulação, garantindo que as diferenças observadas nas respostas temporais, nos erros e nos sinais de controle estejam associadas às estratégias de controle, e não a alterações no método de integração, no tempo de amostragem ou na duração do ensaio computacional.

Tabela 8: Configurações numéricas da simulação.

Estado inicial	Símbolo	Valor	Unidade
Tempo de amostragem	T_s	0,01	s
Tempo total de simulação	T_f	20	s
Número de iterações	N	2000	-
Método de integração	-	Euler explícito	-
Tipo de simulação	-	Discreta	-
Estrutura de controle	-	Malha fechada	-

Fonte: Elaborado pelo autor (2014).

A etapa complementar de retreinamento da RNA residual foi realizada no mesmo ambiente computacional, mantendo-se o passo de integração e o tempo total de simulação. A rede residual foi treinada a partir de amostras geradas com variações paramétricas da planta, com o objetivo de estimar uma correção adicional ao sinal de controle produzido pelo PID. As principais configurações utilizadas nessa etapa são apresentadas na Quadro 8.

Quadro 8: Configurações utilizadas no retreinamento da RNA residual.

Parâmetro	Valor
Tipo de rede	Rede neural feedforward para ajuste de função
Função <i>MATLAB/Simulink</i> utilizada	fitnet
Arquitetura	Duas camadas ocultas com 24 e 16 neurônios
Algoritmo de treinamento	Levenberg-Marquardt (trainlm)
Número de amostras	6000
Número de entradas	16
Número de saídas	4
Entradas da rede	Erros de altitude e atitude, velocidades associadas e fatores relativos da planta
Saídas da rede	Correções residuais para u_1 , u_2 , u_3 e u_4
Divisão dos dados	70% treinamento, 15% validação e 15% teste
Número máximo de épocas	600
Meta de desempenho	$1e-7$
Fator de ponderação alpha	1,0
Lei de controle avaliada	$u = u_{PID} + \alpha u_{RNA,res}$
Cenários considerados	Variações de massa, inércias, amortecimento e cenário combinado
Arquivo da rede treinada	rede_residual_pid_neural.mat

Fonte: Elaborado pelo autor (2014).

5.3 Estratégia de controle avaliada

A estratégia de controle avaliada segue a estrutura de malha fechada apresentada anteriormente na Figura 17, na qual as referências de voo pairado são comparadas com os estados simulados, gerando os erros utilizados pelos controladores. A partir desses erros, o controlador PID ou PID-Neural calcula os sinais de controle aplicados à planta reduzida do quadrotor.

Em cada instante de amostragem, os erros de controle são calculados pela diferença entre os valores de referência e os estados atuais do sistema, conforme indicado nas Equações 92 a 95.

$$e_z = Z_{ref} - Z \quad (92)$$

$$e_\phi = \phi_{ref} - \phi \quad (93)$$

$$e_\theta = \theta_{ref} - \theta \quad (94)$$

$$e_\psi = \psi_{ref} - \psi \quad (95)$$

Esses erros são utilizados para gerar os sinais de controle associados ao empuxo total e aos torques de atitude. No controlador PID adotado como base, o sinal de controle da altitude considera a compensação do peso do quadrotor, enquanto os canais de atitude atuam diretamente sobre os torques de rolagem, arfagem e guinada. A forma geral das ações de controle pode ser apresentada conforme a Equação 96 a 99.

$$u_1 = mg + K_{pz}e_z + K_{iz} \int e_z dt - K_{dz}\dot{z} \quad (96)$$

$$u_2 = K_{p\phi}e_\phi + K_{i\phi} \int e_\phi dt - K_{d\phi}\dot{\phi} \quad (97)$$

$$u_3 = K_{p\theta}e_\theta + K_{i\theta} \int e_\theta dt - K_{d\theta}\dot{\theta} \quad (98)$$

$$u_4 = K_{p\psi}e_\psi + K_{i\psi} \int e_\psi dt - K_{d\psi}\dot{\psi} \quad (99)$$

Na implementação utilizada, os termos integrais são computados com limitação para evitar crescimento excessivo, porém os ganhos integrais adotados na simulação padrão são nulos. Assim, o controlador atua predominantemente com ação proporcional e derivativa, mantendo a compensação gravitacional no canal de altitude. Os ganhos empregados no controlador PID são apresentados na Tabela 9.

Tabela 9: Ganhos utilizados no controlador PID.

Canal de controle	Ganho proporcional	Ganho integral	Ganho derivativo
Altitude z	$K_{pz} = 10$	$K_{iz} = 0$	$K_{dz} = 6$
Rolagem ϕ	$K_{p\phi} = 0,8$	$K_{i\phi} = 0$	$K_{d\phi} = 0,15$
Arfagem θ	$K_{p\theta} = 0,8$	$K_{i\theta} = 0$	$K_{d\theta} = 0,15$
Guinada ψ	$K_{p\psi} = 1,2$	$K_{i\psi} = 0$	$K_{d\psi} = 0,20$

Fonte: Elaborado pelo autor (2014).

Já a estratégia PID-Neural utiliza uma RNA como complemento à ação de controle produzida pelo controlador PID. A entrada da rede é composta pelos estados do sistema, erros de controle, integrais dos erros e velocidades associadas à altitude e à atitude. A saída da rede fornece uma estimativa dos sinais de controle, posteriormente combinada com a ação PID.

No projeto implementado, a combinação entre o controle PID e a saída neural é realizada por meio de uma ponderação, conforme a Equação (100).

$$u = (1 - \alpha)u_{PID} + \alpha u_{RNA} \quad (100)$$

em que u_{PID} representa o vetor de controle produzido pelo controlador PID, u_{RNA} representa o vetor de controle estimado pela rede neural e α é o fator de ponderação entre as duas ações. Na simulação considerada, foi utilizado $\alpha = 0,1$, indicando que a ação de controle é composta por 90% da resposta PID e 10% da contribuição neural.

Após o cálculo dos sinais de controle, são aplicadas limitações de empuxo e torque, respeitando os valores máximos e mínimos definidos nos parâmetros físicos da planta. Essa saturação impede que os comandos ultrapassem os limites estabelecidos no modelo de simulação, tornando a análise mais coerente com uma situação de atuação física limitada.

Dessa forma, a avaliação proposta neste capítulo busca verificar se a inclusão da rede neural contribui para melhorar a estabilização em voo pairado, reduzindo oscilações, erro em regime permanente ou esforço de controle quando comparada à estrutura PID utilizada como referência.

Após a análise inicial, verificou-se que a configuração originalmente implementada do controlador PID-Neural não produziu alteração significativa no desempenho em relação ao PID. Em seguida, foi avaliada uma estratégia complementar baseada em uma rede neural

residual, na qual a RNA atua como termo adicional de correção da ação PID. Entretanto, quando a correção neural foi aplicada a todos os canais de controle, observou-se degradação nas respostas angulares. Por esse motivo, a formulação final restringiu a contribuição neural ao canal de empuxo u_1 , mantendo os torques de rolagem, arfagem e guinada definidos pelo controlador PID.

Assim, a ação de controle do PID-Neural residual, na formulação final adotada, pode ser representada pelas Equações (101) a (104):

$$u_1(t) = u_{1,PID}(t) + \alpha \Delta u_{1,RNA,res}(t) \quad (101)$$

$$u_2(t) = u_{2,PID}(t) + \alpha \Delta u_{2,RNA,res}(t) \quad (102)$$

$$u_3(t) = u_{3,PID}(t) + \alpha \Delta u_{3,RNA,res}(t) \quad (103)$$

$$u_4(t) = u_{4,PID}(t) + \alpha \Delta u_{4,RNA,res}(t) \quad (104)$$

em que u_1 representa o sinal de empuxo total aplicado à planta, $u_{1,PID}$ corresponde ao empuxo calculado pelo controlador PID, $u_{1,RNA,res}$ representa a correção residual estimada pela rede neural para o canal vertical, e α é o fator de ponderação associado à contribuição neural. Os sinais u_2 , u_3 e u_4 , associados respectivamente aos torques de rolagem, arfagem e guinada, permanecem definidos pela ação PID, de modo a preservar a estabilização angular do quadrotor.

5.4 Cenários de simulação

Os cenários de simulação foram definidos com o objetivo de avaliar o comportamento do quadrotor em voo pairado sob diferentes estratégias de controle. Para garantir uma comparação adequada, os controladores foram submetidos às mesmas condições de simulação, mantendo-se constantes as referências de *hover*, as condições iniciais e as configurações numéricas apresentadas nas seções anteriores. A primeira etapa corresponde à comparação entre o PID e a configuração inicial do PID-Neural, considerando a condição nominal de voo pairado. A segunda etapa contempla testes complementares com variações

paramétricas da planta e uso da rede neural residual restrita ao canal de empuxo, permitindo avaliar a robustez da estratégia proposta frente a alterações de massa, inércia e amortecimento.

Inicialmente, considerou-se a estabilização do quadrotor a partir de uma condição inicial perturbada. Nesse caso, o sistema parte com altitude inicial nula e ângulos iniciais diferentes de zero, representando uma situação em que o veículo precisa corrigir desvios de atitude e atingir a altitude de referência. A referência adotada corresponde a $z_{ref} = 1,0\text{ m}$ e ângulos de atitude nulos, caracterizando a condição desejada de voo pairado.

A primeira etapa da avaliação consiste em comparar a resposta obtida com o controlador PID convencional/adaptativo e com a configuração inicial do controlador PID-Neural. Os cenários considerados nessa etapa são resumidos na Quadro 9.

Quadro 9: Cenários de simulação avaliados na configuração inicial.

Cenário	Controlador avaliado	Objetivo da simulação	Variáveis analisadas
C1	PID convencional/adaptativo	Avaliar a estabilização em <i>hover</i> com a estratégia PID de referência	Altitude, atitude, erros e sinais de controle
C2	PID-Neural	Avaliar a contribuição da rede neural combinada ao PID	Altitude, atitude, erros e sinais de controle
C3	Comparação PID $\times$ PID-Neural	Comparar o desempenho das estratégias sob as mesmas condições iniciais e referências	Tempo de resposta, oscilações, erro em regime e esforço de controle

Fonte: Elaborado pelo autor (2014).

Após a avaliação da configuração inicial, realizou-se uma segunda etapa de simulações com o controlador PID-Neural residual aplicado exclusivamente ao canal de empuxo. Nessa etapa, foram introduzidas variações na massa, nos momentos de inércia e nos coeficientes de amortecimento da planta, com o objetivo de verificar a robustez da estratégia proposta diante de incertezas paramétricas. Em todos os casos, os resultados do controlador PID convencional foram comparados aos obtidos com o PID-Neural residual, mantendo-se as mesmas condições iniciais, referências de voo e configurações numéricas. Os cenários considerados nessa etapa são apresentados no Quadro 10.

Quadro 10: Cenários avaliados para o controlador PID-Neural residual.

Cenário	Variação paramétrica	Controladores comparados	Objetivo da simulação
C1	Massa reduzida em 20%	PID e PID-Neural residual	Avaliar o desempenho da dinâmica vertical diante da redução da massa.
C2	Massa aumentada em 20%	PID e PID-Neural residual	Avaliar a robustez do controle de altitude diante do aumento da massa.
C3	Momentos de inércia aumentados em 20%	PID e PID-Neural residual	Verificar o comportamento do sistema diante de alterações na dinâmica rotacional.
C4	Coefficientes de amortecimento reduzidos em 30%	PID e PID-Neural residual	Avaliar a resposta do controlador diante da redução do amortecimento.
C5	Massa reduzida, inércias aumentadas e amortecimento reduzido	PID e PID-Neural residual	Avaliar a robustez da estratégia em uma condição combinada de incertezas paramétricas.

Fonte: Elaborado pelo autor (2014).

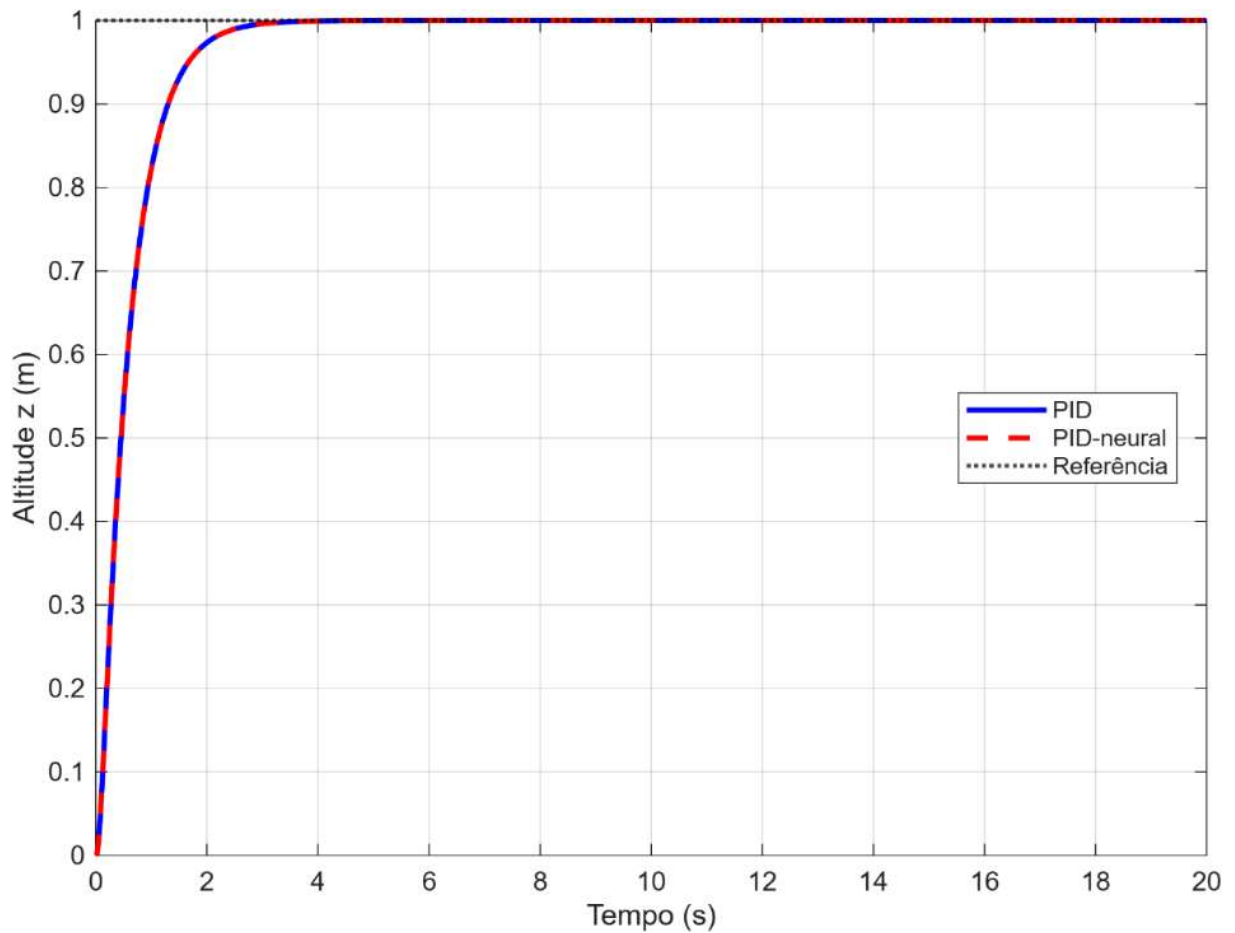
Em todos os cenários, são analisadas as respostas temporais da altitude z e dos ângulos de atitude ϕ , θ e ψ , além dos sinais de controle associados ao empuxo e aos torques aplicados à planta reduzida. Essa análise permite observar se o controlador consegue conduzir o sistema à condição de equilíbrio desejada, reduzir oscilações e manter os estados próximos das referências estabelecidas.

5.5 Resultados da estabilização em voo pairado

Nesta seção são apresentados os resultados obtidos nas simulações de estabilização em voo pairado do quadrotor, considerando as estratégias de controle avaliadas anteriormente. A análise concentra-se nas respostas temporais da altitude e dos ângulos de atitude, bem como nos sinais de controle gerados durante a simulação. O objetivo é verificar se o sistema é capaz de atingir a altitude de referência e reduzir os desvios angulares iniciais, mantendo o veículo próximo da condição de *hover*.

A resposta da altitude permite avaliar a capacidade do controlador de conduzir o quadrotor até a referência estabelecida de 1,0 m. Conforme apresentado na Figura 29, observa-se o comportamento temporal da variável z , permitindo identificar características como tempo de subida, presença de oscilações, aproximação da referência e erro em regime permanente.

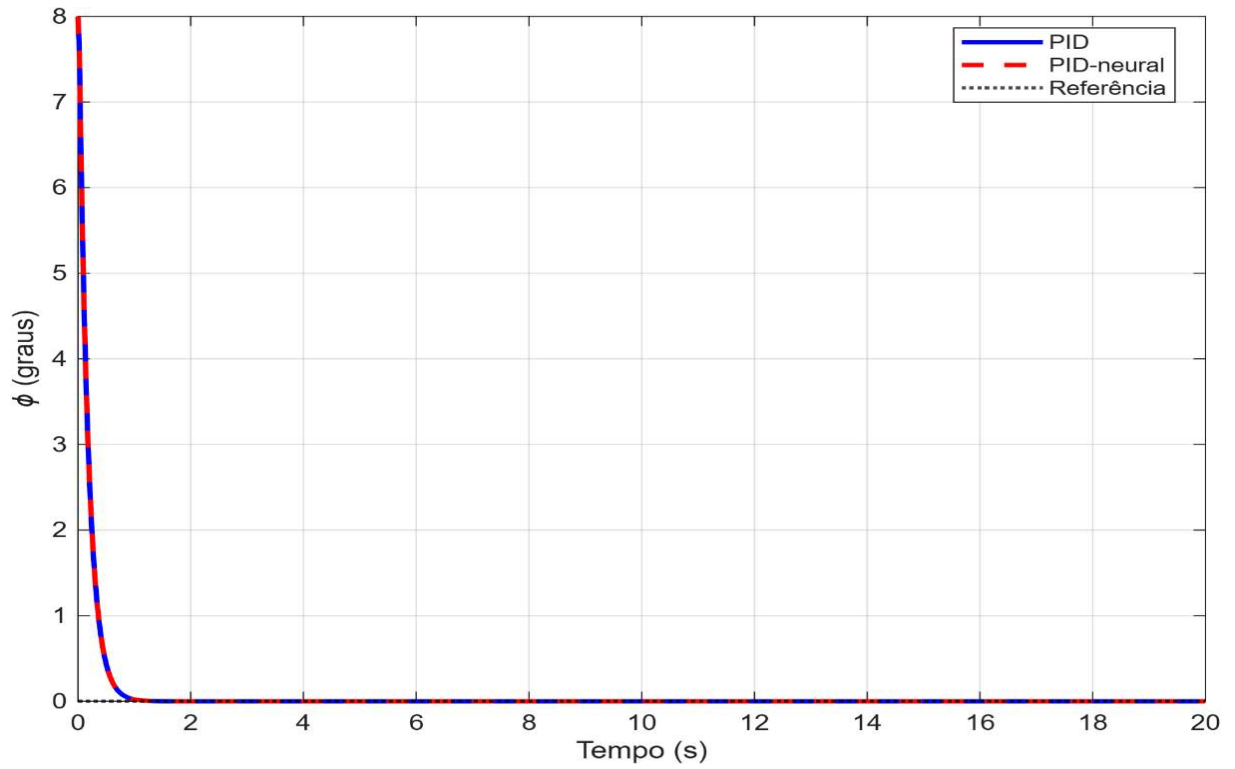
Figura 29 – Resposta da altitude durante a estabilização em voo pairado.



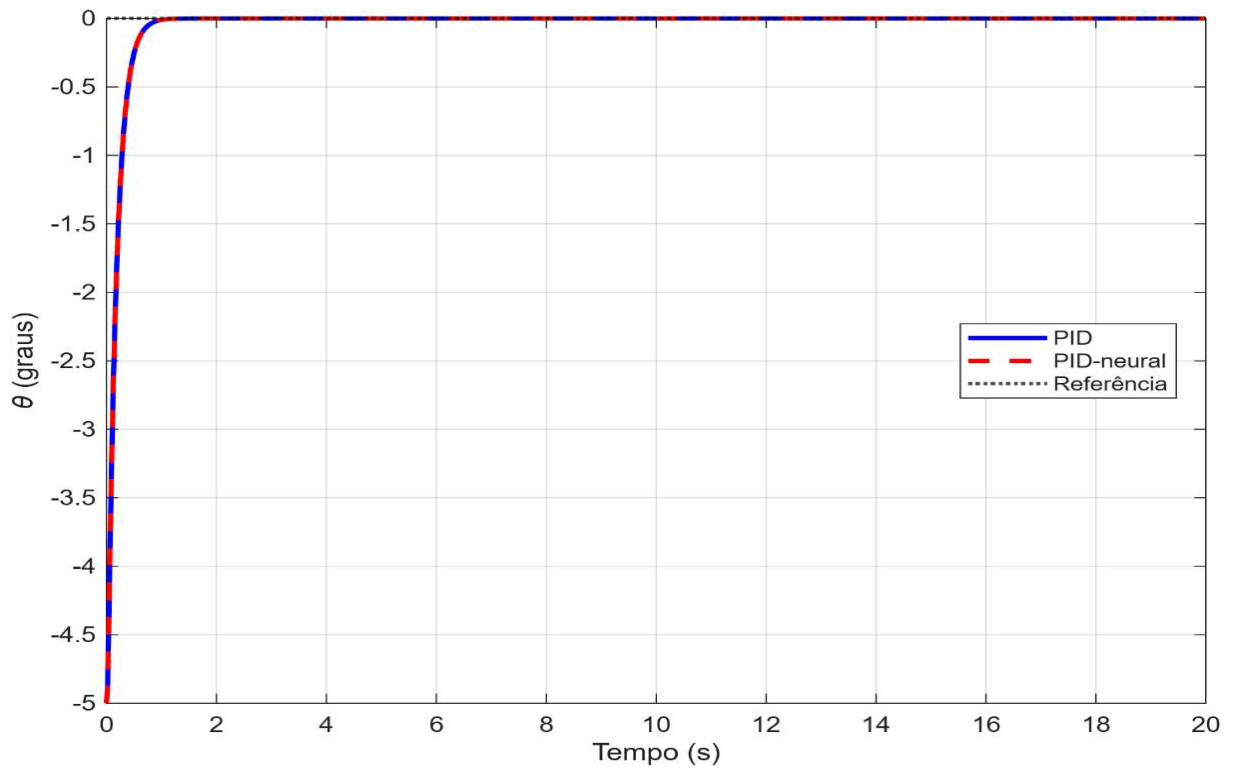
Fonte: Elaborado pelo autor (2014).

Conforme ilustrado na Figura 29, a altitude converge rapidamente para o valor de referência, apresentando comportamento amortecido e pequeno sobressinal. Esse resultado evidencia a estabilidade do controlador durante a fase de estabilização vertical.

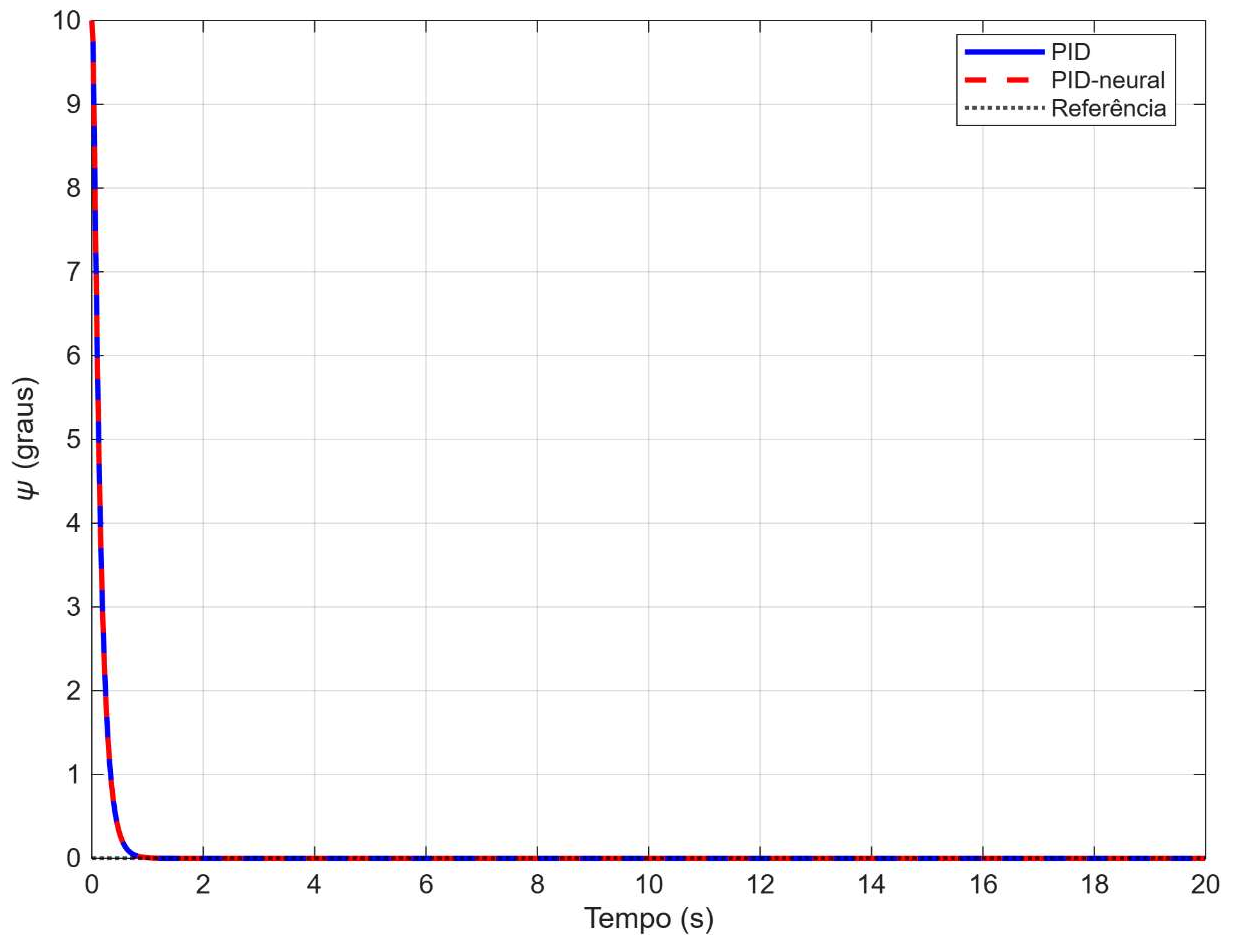
Além da altitude, foram analisadas as respostas dos ângulos de atitude ϕ , θ e ψ , correspondentes aos movimentos de rolagem, arfagem e guinada. Essas variáveis são fundamentais para a estabilização em *hover*, pois desvios angulares podem comprometer o equilíbrio do quadrotor. As Figuras 30, 31 e 32 apresentam, respectivamente, as respostas de rolagem, arfagem e guinada ao longo do tempo.

Figura 30 – Resposta do ângulo de rolagem durante a estabilização.

Fonte: Elaborado pelo autor (2014).

Figura 31 – Resposta do ângulo de arfagem durante a estabilização.

Fonte: Elaborado pelo autor (2014).

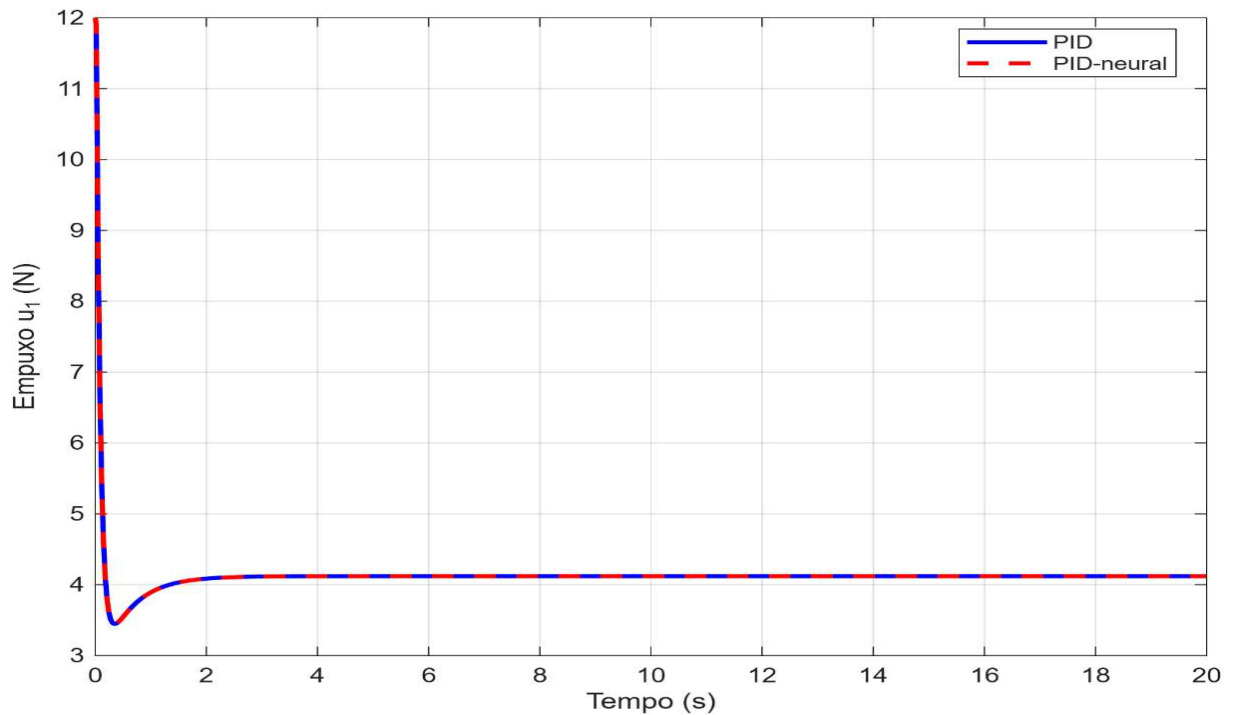
Figura 32 – Resposta do ângulo de guinada durante a estabilização.

Fonte: Elaborado pelo autor (2014).

De modo geral, espera-se que os ângulos de atitude converjam para valores próximos de zero, uma vez que a condição de voo pairado é caracterizada pela manutenção do quadrotor nivelado. A análise dessas respostas permite verificar a eficiência do controlador na correção das perturbações iniciais impostas ao sistema.

Também foram avaliados os sinais de controle produzidos durante a simulação, especialmente o empuxo total e os torques associados aos movimentos de atitude. Essa análise é importante porque permite observar se a estabilização é obtida com comandos suaves e dentro dos limites definidos para a planta.

A Figura 33 apresenta o comportamento do sinal de empuxo ao longo da simulação.

Figura 33 – Sinal de empuxo gerado durante a estabilização em voo pairado.

Fonte: Elaborado pelo autor (2014).

Além da análise visual das respostas temporais, os principais comportamentos esperados para cada variável de interesse são sintetizados na Tabela 10. Essa tabela tem como objetivo organizar os critérios qualitativos utilizados na interpretação dos resultados de estabilização em voo pairado, relacionando cada variável analisada à sua respectiva referência e ao comportamento esperado durante a simulação. Dessa forma, a Tabela 10 auxilia na leitura dos gráficos apresentados anteriormente e serve como base para a discussão comparativa entre as estratégias de controle avaliadas.

Tabela 10: Síntese dos resultados de estabilização em voo pairado.

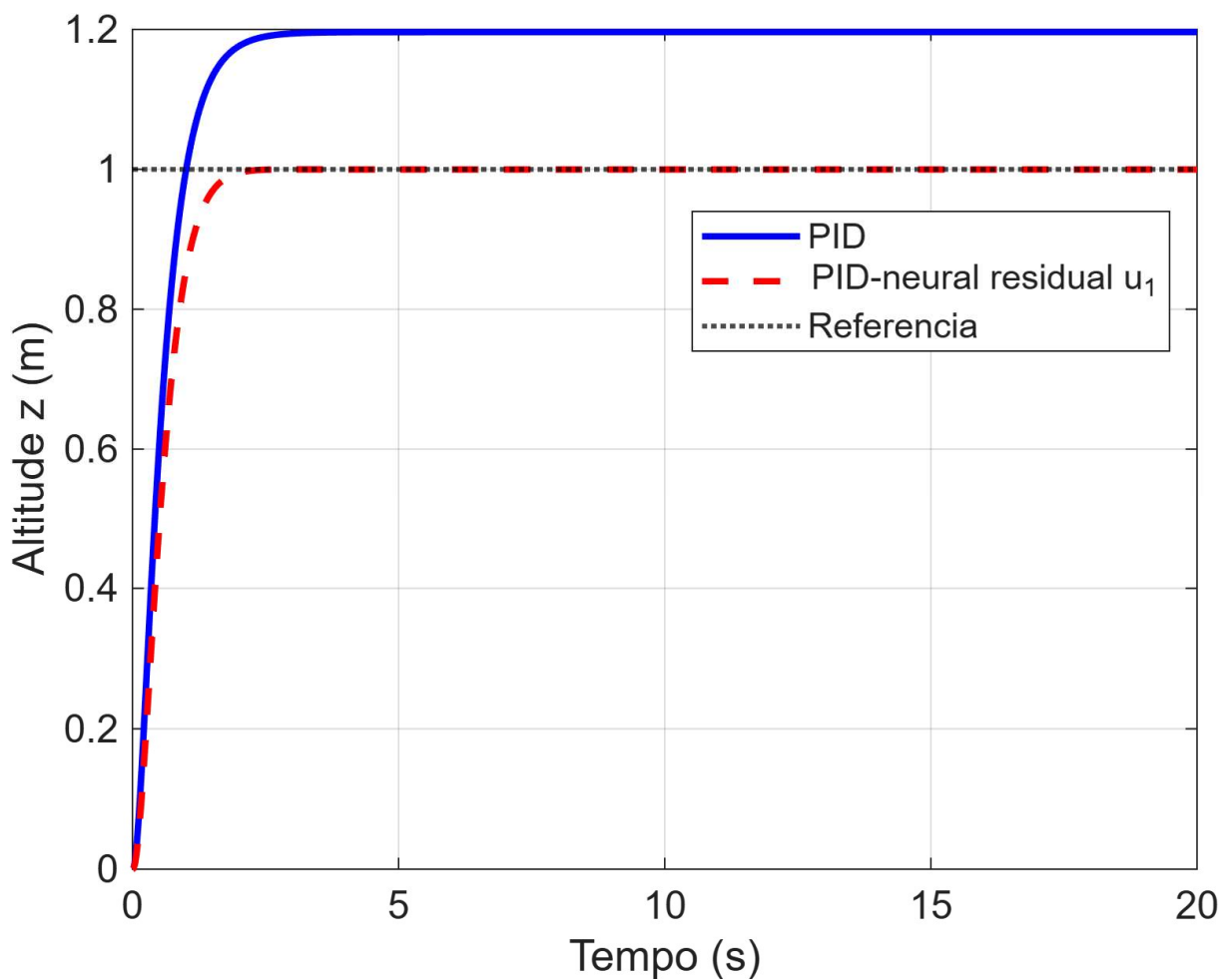
Variável analisada	Referência	Comportamento esperado
Altitude z	1,0 m	Convergência para a altitude de referência
Rolagem ϕ	0°	Redução do desvio inicial e estabilização próxima de zero
Arfagem θ	0°	Redução do desvio inicial e estabilização próxima de zero
Guinada ψ	0°	Redução do desvio inicial e estabilização próxima de zero
Empuxo u_1	-	Comando limitado e coerente com a estabilização

Fonte: Elaborado pelo autor (2014).

Os resultados iniciais da estabilização em voo pairado indicaram que o controlador PID e a configuração original do controlador PID-Neural apresentaram respostas muito próximas para a condição nominal de simulação. Essa semelhança foi observada tanto na resposta de altitude quanto nas respostas angulares, indicando que a contribuição neural inicialmente implementada não modificou de forma significativa a dinâmica de estabilização.

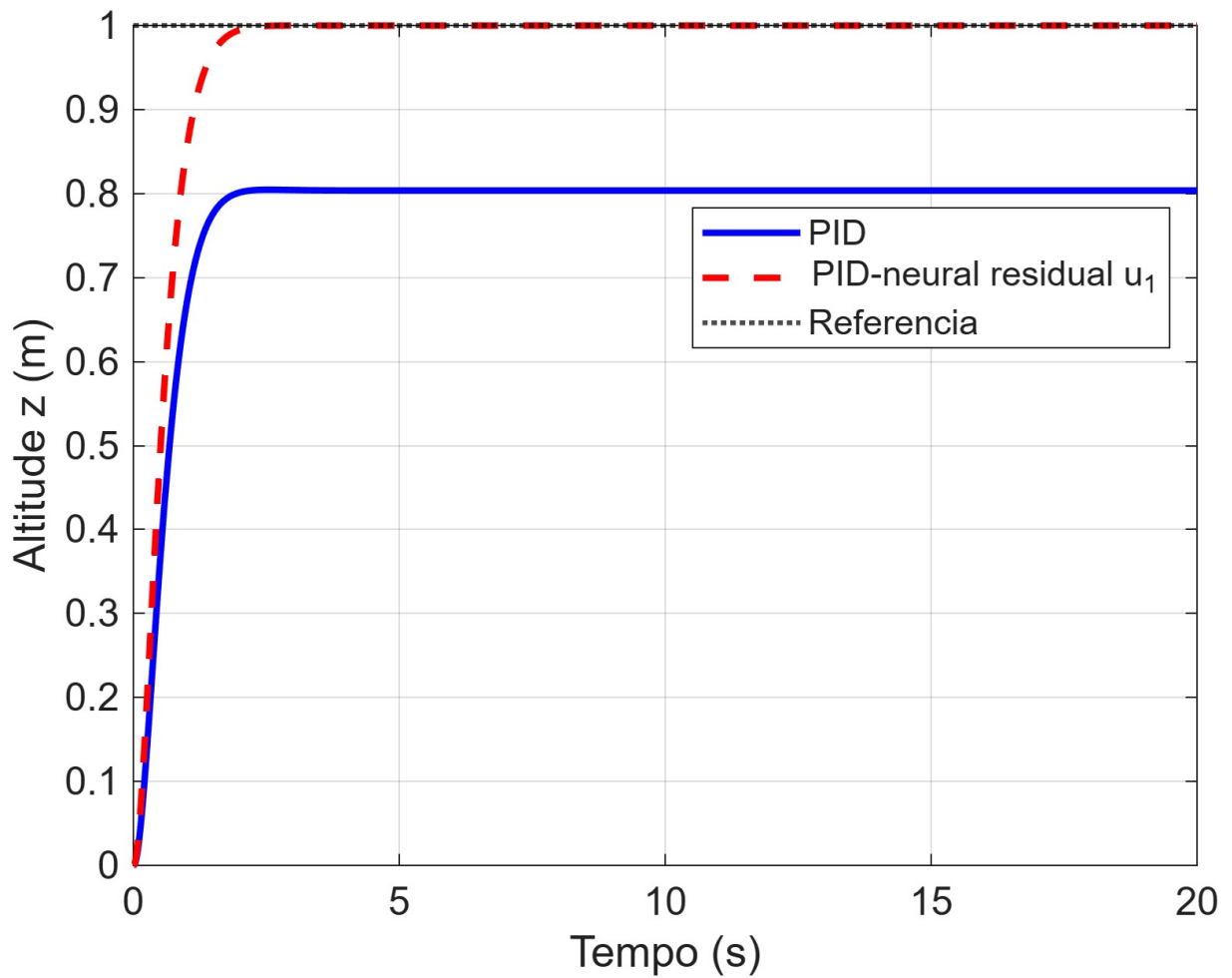
Após restringir a atuação da RNA residual ao canal de empuxo u_1 , foram obtidas novas respostas temporais para os cenários com variações paramétricas da planta. As Figuras 34 e 35 apresentam as respostas de altitude para os cenários de massa reduzida e massa aumentada, respectivamente. Observa-se que o controlador PID-Neural residual aplicado ao empuxo apresentou melhor aproximação da referência de voo pairado em relação ao PID.

Figura 34 – Resposta da altitude no cenário com massa reduzida em 20%.



Fonte: Elaborado pelo autor (2014).

Figura 35 – Resposta da altitude no cenário com massa aumentada em 20%.

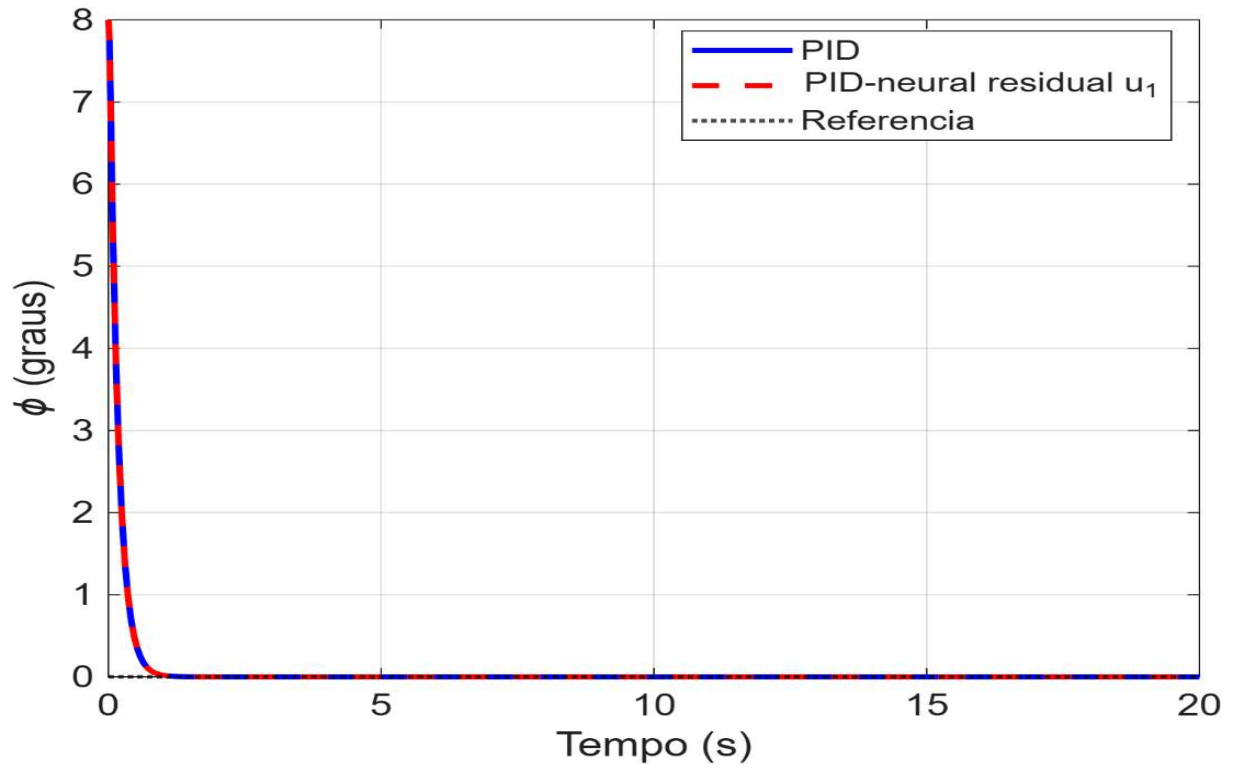


Fonte: Elaborado pelo autor (2014).

A Figura 35 mostra que, mesmo com aumento de 20% da massa do quadrotor, o controlador preserva a estabilidade da altitude, apresentando apenas pequena alteração no tempo de acomodação.

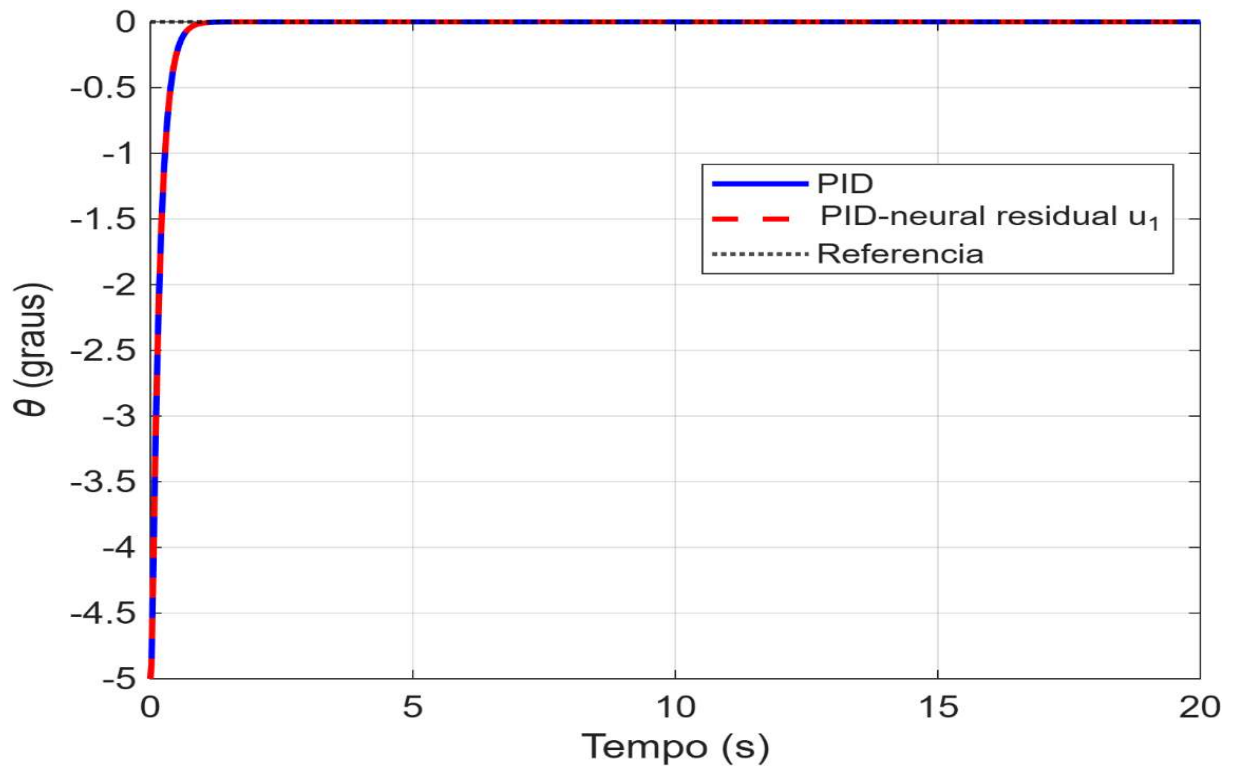
As respostas angulares no cenário combinado são apresentadas nas Figuras 36, 37 e 38. Nota-se que os ângulos de rolagem, arfagem e guinada mantiveram comportamento equivalente ao obtido com o PID, indicando que a restrição da contribuição neural ao canal vertical preservou a estabilização de atitude.

Figura 36 – Resposta do ângulo de rolagem no cenário combinado.



Fonte: Elaborado pelo autor (2014).

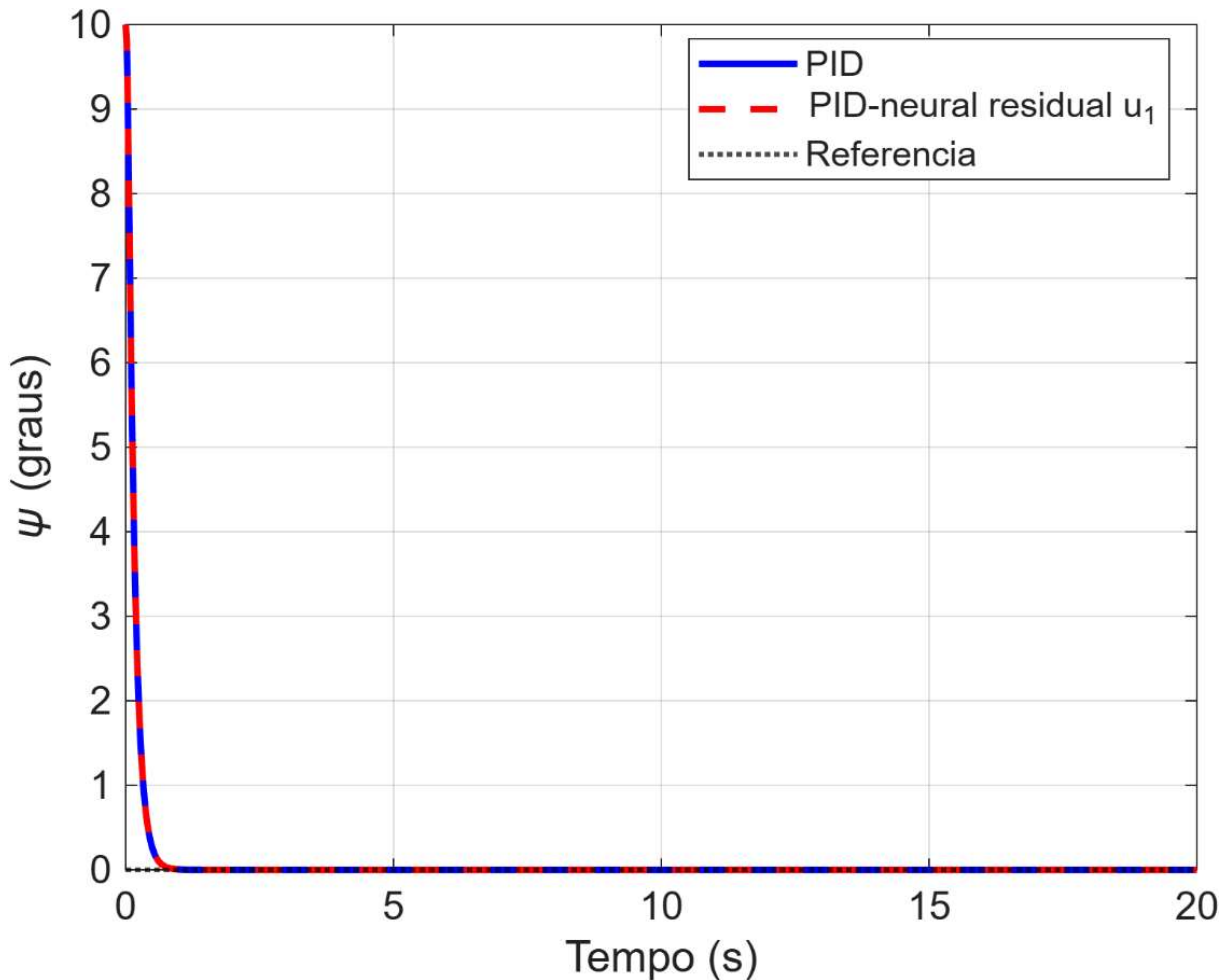
Figura 37 – Resposta do ângulo de arfagem no cenário combinado.



Fonte: Elaborado pelo autor (2014).

A Figura 37 demonstra que o ângulo de arfagem permanece limitado durante todo o experimento, convergindo para a condição de equilíbrio, o que evidencia a robustez do sistema frente ao cenário combinado.

Figura 38 – Resposta do ângulo de guinada no cenário combinado.

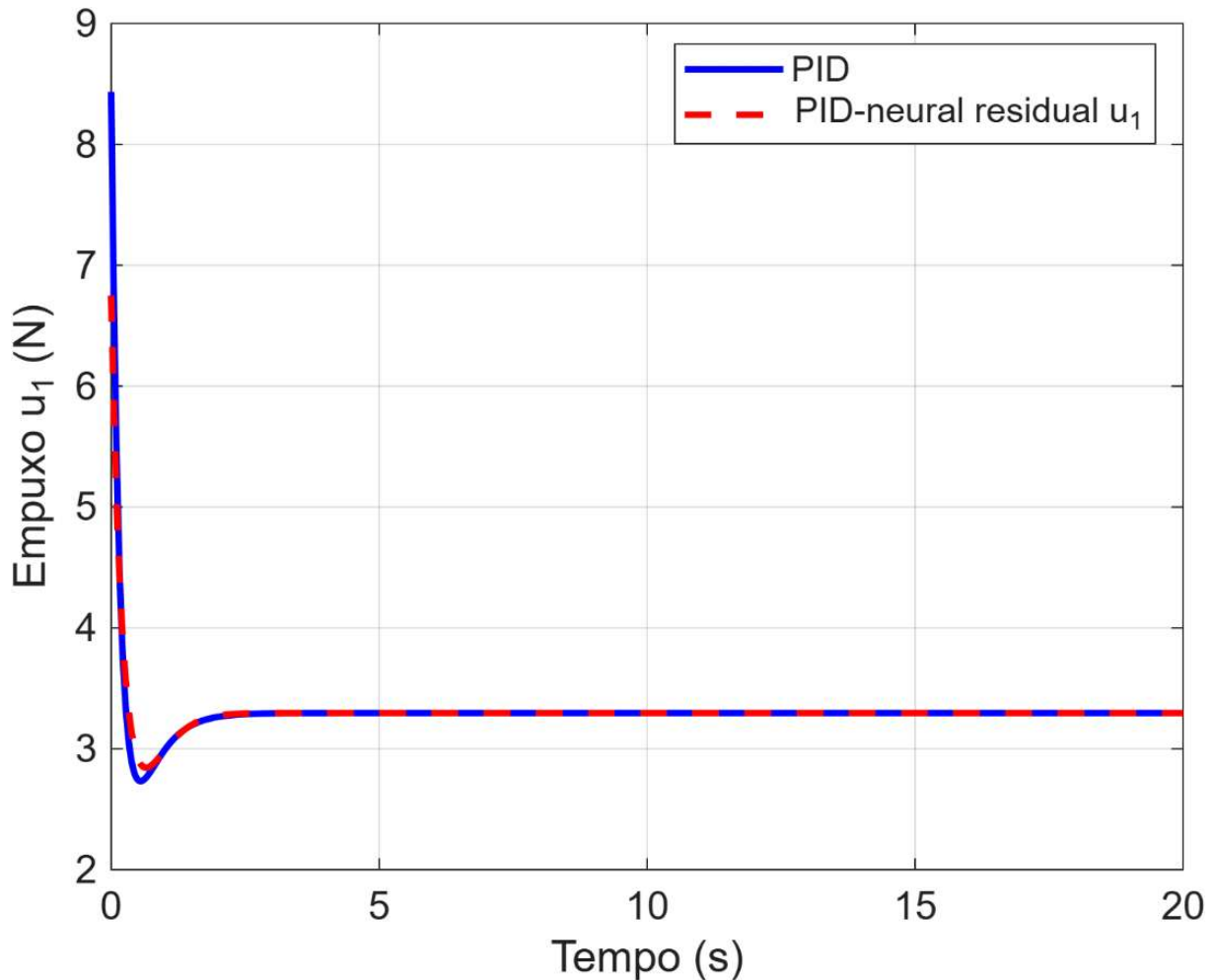


Fonte: Elaborado pelo autor (2014).

Conforme apresentado na Figura 38, a resposta da guinada permanece estável durante toda a simulação, sem oscilações persistentes, indicando que a estratégia de controle preserva o comportamento angular do veículo.

A Figura 39 apresenta o sinal de empuxo no cenário combinado. Observa-se que a melhoria na resposta de altitude foi obtida sem aumento relevante do esforço médio de empuxo, reforçando a adequação da estratégia residual restrita a u_1 .

Figura 39 – Sinal de empuxo no cenário combinado.



Fonte: Elaborado pelo autor (2014).

5.6 Comparação entre PID convencional e PID-Neural

A comparação entre o controlador PID convencional/adaptativo e o controlador PID-Neural foi realizada com o objetivo de avaliar a influência da componente neural no desempenho da estabilização em voo pairado. Para isso, ambos os controladores foram submetidos às mesmas condições iniciais, referências de *hover*, parâmetros físicos e configurações numéricas de simulação, garantindo que as diferenças observadas estejam relacionadas exclusivamente à estratégia de controle empregada.

A comparação inicial entre o PID e o PID-Neural original indicou respostas praticamente equivalentes, sem melhoria quantitativa relevante associada à contribuição neural. Com o retreinamento residual, observou-se melhoria na resposta de altitude; entretanto, quando a correção neural foi aplicada a todos os canais, houve degradação das

respostas angulares. Diante disso, a formulação final restringiu a atuação neural ao canal de empuxo u_1 , mantendo os torques de rolagem, arfagem e guinada definidos pelo PID.

Conforme apresentado nas Figuras 20 a 25, a análise comparativa considera as respostas temporais da altitude, dos ângulos de atitude e do sinal de empuxo. As respostas de altitude permitem verificar a capacidade dos controladores de conduzir o quadrotor à referência de 1,0 m, enquanto as respostas angulares permitem avaliar se a estabilização vertical ocorre sem comprometer a condição de atitude próxima de zero.

Na formulação final do controlador PID-Neural residual, a rede neural atua como uma contribuição complementar apenas no canal vertical. Assim, a correção neural busca reduzir o erro de altitude, enquanto a estabilidade angular permanece assegurada pela ação PID nos canais de torque. Essa escolha evita que a RNA interfira diretamente nas dinâmicas de rolagem, arfagem e guinada, preservando o comportamento angular observado com o controlador de referência.

Além das variáveis de estado, o sinal de empuxo u_1 também foi analisado, pois indica o esforço de controle aplicado para estabilizar a altitude. Essa análise é importante porque a melhoria da resposta vertical deve ser avaliada em conjunto com a solicitação do atuador, evitando que a redução do erro ocorra à custa de comandos excessivamente elevados ou oscilatórios.

A síntese qualitativa dos critérios utilizados na comparação entre os controladores é apresentada na Quadro 11.

Quadro 11: Critérios de comparação entre o PID e o PID-Neural residual aplicado ao canal de empuxo.

Critério analisado	Interpretação esperada
Altitude z	Verificar convergência para 1,0 m, oscilações e erro em regime permanente
Rolagem ϕ	Avaliar a redução do desvio inicial e estabilização próxima de zero
Arfagem θ	Avaliar a redução do desvio inicial e estabilização próxima de zero
Guinada ψ	Avaliar a redução do desvio inicial e estabilização próxima de zero
Empuxo u_1	Verificar suavidade do comando e respeito aos limites da planta
Desempenho geral	Comando limitado e coerente com a estabilização

Fonte: Elaborado pelo autor (2014).

5.7 Análise quantitativa dos resultados

Além da análise gráfica das respostas temporais, foi realizada uma avaliação quantitativa com o objetivo de comparar, de forma mais objetiva, o desempenho dos controladores PID convencional/adaptativo e PID-Neural. Para isso, foram consideradas métricas associadas ao erro de rastreamento, à estabilização dos estados e ao esforço de controle aplicado durante a simulação.

As variáveis analisadas foram a altitude z , os ângulos de atitude ϕ , θ e ψ , e o sinal de empuxo u_1 . No projeto implementado, os erros de controle são armazenados no vetor E , enquanto os sinais de controle são armazenados no vetor U , permitindo calcular indicadores numéricos a partir dos resultados gerados pela função simulador *Hover*.

Para quantificar o erro de estabilização, podem ser utilizados indicadores como o erro absoluto médio, o erro quadrático médio e a integral do erro quadrático. Essas métricas são definidas pelas Equações 105 a 107.

$$MAE = \frac{1}{N} \sum_{k=1}^N |e(k)| \quad (105)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N e(k)^2} \quad (106)$$

$$ISE = \sum_{k=1}^N e(k)^2 T_s \quad (107)$$

em que $e(k)$ representa o erro no instante de amostragem k , N é o número total de amostras e T_s é o tempo de amostragem da simulação.

Além dos erros de rastreamento, também pode ser avaliado o esforço de controle associado ao empuxo total, uma vez que comandos excessivamente elevados ou oscilatórios podem indicar maior solicitação da planta. Uma forma simples de quantificar esse comportamento é por meio da média do valor absoluto do sinal de controle, conforme indicado na Equação 108.

$$\bar{u}_1 = \frac{1}{N} \sum_{k=1}^N |u_1(k)| \quad (108)$$

A Tabela 11 apresenta os indicadores quantitativos obtidos para o controlador PID e para o controlador PID-Neural residual aplicado apenas ao canal de empuxo u_1 . Os valores foram calculados a partir dos dados gerados nas simulações dos cenários com variações paramétricas da planta, permitindo comparar o desempenho das estratégias quanto ao erro de altitude, à estabilização angular e ao esforço de controle.

Tabela 11: Indicadores quantitativos do PID e do PID-Neural residual aplicado ao canal de empuxo.

Cenário	Descrição	RMSE z PID	RMSE z PID- Neural residual u_1	MAE z PID	MAE z PID- Neural residual u_1	Esforço u_1 PID	Esforço u_1 PID- Neural residual	Melhoria RMSE z (%)
C1	Massa reduzida em 20%	0,22580	0,13907	0,20484	0,030129	3,3016	3,3008	38,41
C2	Massa aumentada em 20%	0,24860	0,13913	0,22151	0,030178	4,9498	4,9511	44,03
C3	Inércias aumentadas em 20%	0,14073	0,13908	0,031234	0,030333	4,1259	4,1259	1,17
C4	Amortecimento reduzido em 30%	0,14025	0,13907	0,030860	0,030162	4,1244	4,1244	0,84
C5	Massa menor, inércias maiores e amortecimento menor	0,22574	0,13910	0,20489	0,030200	3,3001	3,2995	38,38

Fonte: Elaborado pelo autor (2014).

Os resultados quantitativos confirmam a melhoria observada nas respostas temporais. A aplicação da RNA residual apenas ao canal u_1 reduziu o RMSE da altitude em aproximadamente 38,41% no cenário de massa reduzida, 44,03% no cenário de massa aumentada e 38,38% no cenário combinado. Ao mesmo tempo, os indicadores de rolagem, arfagem e guinada permaneceram equivalentes aos do PID, demonstrando que a estratégia proposta melhorou a dinâmica vertical sem comprometer a estabilização angular.

5.8 Discussão dos resultados

Os resultados obtidos nas simulações permitem avaliar a evolução da estratégia de controle proposta ao longo das etapas analisadas neste capítulo. Inicialmente, a comparação entre o controlador PID convencional/adaptativo e o controlador PID-Neural originalmente implementado indicou respostas praticamente equivalentes. Esse comportamento mostrou que, na configuração inicial, a contribuição da rede neural não foi suficiente para produzir melhoria significativa na estabilização em voo pairado, uma vez que os indicadores de erro e esforço de controle permaneceram semelhantes aos do PID.

A partir dessa constatação, foi realizada uma etapa complementar com o uso de uma rede neural residual. Nessa abordagem, a RNA passou a atuar como um termo de correção adicional à ação PID, em vez de substituir diretamente a estrutura de controle de referência. A primeira formulação residual, aplicada a todos os canais de controle, apresentou melhoria na resposta de altitude, porém provocou degradação nas respostas angulares. Esse resultado indicou que a atuação neural simultânea sobre os torques de rolagem, arfagem e guinada poderia interferir negativamente na estabilização de atitude.

Diante disso, a estratégia final restringiu a contribuição neural residual ao canal de empuxo u_1 , mantendo os torques u_2 , u_3 e u_4 definidos pelo controlador PID. Essa formulação, apresentada na Seção 6.3, mostrou-se mais adequada ao objetivo do trabalho, pois permitiu melhorar a dinâmica vertical sem comprometer a estabilização angular. As respostas temporais apresentadas nas Figuras 6.B a 6.G evidenciam esse comportamento, principalmente nos cenários com variação de massa e no cenário combinado.

A melhoria mais expressiva ocorreu nos cenários em que a massa do quadrotor foi modificada. Conforme indicado na Tabela 11, a redução do RMSE da altitude foi de aproximadamente 38,41% para massa reduzida em 20%, 44,03% para massa aumentada em 20% e 38,38% no cenário combinado. Esses resultados sugerem que a RNA residual aplicada ao empuxo foi capaz de compensar alterações na relação entre força vertical e aceleração do veículo, contribuindo para uma resposta de altitude mais próxima da referência de voo pairado.

Nos cenários envolvendo aumento dos momentos de inércia e redução do amortecimento, a melhoria quantitativa foi menor. Esse comportamento é coerente com a formulação adotada, uma vez que a ação neural foi restringida ao canal u_1 , associado principalmente à altitude. Como os torques de atitude permaneceram sob responsabilidade do PID, não se esperava uma alteração significativa nos indicadores de rolagem, arfagem e

guinada. Ainda assim, esse resultado é positivo, pois mostra que a melhoria na altitude não ocorreu à custa de degradação angular.

Outro ponto relevante é o esforço de controle. Conforme apresentado na Tabela 11, os valores médios de empuxo permaneceram muito próximos entre o PID e o PID-Neural residual aplicado ao canal u_1 . Isso indica que a melhoria no erro de altitude não exigiu aumento expressivo da solicitação do atuador, diferentemente da formulação residual aplicada a todos os canais, que apresentou comportamento menos adequado nas respostas angulares.

De modo geral, os resultados indicam que a contribuição neural é mais efetiva quando utilizada como mecanismo complementar e direcionado, em vez de aplicada indiscriminadamente a todos os sinais de controle. Para o problema de estabilização em voo pairado considerado neste trabalho, a atuação residual no canal de empuxo mostrou-se suficiente para melhorar a resposta vertical e preservar a estabilidade de atitude. Assim, a estratégia PID-Neural residual restrita a u_1 representa a configuração mais consistente entre as alternativas avaliadas neste capítulo.

Portanto, pode-se afirmar que o controlador PID-Neural proposto apresentou desempenho superior ao PID nos cenários de maior sensibilidade à variação de massa, mantendo comportamento angular equivalente e esforço de controle semelhante. Esses resultados reforçam a viabilidade da utilização de redes neurais como elementos auxiliares de compensação em sistemas de controle de quadrotor, desde que sua atuação seja adequadamente delimitada em função das dinâmicas que se deseja melhorar.

Os resultados demonstram que a principal contribuição desta pesquisa não está na substituição do controlador PID por uma Rede Neural Artificial, mas na utilização da RNA como compensador residual especializado na dinâmica vertical do sistema. Essa estratégia preservou integralmente a estabilidade da dinâmica angular enquanto reduziu significativamente os erros associados ao controle da altitude, particularmente nos cenários em que ocorreram variações da massa do veículo.

As reduções do erro quadrático médio (RMSE) da altitude observadas nos cenários de massa reduzida (38,41%), massa aumentada (44,03%) e cenário combinado (38,38%) confirmam quantitativamente a efetividade da arquitetura proposta. Em contrapartida, os cenários envolvendo apenas alterações dos momentos de inércia (1,17%) e dos coeficientes de amortecimento (0,84%) apresentaram ganhos mais modestos, resultado coerente com a estratégia adotada, uma vez que a Rede Neural Artificial atua exclusivamente sobre o canal de empuxo u_1 .

5.9 Limitações das simulações

As simulações realizadas neste capítulo permitiram avaliar o desempenho do controlador PID-Neural em voo pairado, porém os resultados devem ser interpretados considerando as simplificações adotadas no modelo e no ambiente computacional. A principal limitação está associada ao uso de um modelo reduzido da dinâmica do quadrotor, apresentado na Seção 6.2.1. Embora essa representação seja adequada ao objetivo de analisar a estabilização em *hover*, ela considera apenas os estados de altitude e atitude, não contemplando explicitamente os deslocamentos horizontais e acoplamentos completos presentes em um modelo de seis graus de liberdade.

Outra limitação está relacionada à ausência de efeitos aerodinâmicos mais complexos. A planta reduzida utiliza coeficientes simplificados de amortecimento para representar efeitos dissipativos, conforme os parâmetros apresentados na Tabela 4. Entretanto, fenômenos como arrasto não linear, turbulência, efeito solo, atrasos dos motores, saturações individuais dos rotores, ruídos de sensores e incertezas de medição não foram detalhadamente modelados. Assim, o comportamento obtido nas simulações representa uma aproximação idealizada da condição de voo pairado.

Também deve ser observado que a integração numérica foi realizada com passo fixo, conforme descrito na Seção 6.2.3. Essa escolha facilita a implementação e a comparação entre os controladores, mas não substitui uma análise mais ampla com diferentes métodos de integração, passos de amostragem e condições de discretização. Dessa forma, os resultados obtidos estão diretamente associados às configurações numéricas adotadas no simulador.

No caso da estratégia PID-Neural, uma limitação importante está associada ao treinamento da rede neural. A comparação inicial mostrou que a configuração originalmente implementada não produziu melhoria significativa em relação ao PID. A melhoria foi observada apenas após o retreinamento da RNA como termo residual e, principalmente, após a restrição da atuação neural ao canal de empuxo u_1 . Esse resultado indica que o desempenho da estratégia neural depende fortemente da forma de treinamento, das entradas utilizadas, dos cenários considerados e da maneira como a saída da rede é incorporada à lei de controle.

Além disso, os testes complementares foram realizados a partir de variações paramétricas específicas da planta, envolvendo alterações de massa, inércia e amortecimento, conforme apresentado na Tabela 5. Embora esses cenários permitam avaliar a robustez da estratégia proposta, eles não esgotam todas as condições possíveis de operação do quadrotor. Situações como rajadas de vento, variações bruscas de referência, falhas parciais de atuadores,

ruídos nos sinais de realimentação e trajetórias fora do *hover* não foram consideradas neste capítulo.

Por fim, os resultados foram obtidos exclusivamente em ambiente de simulação. Portanto, ainda não é possível afirmar que o mesmo desempenho será reproduzido em uma plataforma física sem validações adicionais. Ensaio em bancada, testes com modelo de motor mais detalhado, simulações em ambiente de maior fidelidade e experimentos reais seriam necessários para verificar a aplicabilidade prática do controlador PID-Neural residual. Assim, as conclusões apresentadas neste capítulo devem ser entendidas como uma avaliação inicial da estratégia proposta, limitada ao contexto computacional e às hipóteses adotadas para a simulação em voo pairado.

5.10 Considerações finais do capítulo

Neste capítulo, foram apresentados o ambiente de simulação, o modelo reduzido de voo pairado, os parâmetros físicos adotados, as configurações numéricas e os cenários utilizados para avaliar o desempenho do controlador PID-Neural proposto. A análise foi conduzida a partir da comparação com o controlador PID convencional/adaptativo, considerando respostas temporais, métricas quantitativas e esforço de controle.

Inicialmente, verificou-se que a configuração original do PID-Neural apresentou comportamento praticamente equivalente ao PID, sem melhoria significativa nos indicadores analisados. Esse resultado motivou a realização de uma etapa complementar com uma rede neural residual, na qual a RNA passou a atuar como termo de correção adicional à ação PID. A formulação residual aplicada a todos os canais de controle melhorou a resposta de altitude, porém comprometeu as respostas angulares, indicando a necessidade de restringir a atuação neural.

A configuração final adotada restringiu a contribuição da RNA residual ao canal de empuxo u_1 , mantendo os torques de rolagem, arfagem e guinada sob responsabilidade do controlador PID. Conforme mostrado nas Figuras 33 a 38 e na Tabela 9, essa estratégia permitiu reduzir o erro de altitude nos cenários de variação de massa e no cenário combinado, sem prejudicar a estabilização angular do quadrotor.

Os resultados quantitativos confirmaram essa melhoria, com redução do RMSE da altitude de aproximadamente 38,41% no cenário de massa reduzida, 44,03% no cenário de massa aumentada e 38,38% no cenário combinado. Além disso, os indicadores de atitude permaneceram equivalentes aos obtidos com o PID, e o esforço médio de empuxo apresentou

valores muito próximos entre as estratégias avaliadas. Isso demonstra que a melhoria da dinâmica vertical não ocorreu à custa de maior degradação angular ou aumento expressivo da solicitação do atuador.

Dessa forma, conclui-se que a estratégia PID-Neural residual aplicada apenas ao canal de empuxo apresentou o melhor compromisso entre desempenho e estabilidade entre as alternativas avaliadas. Os resultados reforçam que a rede neural pode contribuir para a compensação de variações paramétricas da planta, especialmente relacionadas à massa do quadrotor, desde que sua atuação seja adequadamente delimitada. Assim, o capítulo evidencia a viabilidade da proposta no contexto simulado de estabilização em voo pairado, ao mesmo tempo em que indica a necessidade de validações futuras em modelos mais completos e em ambientes experimentais

Os resultados demonstram que a principal contribuição desta pesquisa não está na substituição do controlador PID por uma Rede Neural Artificial, mas na utilização da RNA como compensador residual especializado na dinâmica vertical do sistema. Essa estratégia preservou integralmente a estabilidade da dinâmica angular enquanto reduziu significativamente os erros associados ao controle da altitude, particularmente nos cenários em que ocorreram variações da massa do veículo.

As reduções do RMSE observadas nos cenários de massa reduzida (38,41%), massa aumentada (44,03%) e cenário combinado (38,38%) confirmam quantitativamente a efetividade da arquitetura proposta.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as conclusões gerais da pesquisa e as possibilidades de continuidade do trabalho desenvolvido. Inicialmente, são retomados o objetivo da dissertação, a estratégia de controle proposta e os principais resultados obtidos nas simulações. Em seguida, são indicadas limitações e sugestões de trabalhos futuros, considerando a ampliação do modelo, o aprimoramento da rede neural e a validação da estratégia em ambientes de maior fidelidade ou em plataformas experimentais.

6.1 Conclusões

Este trabalho apresentou o desenvolvimento e a avaliação de um controlador PID-Neural para estabilização em voo pairado de um quadrotor. A proposta combinou a estrutura

clássica do controlador PID com uma Rede Neural Artificial (RNA), buscando melhorar o desempenho do sistema diante de variações paramétricas da planta.

Inicialmente, foram apresentados os fundamentos teóricos relacionados aos veículos aéreos não tripulados, à modelagem dinâmica de quadrotor, ao controlador PID e às Redes Neurais Artificiais aplicadas ao controle. Com base nesses conceitos, desenvolveu-se o modelo matemático da aeronave e implementou-se um ambiente computacional para simulação e comparação de diferentes estratégias de controle.

A primeira etapa avaliou o controlador PID convencional em relação ao PID-Neural originalmente implementado. Os resultados mostraram desempenho semelhante entre as duas abordagens, evidenciando que a simples inserção de uma RNA na malha de controle não garante melhorias significativas.

Diante desse resultado, foi desenvolvida uma estratégia baseada em compensação neural residual. Inicialmente aplicada aos quatro canais de controle, essa abordagem melhorou a resposta em altitude, porém comprometeu a estabilização angular. Como consequência, a formulação final restringiu a atuação da RNA ao canal de empuxo u_1 , mantendo os torques de rolagem, arfagem e guinada sob responsabilidade do controlador PID.

A estratégia residual apresentou desempenho superior ao PID nos cenários com maior sensibilidade à variação de massa, reduzindo o RMSE da altitude em 38,41% para massa reduzida em 20%, 44,03% para massa aumentada em 20% e 38,38% no cenário combinado. Além disso, os indicadores de atitude permaneceram equivalentes aos do PID, demonstrando que a melhoria da dinâmica vertical não comprometeu a estabilização angular.

Conclui-se, portanto, que o objetivo da pesquisa foi alcançado. A utilização da RNA como compensador residual restrito ao canal de empuxo mostrou-se mais adequada do que sua aplicação simultânea a todos os canais de controle, preservando a robustez do PID e explorando a capacidade adaptativa da rede neural. Assim, a principal contribuição desta dissertação consiste na demonstração de que uma arquitetura híbrida baseada em compensação residual pode aumentar o desempenho do controle de altitude sem degradar a estabilidade da atitude.

Como limitações, destacam-se a utilização de um modelo reduzido para voo pairado e a ausência de efeitos como ruídos de sensores, rajadas de vento, atrasos dos atuadores, saturação individual dos motores e validação experimental. Dessa forma, os resultados devem ser interpretados como uma avaliação da estratégia proposta em ambiente computacional.

6.2 Trabalhos futuros

Como trabalhos futuros, recomenda-se a ampliação do modelo de simulação para uma dinâmica completa de seis graus de liberdade, incluindo os deslocamentos translacionais nos eixos horizontais e os acoplamentos dinâmicos entre posição e atitude. Também é recomendável inserir modelos mais realistas dos motores, sensores e atuadores, contemplando atrasos, ruídos, saturações individuais e limitações físicas do sistema.

Outra possibilidade de continuidade consiste em aprimorar o treinamento da rede neural residual, utilizando bases de dados mais abrangentes, com diferentes condições iniciais, perturbações externas, variações de referência e cenários de incerteza paramétrica. Além disso, podem ser investigadas outras arquiteturas de redes neurais, estratégias de aprendizado online, redes recorrentes ou métodos híbridos combinando controle clássico, controle adaptativo e inteligência computacional.

Também se recomenda avaliar a estratégia proposta em cenários mais exigentes, como mudanças de referência de altitude, presença de rajadas de vento, falhas parciais de atuadores, variações de carga útil e trajetórias fora da condição estrita de *hover*. Esses testes permitiriam verificar se a atuação residual restrita ao canal de empuxo permanece adequada ou se outras formas de integração entre PID e RNA seriam necessárias.

Por fim, recomenda-se a validação da estratégia em ambientes de simulação de maior fidelidade e, posteriormente, em uma plataforma experimental. Essa etapa permitiria verificar a aplicabilidade prática do controlador PID-Neural residual e avaliar seu desempenho diante de condições reais de operação. Com isso, seria possível avançar da validação computacional para uma aplicação mais próxima de sistemas embarcados em quadrotor.

Os scripts principais utilizados na implementação e reprodução das simulações são apresentados nos Apêndices A, B e C. Resultados complementares, incluindo testes intermediários realizados durante a definição da estratégia PID-Neural residual, são apresentados no Apêndice D.

REFERÊNCIAS BIBLIOGRÁFICAS

ÅSTRÖM, Karl Johan; HÄGGLUND, Tore. **Advanced PID Control**. Research Triangle Park: ISA – The Instrumentation, Systems, and Automation Society, 2006. ISBN 1-55617-942-1.

BOUABDALLAH, Samir; BECKER, Marcelo; DE PERROT, Vincent; SIEGWART, Roland. **Toward obstacle avoidance on quadrotors**. In: **INTERNATIONAL SYMPOSIUM ON DYNAMIC PROBLEMS OF MECHANICS (DINAME)**, 12., 2007, Ilhabela, SP. Proceedings... Ilhabela: ABCM, 2007.

BRANDÃO, Alexandre Santos *et al.* **Modelagem e controle não linear subatuado de um quadrotor: parte 1**. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 19., 2012, Campina Grande. Anais... Campina Grande: Sociedade Brasileira de Automática, 2012. p. 449 – 454.

BRANDÃO, Alexandre Santos *et al.* **Modelagem e controle não linear subatuado de um quadrotor: parte 2**. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 19., 2012, Campina Grande. Anais... Campina Grande: Sociedade Brasileira de Automática, 2012. p. 3102 – 3109.

CASTILLO, Pedro; GARCÍA, Pedro; LOZANO, Rogelio; ALBERTOS, Pedro. **Modelado y estabilización de un helicóptero con cuatro rotores**. Revista Iberoamericana de Automática e Informática Industrial, Valencia, v. 4, n. 1, p. 41 – 57, jan. 2007.

CUI, Xianzhong; SHIN, Kang G. **Direct Control and Coordination Using Neural Networks**. IEEE Transactions on Systems, Man, and Cybernetics, v. 23, n. 3, p. 686 – 697, Mai/Jun. 1993.

FERRARI, Allan Christian Krainski. **Controlador PID sintonizado por redes neurais artificiais**. 2011. Monografia (Graduação em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, Curitiba, 2011.

HAYKIN, Simon. **Neural networks: a comprehensive foundation**. 2. ed. Upper Saddle River: Prentice Hall, 1999.

HAYKIN, Simon. **Redes neurais: princípios e prática**, Tradução de Paulo Martins Engel. – 2.ed., Porto Alegre: Bookman, 2008.

HAYKIN, Simon. **Neural networks and learning machines**. 3. ed. New Delhi: Pearson Education India, 2010.

KOHONEN, Teuvo. **The self-organizing map**. Proceedings of the IEEE, New York, v. 78, n. 9, p. 1464 – 1480, 1990.

LEISHMAN, J. Gordon. **The breguet-richet quad-rotor helicopter of 1907**. Vertiflite, v. 47, n. 3, p. 58 – 60, 2002.

LOPES, Leonardo L.; BRANDÃO, Alexandre S.; SARCINELLI FILHO, Mario; CARELLI, Ricardo. **Modelagem e validação de um quadrimotor ArDrone Parrot**. In: **CONGRESSO BRASILEIRO DE AUTOMÁTICA**, 20., 2014, Belo Horizonte, MG. Anais... Belo Horizonte: Sociedade Brasileira de Automática, 2014. p. 2689 – 2696.

MEHROTRA, Kishan; MOHAN, Chilukuri K.; RANKA, Sanjay. **Elements of artificial neural networks**. Cambridge, MA: MIT Press, 1996.

MUÑOZ, Miguel Enrique Parra. **Modelagem matemática e controle de um quadrimotor**. 2012. 104 f. Dissertação (Mestrado em Sistemas Mecatrônicos) – Departamento de Engenharia Mecânica, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 2012.

PARROT. **AR.Drone 2.0: user guide**. [S.l.]: Parrot, [2012]. 38 p. Manual do usuário. Disponível em: <https://ardrone2.parrot.com/support>. Acesso em: 17 jun. 2014.

RAFFO, Guilherme Vianna. **Modelado y control de un helicóptero quad-rotor**. 2007. Tese (Doutorado em Ingeniería de Sistemas y Automática) – Universidad de Sevilla, Sevilla, 2007.

RALL, W. **Some historical notes**. In: SCHWARTZ, Eric L. (ed.). **Computational neuroscience**. Cambridge: MIT Press, 1990. p. 3 – 8.

ROSENBLATT, Frank. **The perceptron: a probabilistic model for information storage and organization in the brain**. *Psychological Review*, Washington, DC, v. 65, n. 6, p. 386 – 408, nov. 1958.

RUMELHART, David E.; MCCLELLAND, James L.; PDP RESEARCH GROUP. **Parallel distributed processing: explorations in the microstructure of cognition. v. 1: Foundations**. Cambridge, MA: MIT Press, 1986. (Computational Models of Cognition and Perception).

SOUSA, José Duarte Alves de. **Simulação e desenvolvimento de um veículo aéreo autônomo de quatro rotores**. 2011. 178 f. Dissertação (Mestrado Integrado em Engenharia Electrotécnica e de Computadores – Major Automação) – Faculdade de Engenharia, Universidade do Porto, Porto, 2011.

UNITED STATES. Joint Chiefs of Staff. Office of the Chairman. **DoD Dictionary of Military and Associated Terms**. Washington, DC: The Joint Staff, jun. 2025. 300 p.

WANNBERG, Mathias. **The quadrotor platform from a military point of view**. Stockholm: KTH Royal Institute of Technology; Etteplan, 2012. 62 p. Relatório técnico. Disponível em: <https://www.diva-portal.org/smash/get/diva2:550541/FULLTEXT01.pdf>. Acesso em: 17 jun. 2014.

WIDROW, Bernard; HOFF, Marcian E. **Adaptive switching circuits**. In: **IRE WESCON CONVENTION RECORD**, 1960. Proceedings... Part 4. New York: Institute of Radio Engineers, 1960. p. 96 – 104. Disponível em: <https://www-isl.stanford.edu/~widrow/papers/c1960adaptiveswitching.pdf>. Acesso em: 17 jun. 2014.

APÊNDICE

APÊNDICE A — SCRIPTS DA SIMULAÇÃO E DOS CONTROLADORES

Este apêndice apresenta os principais scripts utilizados na implementação do ambiente de simulação e dos controladores avaliados nesta dissertação. Os arquivos foram desenvolvidos em *MATLAB/Simulink* e organizados de modo a permitir a definição dos parâmetros físicos do quadrotor, das referências de voo pairado, das condições iniciais, da planta reduzida em *hover* e das estratégias de controle PID e PID-Neural. O objetivo deste apêndice é complementar a descrição metodológica apresentada no Capítulo 6, fornecendo os elementos computacionais necessários para a reprodução das simulações realizadas.

A.1 Script de parâmetros físicos do quadrotor

Este item apresenta o código da função responsável pela definição dos parâmetros físicos do quadrotor, incluindo massa, aceleração da gravidade, momentos de inércia, limites de empuxo e torque, além dos coeficientes de amortecimento utilizados na planta reduzida em voo pairado. O código correspondente é apresentado a seguir.

[illegible]

```

D.maxThrust = 12;
D.minThrust = 0;
D.maxTorque = 2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Amortecimento (modelo simplificado)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D.kz = 0.25;
D.kphi = 0.08;
D.ktheta = 0.08;
D.kpsi = 0.05;
End

```

A.2 Script de referências de voo pairado

Este item apresenta o código responsável pela definição das referências de voo pairado utilizadas nas simulações. No projeto, a referência principal corresponde à altitude desejada de 1,0 m, enquanto os ângulos de rolagem, arfagem e guinada são definidos como nulos, caracterizando a condição desejada de *hover*. O código correspondente é apresentado a seguir.

```

function R = referencias()
R.z = 1.0;
R.phi = 0;
R.theta = 0;
R.psi = 0;
end

```

A.3 Script da planta reduzida em *hover*

Este item apresenta o código da função responsável pela representação da planta reduzida do quadrotor em voo pairado. A função calcula a evolução dos estados de altitude e atitude a partir dos sinais de controle aplicados, considerando a dinâmica vertical, os movimentos de rolagem, arfagem e guinada, os momentos de inércia e os coeficientes de amortecimento adotados na simulação. O código correspondente é apresentado a seguir.

```
function dX = plantaHover(~,X,U,D)
```

[illegible]

ESTADOS

[illegible]
$$z = X(1);$$
$$dz = X(2);$$
$$\phi = X(3);$$

p = X(4);

```
theta = X(5);
```

$$q = X(6);$$

```
psi = X(7);
```

$$\mathbf{r} = \mathbf{X}(8);$$

%%%%%%%%%%

ENTRADAS

[illegible]
$$\mathfrak{u}1 = \mathrm{U}(1);$$
$$u_2 = U(2);$$
$$u_3 = U(3);$$
$$u_4 = U(4);$$
[illegible]

DINÂMICA

%%%%%%%%%%

$$ddz = (u1/D.m)*\cos(phi)*\cos(theta) \dots$$

- D.g ...

- D.kz*dz;

$$dp = u^2/D.I_{xx} \dots$$

- D.kphi*p;

$$dq = u^3/D.I_{yy} \dots$$

- D.ktheta*q;

$$dr = u_4/D.l_{zz} \dots$$

- D.kpsi*r;

[illegible]

CINEMÁTICA

%%%%%%%%%%

```

dphi = p;
dtheta = q;
dpsi = r;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
VETOR DERIVADA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dX = [...
dz
ddz
dphi
dp
dtheta
dq
dpsi
dr
];
end

```

A.4 Script principal de simulação

Este item apresenta o código principal de simulação em malha fechada. Esse script realiza a integração numérica do modelo reduzido, aplica o controlador selecionado, armazena os estados, os erros e os sinais de controle, e retorna os resultados utilizados nas análises apresentadas no Capítulo 6. O código correspondente é apresentado a seguir.

```

function Resultado = simuladorHover(Controlador,X0,R)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ARSim Hover
Simulador Discreto
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin < 1 || isempty(Controlador)
    Controlador = @controladorPIDAdaptativo;
end
if nargin < 2 || isempty(X0)

```

```

X0 = [...
    0 ...
    0 ...
    deg2rad(8) ...
    0 ...
    deg2rad(-5) ...
    0 ...
    deg2rad(10) ...
    0];
end
if nargin < 3 || isempty(R)
    R = referencias();
end
D = parametrosDrone();
nomeControlador = func2str(Controlador);
if isvarname(nomeControlador)
    clear(nomeControlador)
end
if strcmp(nomeControlador,'controlador_rna') || strcmp(nomeControlador,'controladorNeural')
    clear controladorPIDAdaptativo
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Tempo de simulacao
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ts = 0.01;
Tf = 20;
N = round(Tf/Ts);
D.Ts = Ts;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Vetores
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = zeros(N,1);
X = zeros(N,8);
U = zeros(N,4);

```

```

E = zeros(N,4);
X(1,:) = X0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Loop Principal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k = 1:N-1
    t(k+1) = t(k)+Ts;
    [uk,ek] = Controlador(X(k,:),D,R);
    dx = plantaHover(t(k),X(k,:)',uk,D);
    X(k+1,:) = X(k,:) + Ts*dx';
    U(k,:) = uk;
    E(k,:) = ek;
end
U(N,:) = U(N-1,:);
E(N,:) = E(N-1,:);
Resultado.t = t;
Resultado.X = X;
Resultado.U = U;
Resultado.E = E;
Resultado.R = R;
Resultado.X0 = X0;
Resultado.Controlador = func2str(Controlador);
end

```

A.5 Scripts dos controladores

Este item apresenta os códigos dos controladores avaliados na dissertação. São incluídos o controlador PID convencional/adaptativo, o controlador PID-Neural originalmente implementado e a versão final do controlador PID-Neural residual com atuação restrita ao canal de empuxo u_1 . Os códigos correspondentes são apresentados a seguir.

A.5.1 Controlador PID Convencional/Adaptativo

```
function [U,E] = controladorPIDAdaptativo(X,D,R)
```

%%%

MEMÓRIA DO CONTROLADOR

%%%

persistent ezInt

persistent ephiInt

persistent ethetaInt

persistent epsiInt

persistent ezAnt

persistent ephiAnt

persistent ethetaAnt

persistent epsiAnt

if isempty(ezInt)

 ezInt = 0;

 ephiInt = 0;

 ethetaInt = 0;

 epsiInt = 0;

 ezAnt = 0;

 ephiAnt = 0;

 ethetaAnt = 0;

 epsiAnt = 0;

end

%%%

ESTADOS

%%%

z = X(1);

dz = X(2);

phi = X(3);

p = X(4);

theta = X(5);

q = X(6);

psi = X(7);

r = X(8);

%%%

ERROS

%%%

ez = R.z - z;

ephi = R.phi - phi;

etheta = R.theta - theta;

epsi = R.psi - psi;

Ts = D.Ts;

ezInt = ezInt + ez*Ts;

ephiInt = ephiInt + ephi*Ts;

ethetaInt = ethetaInt + etheta*Ts;

epsiInt = epsiInt + epsi*Ts;

limiteIntegral = 2;

ezInt = max(min(ezInt,limiteIntegral),-limiteIntegral);

ephiInt = max(min(ephiInt,limiteIntegral),-limiteIntegral);

ethetaInt = max(min(ethetaInt,limiteIntegral),-limiteIntegral);

epsiInt = max(min(epsiInt,limiteIntegral),-limiteIntegral);

%%%

GANHOS (provisórios)

%%%

Kpz = 10;

Kiz = 0;

Kdz = 6;

Kpphi = 0.8;

Kiphi = 0;

Kdphi = 0.15;

Kptheta = 0.8;

Kitheta = 0;

Kdtheta = 0.15;

Kppsi = 1.2;

Kipsi = 0;

Kdpsi = 0.20;

%%%

CONTROLE PID

%%%

u1 = D.m*D.g ...

```

+ Kpz*ez ...
+ Kiz*ezInt ...
- Kdz*dz;
u2 = Kpphi*ephi ...
+ Kiphi*ephiInt ...
- Kdphi*p;
u3 = Kptheta*etheta ...
+ Kitheta*ethetaInt ...
- Kdtheta*q;
u4 = Kppsi*epsi ...
+ Kipsi*epsiInt ...
- Kdpsi*r;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SATURAÇÃO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u1 = min(max(u1,D.minThrust),D.maxThrust);
u2 = min(max(u2,-D.maxTorque),D.maxTorque);
u3 = min(max(u3,-D.maxTorque),D.maxTorque);
u4 = min(max(u4,-D.maxTorque),D.maxTorque);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U = [u1 u2 u3 u4];
E = [...
    ez
    ephi
    etheta
    epsi];
ezAnt = ez;
ephiAnt = ephi;
ethetaAnt = etheta;
epsiAnt = epsi;
end

```

A.5.2 Controlador PID-Neural Original

```

function [U,E] = controlador_rna(X,D,R)
persistent rede
persistent ezInt
persistent ephiInt
persistent ethetaInt
persistent epsiInt
if isempty(rede)
    dados = load('rede_drone.mat','rede');
    rede = dados.rede;
end
if isempty(ezInt)
    ezInt = 0;
    ephiInt = 0;
    ethetaInt = 0;
    epsiInt = 0;
end
z = X(1);
dz = X(2);
phi = X(3);
p = X(4);
theta = X(5);
q = X(6);
psi = X(7);
r = X(8);
phiRNA = max(min(phi,deg2rad(30)),-deg2rad(30));
thetaRNA = max(min(theta,deg2rad(30)),-deg2rad(30));
ez = R.z - z;
ephi = R.phi - phi;
etheta = R.theta - theta;
epsi = R.psi - psi;
ezInt = ezInt + ez*D.Ts;
ephiInt = ephiInt + ephi*D.Ts;
ethetaInt = ethetaInt + etheta*D.Ts;
epsiInt = epsiInt + epsi*D.Ts;

```

```

limiteIntegral = 2;
ezInt = max(min(ezInt,limiteIntegral),-limiteIntegral);
ephiInt = max(min(ephiInt,limiteIntegral),-limiteIntegral);
ethetaInt = max(min(ethetaInt,limiteIntegral),-limiteIntegral);
epsiInt = max(min(epsiInt,limiteIntegral),-limiteIntegral);
entrada = [...
    z
    phiRNA
    thetaRNA
    psi
    ez
    ephi
    etheta
    epsi
    ezInt
    ephiInt
    ethetaInt
    epsiInt
    dz
    p
    q
    r];
URNA = rede(entrada)';
[UPID,~] = controladorPIDAdaptativo(X,D,R);
alpha = 0.0;
U = (1-alpha)*UPID + alpha*URNA;
U(1) = min(max(U(1),D.minThrust),D.maxThrust);
U(2) = min(max(U(2),-D.maxTorque),D.maxTorque);
U(3) = min(max(U(3),-D.maxTorque),D.maxTorque);
U(4) = min(max(U(4),-D.maxTorque),D.maxTorque);
E = [ez ephi etheta epsi];
end

```

A.5.3 Controlador PID-Neural Residual Aplicado Ao Canal De Empuxo

```

%% gerar_figuras_residual_apenas_u1.m
% Gera figuras comparativas entre PID e PID-Neural residual com correção
% neural aplicada apenas ao canal de empuxo u1.
% Lei de controle avaliada:
%  $u1 = u1\_PID + \alpha * u1\_RNA\_res$ 
%  $u2 = u2\_PID$ 
%  $u3 = u3\_PID$ 
%  $u4 = u4\_PID$ 
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
% /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:
% gerar_figuras_residual_apenas_u1
clear
clc
close all
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)
end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual', 'figuras_residual_apenas_u1');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
redeFile = fullfile(pwd, 'resultados_retreino_rna_residual', 'rede_residual_pid_neural.mat');
if exist(redeFile, 'file') ~= 2
    error(['Arquivo da rede residual nao encontrado: ', redeFile, newline, ...
        'Execute primeiro o script testar_retreino_rna_residual.m'])
end
S = load(redeFile);
redeResidual = S.redeResidual;
if isfield(S, 'alpha')
    alpha = S.alpha;

```

```

else
    alpha = 1.0;
end
Dnom = parametrosDrone();
R = referencias();
Ts = 0.01;
Tf = 20;
Dnom.Ts = Ts;
cenarios = struct([]);
cenarios(1).id = 'C01';
cenarios(1).nome = 'massa_menor_20';
cenarios(1).descricao = 'Massa reduzida em 20%';
cenarios(1).D = alterar_planta(Dnom, 0.80, 1.00, 1.00);
cenarios(2).id = 'C02';
cenarios(2).nome = 'massa_maior_20';
cenarios(2).descricao = 'Massa aumentada em 20%';
cenarios(2).D = alterar_planta(Dnom, 1.20, 1.00, 1.00);
cenarios(3).id = 'C03';
cenarios(3).nome = 'inercias_maiores';
cenarios(3).descricao = 'Inercias aumentadas em 20%';
cenarios(3).D = alterar_planta(Dnom, 1.00, 1.20, 1.00);
cenarios(4).id = 'C04';
cenarios(4).nome = 'amortecimento_menor';
cenarios(4).descricao = 'Amortecimento reduzido em 30%';
cenarios(4).D = alterar_planta(Dnom, 1.00, 1.00, 0.70);
cenarios(5).id = 'C05';
cenarios(5).nome = 'combinado';
cenarios(5).descricao = 'Massa menor, inercias maiores e amortecimento menor';
cenarios(5).D = alterar_planta(Dnom, 0.80, 1.20, 0.70);
linhasFig = {};
linhasMet = {};
for c = 1:numel(cenarios)
    Dplant = cenarios(c).D;
    ResPID = simular_local(Dnom, Dplant, R, Ts, Tf, [], alpha, false);

```

```

ResRNA = simular_local(Dnom, Dplant, R, Ts, Tf, redeResidual, alpha, true);
metPID = metricas_locais(ResPID);
metRNA = metricas_locais(ResRNA);
melhoriaRMSE = 100*(metPID.RMSE_z - metRNA.RMSE_z)/metPID.RMSE_z;
if metRNA.RMSE_z < metPID.RMSE_z
    vencedorAltitude = 'PID-Neural residual u1';
else
    vencedorAltitude = 'PID';
end
linhasMet(end+1,:) = { ...
    cenarios(c).id, cenarios(c).nome, cenarios(c).descricao, ...
    metPID.RMSE_z, metRNA.RMSE_z, ...
    metPID.MAE_z, metRNA.MAE_z, ...
    metPID.RMSE_phi, metRNA.RMSE_phi, ...
    metPID.RMSE_theta, metRNA.RMSE_theta, ...
    metPID.RMSE_psi, metRNA.RMSE_psi, ...
    metPID.Esf_u1, metRNA.Esf_u1, ...
    melhoriaRMSE, vencedorAltitude};
prefixo = [cenarios(c).id '_' cenarios(c).nome];
arqAltitude = fullfile(outDir, [prefixo '_altitude.png']);
arqRoll = fullfile(outDir, [prefixo '_roll.png']);
arqPitch = fullfile(outDir, [prefixo '_pitch.png']);
arqYaw = fullfile(outDir, [prefixo '_yaw.png']);
arqEmpuxo = fullfile(outDir, [prefixo '_empuxo.png']);
salvar_figura_variavel(ResPID.t, ResPID.X(:,1), ResRNA.X(:,1), R.z, ...
    'Altitude z (m)', arqAltitude)
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,3)),    rad2deg(ResRNA.X(:,3)),
rad2deg(R.phi), ...
    '\phi (graus)', arqRoll)
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,5)),    rad2deg(ResRNA.X(:,5)),
rad2deg(R.theta), ...
    '\theta (graus)', arqPitch)
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,7)),    rad2deg(ResRNA.X(:,7)),
rad2deg(R.psi), ...

```

```

        '\psi (graus)', arqYaw)
    salvar_figura_controle(ResPID.t, ResPID.U(:,1), ResRNA.U(:,1), ...
        'Empuxo u_1 (N)', arqEmpuxo)
    linhasFig(end+1,:) = {cenarios(c).id, cenarios(c).nome, cenarios(c).descricao, ...
        arqAltitude, arqRoll, arqPitch, arqYaw, arqEmpuxo};
end

TabelaFiguras = cell2table(linhasFig, 'VariableNames', { ...
    'Cenario', 'Identificacao', 'Descricao', ...
    'Figura_altitude', 'Figura_roll', 'Figura_pitch', 'Figura_yaw', 'Figura_empuxo'});
TabelaMetricas = cell2table(linhasMet, 'VariableNames', { ...
    'Cenario', 'Identificacao', 'Descricao', ...
    'RMSE_z_PID', 'RMSE_z_PID_neural_u1', ...
    'MAE_z_PID', 'MAE_z_PID_neural_u1', ...
    'RMSE_phi_PID', 'RMSE_phi_PID_neural_u1', ...
    'RMSE_theta_PID', 'RMSE_theta_PID_neural_u1', ...
    'RMSE_psi_PID', 'RMSE_psi_PID_neural_u1', ...
    'Esforco_u1_PID', 'Esforco_u1_PID_neural_u1', ...
    'Melhoria_RMSE_z_percentual', 'Vencedor_altitude'});
writetable(TabelaFiguras, fullfile(outDir,
'Tabela_arquivos_figuras_residual_apenas_u1.xlsx'))
writetable(TabelaFiguras, fullfile(outDir, 'Tabela_arquivos_figuras_residual_apenas_u1.csv'))
writetable(TabelaMetricas, fullfile(outDir, 'Tabela_metricas_residual_apenas_u1.xlsx'))
writetable(TabelaMetricas, fullfile(outDir, 'Tabela_metricas_residual_apenas_u1.csv'))
disp(TabelaMetricas)
fprintf('\nFiguras e tabelas salvas em: %s\n', outDir)

%% Funcoes locais

function D = alterar_planta(Dnom, fm, fl, fk)

    D = Dnom;
    D.m = fm*Dnom.m;
    D.lxx = fl*Dnom.lxx;
    D.lyy = fl*Dnom.lyy;
    D.lzz = fl*Dnom.lzz;
    D.kz = fk*Dnom.kz;
    D.kphi = fk*Dnom.kphi;

```

```

    D.ktheta = fk*Dnom.ktheta;
    D.kpsi = fk*Dnom.kpsi;
end
function entrada = montar_entrada_rede(x, R, Dplant, Dnom)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    erros = [R.z-z; -dz; R.phi-phi; -p; R.theta-theta; -q; R.psi-psi; -r];
    fatores = [
        Dplant.m/Dnom.m
        Dplant.Ixx/Dnom.Ixx
        Dplant.Iyy/Dnom.Iyy
        Dplant.Izz/Dnom.Izz
        Dplant.kz/Dnom.kz
        Dplant.kphi/Dnom.kphi
        Dplant.ktheta/Dnom.ktheta
        Dplant.kpsi/Dnom.kpsi
    ];
    entrada = [erros; fatores];
end
function u = pid_base_local(x, R, D)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    ez = R.z - z;
    ephi = R.phi - phi;
    etheta = R.theta - theta;
    epsi = R.psi - psi;
    Kpz = 10;    Kdz = 6;
    Kpphi = 0.8; Kdphi = 0.15;
    Kptheta = 0.8; Kdtheta = 0.15;
    Kppsi = 1.2; Kdpsi = 0.20;

```

```

u1 = D.m*(D.g + Kpz*ez - Kdz*dz)/(cos(phi)*cos(theta));
u2 = Kpphi*ephi - Kdphi*p;
u3 = Kptheta*etheta - Kdtheta*q;
u4 = Kppsi*epsi - Kdpsi*r;
u = saturar_local([u1 u2 u3 u4], D);
end
function Res = simular_local(Dctrl, Dplant, R, Ts, Tf, redeResidual, alpha,
usarRNAapenasU1)
t = (0:Ts:Tf)';
N = numel(t);
X = zeros(N,8);
U = zeros(N,4);
E = zeros(N,4);
X(1,:) = [0 0 deg2rad(8) 0 deg2rad(-5) 0 deg2rad(10) 0];
for k = 1:N-1
    x = X(k,:);
    uPID = pid_base_local(x, R, Dctrl);
    u = uPID;
    if ~isempty(redeResidual) && usarRNAapenasU1
        entrada = montar_entrada_rede(x, R, Dplant, Dctrl);
        uCorr = redeResidual(entrada);
        u(1) = uPID(1) + alpha*uCorr(1);
        u(2) = uPID(2);
        u(3) = uPID(3);
        u(4) = uPID(4);
    end
    u = saturar_local(u, Dctrl);
    dx = planta_hover_local(x, u, Dplant);
    X(k+1,:) = x + Ts*dx(:)';
    U(k,:) = u;
    E(k,:) = [R.z-x(1), R.phi-x(3), R.theta-x(5), R.psi-x(7)];
end
U(N,:) = U(N-1,:);
E(N,:) = E(N-1,:);

```

```

    Res.t = t;
    Res.X = X;
    Res.U = U;
    Res.E = E;
    Res.R = R;
    Res.Dctrl = Dctrl;
    Res.Dplant = Dplant;
end

function dx = planta_hover_local(x, u, D)
    dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    r = x(8);
    u1 = u(1);
    u2 = u(2);
    u3 = u(3);
    u4 = u(4);
    ddz = (u1/D.m)*cos(phi)*cos(theta) - D.g - D.kz*dz;
    dp = u2/D.Ixx - D.kphi*p;
    dq = u3/D.Iyy - D.ktheta*q;
    dr = u4/D.Izz - D.kpsi*r;
    dx = [dz; ddz; p; dp; q; dq; r; dr];
end

function u = saturar_local(u, D)
    u(1) = min(max(u(1), D.minThrust), D.maxThrust);
    u(2) = min(max(u(2), -D.maxTorque), D.maxTorque);
    u(3) = min(max(u(3), -D.maxTorque), D.maxTorque);
    u(4) = min(max(u(4), -D.maxTorque), D.maxTorque);
end

function met = metricas_locais(Res)
    E = Res.E;
    U = Res.U;
    met.MAE_z = mean(abs(E(:,1)));
    met.RMSE_z = sqrt(mean(E(:,1).^2));

```

```

    met.RMSE_phi = sqrt(mean(E(:,2).^2));
    met.RMSE_theta = sqrt(mean(E(:,3).^2));
    met.RMSE_psi = sqrt(mean(E(:,4).^2));
    met.Esf_u1 = mean(abs(U(:,1)));
end

function salvar_figura_variavel(t, yPID, yRNA, ref, ylabelTxt, nomeArquivo)
    f = figure('Color','w','Visible','off');
    plot(t, yPID, 'b', 'LineWidth', 2)
    hold on
    plot(t, yRNA, 'r--', 'LineWidth', 2)
    yline(ref, 'k:', 'LineWidth', 1.5)
    grid on
    xlabel('Tempo (s)')
    ylabel(ylabelTxt)
    legend('PID', 'PID-Neural residual u_1', 'Referencia', 'Location', 'best')
    set(gca, 'FontSize', 12)
    exportgraphics(f, nomeArquivo, 'Resolution', 300)
    close(f)
end

function salvar_figura_controle(t, yPID, yRNA, ylabelTxt, nomeArquivo)
    f = figure('Color','w','Visible','off');
    plot(t, yPID, 'b', 'LineWidth', 2)
    hold on
    plot(t, yRNA, 'r--', 'LineWidth', 2)
    grid on
    xlabel('Tempo (s)')
    ylabel(ylabelTxt)
    legend('PID', 'PID-Neural residual u_1', 'Location', 'best')
    set(gca, 'FontSize', 12)
    exportgraphics(f, nomeArquivo, 'Resolution', 300)
    close(f)
end

```

APÊNDICE B — TREINAMENTO E RETREINAMENTO DA REDE NEURAL

Este item apresenta os procedimentos utilizados para a geração da base de dados empregada no treinamento da rede neural. No projeto, essa etapa está associada aos arquivos `gerarDataset.m`, `preProcessarDataset.m` e `analisarDataset.m`, além dos dados armazenados em `dadosRNA.mat`. A base de dados contempla informações dos estados do sistema, erros de controle, sinais de referência e demais variáveis utilizadas para compor as entradas e saídas da RNA. O código principal de geração da base de dados é apresentado a seguir.

B.1 Geração da base de dados

B.1.1 Geração do dataset

```
function gerarDataset()
clc
clear
close all

disp('=====')
disp('GERADOR DE DATASET PARA PID-NEURAL')
disp('=====')

%%%%%%%%%%%%%%
Número de simulações
%%%%%%%%%%%%%%
Nsim = 300;

%%%%%%%%%%%%%%
Banco de Dados
%%%%%%%%%%%%%%
Dataset = cell(Nsim,1);
BancoEntrada = [];
BancoSaida = [];

%%%%%%%%%%%%%%
Barra de progresso
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
h = waitbar(0,'Gerando Dataset...');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Loop Principal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for k = 1:Nsim
```

```
    waitbar(k/Nsim,h,...
```

```
        sprintf('Simulação %d de %d',k,Nsim));
```

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

```
    % Condição inicial
```

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

```
    X0 = gerarCondicaoInicial();
```

```
    R = gerarReferencia();
```

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

```
    % Executa Simulação
```

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

```
    Resultado = executarSimulacao(...
```

```
        @controladorPIDAdaptativo,...
```

```
        X0,...
```

```
        R);
```

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

```
    % Extrai Dados
```

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

```
    Dados = extrairDados(Resultado);
```

```
    Dados.Referencia = R;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Seleção Inteligente das Amostras

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    tempo = Dados.t;
```

```
    erroZ    = Dados.Entrada(:,5);
```

```
    erroPhi  = Dados.Entrada(:,6);
```

```
    erroTheta = Dados.Entrada(:,7);
```

```
    erroPsi   = Dados.Entrada(:,8);
```

```
    u2 = Dados.Saida(:,2);
```

```

u3 = Dados.Saida(:,3);
u4 = Dados.Saida(:,4);
indice = tempo <= 5 ...
    | abs(erroZ)    > 0.01 ...
    | abs(erroPhi) > deg2rad(0.5) ...
    | abs(erroTheta) > deg2rad(0.5) ...
    | abs(erroPsi) > deg2rad(1.0) ...
    | abs(u2) > 0.005 ...
    | abs(u3) > 0.005 ...
    | abs(u4) > 0.005;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Guarda apenas amostras relevantes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Dados.Entrada = Dados.Entrada(indice,:);
Dados.Saida = Dados.Saida(indice,:);
Dados.t = Dados.t(indice);
Dados.X = Dados.X(indice,:);
Dados.U = Dados.U(indice,:);
Dados.E = Dados.E(indice,:);

%-----
% Guarda voo completo
%-----

Dataset{k} = Dados;

%-----
% Banco consolidado
%-----

BancoEntrada = [BancoEntrada
                Dados.Entrada];
BancoSaida = [BancoSaida
              Dados.Saida];

end

close(h)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Salvar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save('dadosTreinamento.mat', ...
    'Dataset', ...
    'BancoEntrada', ...
    'BancoSaida');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Estatísticas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n');
fprintf('=====\\n');
fprintf('DATASET GERADO COM SUCESSO\\n');
fprintf('=====\\n');
fprintf('Número de voos.....: %d\\n',Nsim);
fprintf('Número de amostras.....: %d\\n',size(BancoEntrada,1));
fprintf('Entradas da RNA.....: %d\\n',size(BancoEntrada,2));
fprintf('Saídas da RNA.....: %d\\n',size(BancoSaida,2));
fprintf('=====\\n');
end

```

B.1.2 Pré-processamento do dataset

```

function [Treino,Validacao,Teste] = preProcessarDataset(arquivoEntrada,arquivoSaida)
if nargin < 1
    arquivoEntrada = 'dadosTreinamento.mat';
end
if nargin < 2
    arquivoSaida = 'dadosRNA.mat';
end
clc
disp('=====')
disp('PRE-PROCESSAMENTO DO DATASET')
disp('=====')
load(arquivoEntrada,'BancoEntrada','BancoSaida')

```

```

if size(BancoEntrada,1) ~= size(BancoSaida,1)
    error('BancoEntrada e BancoSaida devem ter o mesmo numero de amostras.')
end
rng(1)
N = size(BancoEntrada,1);
idx = randperm(N);
nTreino = round(0.70*N);
nValidacao = round(0.15*N);
idxTreino = idx(1:nTreino);
idxValidacao = idx(nTreino+1:nTreino+nValidacao);
idxTeste = idx(nTreino+nValidacao+1:end);
Treino.X = BancoEntrada(idxTreino,:);
Treino.Y = BancoSaida(idxTreino,:);
Validacao.X = BancoEntrada(idxValidacao,:);
Validacao.Y = BancoSaida(idxValidacao,:);
Teste.X = BancoEntrada(idxTeste,:);
Teste.Y = BancoSaida(idxTeste,:);
save(arquivoSaida,'Treino','Validacao','Teste')
fprintf('Amostras de treino.....: %d\n',size(Treino.X,1));
fprintf('Amostras de validacao.: %d\n',size(Validacao.X,1));
fprintf('Amostras de teste.....: %d\n',size(Teste.X,1));
fprintf('Entradas da RNA.....: %d\n',size(Treino.X,2));
fprintf('Saidas da RNA.....: %d\n',size(Treino.Y,2));
fprintf('Arquivo salvo em.....: %s\n',arquivoSaida);
end

```

B.1.3 Análise do dataset

```

function analisarDataset()
clc
clear
close all
disp('=====')
disp('ANÁLISE DO DATASET')
disp('=====')

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Carregar Dataset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load('dadosTreinamento.mat')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Informações Gerais
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n');
fprintf('Número de amostras : %d\n',size(BancoEntrada,1));
fprintf('Número de entradas : %d\n',size(BancoEntrada,2));
fprintf('Número de saídas : %d\n',size(BancoSaida,2));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Verificação de NaN e Inf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n');
fprintf('NaN Entrada : %d\n',any(isnan(BancoEntrada(:))));
fprintf('Inf Entrada : %d\n',any(isinf(BancoEntrada(:))));
fprintf('NaN Saída : %d\n',any(isnan(BancoSaida(:))));
fprintf('Inf Saída : %d\n',any(isinf(BancoSaida(:))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Estatísticas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
media = mean(BancoEntrada);
desvio = std(BancoEntrada);
minimo = min(BancoEntrada);
maximo = max(BancoEntrada);
Tabela = table( ...
    media',...
    desvio',...
    minimo',...
    maximo',...
    'VariableNames',...
    {'Media','DesvioPadrao','Minimo','Maximo'});
disp(Tabela)

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Histograma das Entradas

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
nomes = {...
```

```
    'z','phi','theta','psi',...
```

```
    'erro_z','erro_phi','erro_theta','erro_psi',...
```

```
    'int_z','int_phi','int_theta','int_psi',...
```

```
    'dz','p','q','r'};
```

```
figure('Name','Histogramas das Entradas')
```

```
for k = 1:16
```

```
    subplot(4,4,k)
```

```
    histogram(BancoEntrada(:,k),40)
```

```
    grid on
```

```
    title(nomes{k})
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Histogramas das Saídas

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
figure('Name','Histogramas das Saidas')
```

```
nomesU = {'u1','u2','u3','u4'};
```

```
for k = 1:4
```

```
    subplot(2,2,k)
```

```
    histogram(BancoSaida(:,k),40)
```

```
    grid on
```

```
    title(nomesU{k})
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Correlação

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
R = corrcoef(BancoEntrada);
```

```
figure('Name','Matriz de Correlação')
```

```
imagesc(R)
```

```
axis square
```

```
colorbar
```

```

title('Correlação das Entradas')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Boxplot
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Name','Boxplot')
boxplot(BancoEntrada)
grid on
title('Entradas')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Outliers
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
outliers = isoutlier(BancoEntrada);
fprintf('\n');
fprintf('Número de Outliers : %d\n',nnz(outliers));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Relatório
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n');
fprintf('=====\\n');
fprintf('ANÁLISE FINALIZADA\\n');
fprintf('=====\\n');
end

```

B.2 Configuração da rede neural

Este item apresenta a configuração da rede neural utilizada no retreinamento residual, contemplando o número de entradas, o número de saídas, a arquitetura adotada, a quantidade de neurônios nas camadas ocultas, o algoritmo de treinamento e a divisão dos dados em conjuntos de treinamento, validação e teste.

Tabela B.1 — Configurações utilizadas no retreinamento da RNA residual.

Parametro	Valor
Tipo de rede	Rede neural feedforward para ajuste de funcao
Funcao <i>MATLAB</i> utilizada	fitnet

Arquitetura	Duas camadas ocultas com 24 e 16 neuronios
Algoritmo de treinamento	Levenberg-Marquardt (trainlm)
Numero de amostras	6000
Numero de entradas	16
Numero de saidas	4
Entradas da rede	Erros de altitude e atitude, velocidades associadas e fatores relativos da planta
Saidas da rede	Correcoes residuais para u1, u2, u3 e u4
Divisao dos dados	70% treinamento, 15% validacao e 15% teste
Numero maximo de epocas	600
Meta de desempenho	1e-7
Fator de ponderacao alpha	1,0
Lei de controle avaliada	$u = u_PID + \alpha * u_RNA, res$
Cenarios considerados	Variacoes de massa, inercias, amortecimento e cenario combinado
Arquivo da rede treinada	rede_residual_pid_neural.mat

Fonte: Elaborado pelo autor (2014).

B.3 Retreinamento da RNA residual

Este item apresenta o script utilizado para o retreinamento da rede neural residual. Esse procedimento foi desenvolvido com o objetivo de estimar uma correção adicional ao sinal de controle gerado pelo PID, permitindo avaliar a contribuição da RNA como elemento complementar na estabilização em voo pairado. O código correspondente é apresentado a seguir.

```
%% testar_retreino_rna_residual.m
% Teste de retreinamento de uma RNA residual para o controlador PID-Neural.
% Ideia testada:
% Em vez de a RNA substituir o PID, ela aprende uma correcao residual:
%   U = U_PID + alpha*U_RNA_residual
% O projeto original nao e alterado. A rede treinada, tabelas e figuras sao
% salvas em uma pasta de resultados.
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
%   /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:
```

```

%% testar_retreino_rna_residual
clear
clc
close all
rng(42)
%% Pasta do projeto
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)
end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
%% Parametros nominais
Dnom = parametrosDrone();
R = referencias();
Ts = 0.01;
Tf = 20;
Dnom.Ts = Ts;
%% Configuracoes do treinamento
nAmostras = 6000;
alpha = 1.0;
% Entrada da rede:
% [erros e velocidades; fatores relativos de massa, inercias e amortecimentos]
nEntradas = 16;
nSaidas = 4;
Xtrain = zeros(nEntradas, nAmostras);
Ytrain = zeros(nSaidas, nAmostras);
for i = 1:nAmostras
    Dplant = sortear_planta(Dnom);
    x = sortear_estado();
    uPID = pid_base_local(x, R, Dnom);
    uIdeal = controle_inverso_local(x, R, Dplant);

```

```

    uResidual = uIdeal - uPID;
    Xtrain(:,i) = montar_entrada_rede(x, R, Dplant, Dnom);
    Ytrain(:,i) = uResidual(:);
end

%% Treinamento da rede residual
redeResidual = fitnet([24 16], 'trainlm');
redeResidual.trainParam.showWindow = true;
redeResidual.trainParam.epochs = 600;
redeResidual.trainParam.goal = 1e-7;
redeResidual.divideParam.trainRatio = 0.70;
redeResidual.divideParam.valRatio = 0.15;
redeResidual.divideParam.testRatio = 0.15;
[redeResidual, tr] = train(redeResidual, Xtrain, Ytrain);
save(fullfile(outDir, 'rede_residual_pid_neural.mat'), ...
    'redeResidual', 'tr', 'Xtrain', 'Ytrain', 'alpha')

%% Cenarios de teste
cenarios = struct([]);
cenarios(1).nome = 'massa_menor_20';
cenarios(1).descricao = 'Massa reduzida em 20%';
cenarios(1).D = alterar_planta(Dnom, 0.80, 1.00, 1.00, 1.00);
cenarios(2).nome = 'massa_maior_20';
cenarios(2).descricao = 'Massa aumentada em 20%';
cenarios(2).D = alterar_planta(Dnom, 1.20, 1.00, 1.00, 1.00);
cenarios(3).nome = 'inercias_maiores';
cenarios(3).descricao = 'Inercias aumentadas em 20%';
cenarios(3).D = alterar_planta(Dnom, 1.00, 1.20, 1.00, 1.00);
cenarios(4).nome = 'amortecimento_menor';
cenarios(4).descricao = 'Amortecimento reduzido em 30%';
cenarios(4).D = alterar_planta(Dnom, 1.00, 1.00, 0.70, 1.00);
cenarios(5).nome = 'combinado';
cenarios(5).descricao = 'Massa menor, inercias maiores e amortecimento menor';
cenarios(5).D = alterar_planta(Dnom, 0.80, 1.20, 0.70, 1.00);
linhas = {};
for c = 1:numel(cenarios)

```

```

Dplant = cenarios(c).D;
ResPID = simular_local(Dnom, Dplant, R, Ts, Tf, [], 0);
ResRNA = simular_local(Dnom, Dplant, R, Ts, Tf, redeResidual, alpha);
metPID = metricas_locais(ResPID);
metRNA = metricas_locais(ResRNA);
if metRNA.RMSE_z < metPID.RMSE_z
    vencedor = 'PID-Neural residual';
else
    vencedor = 'PID';
end
melhoriaRMSE = 100*(metPID.RMSE_z - metRNA.RMSE_z)/metPID.RMSE_z;
linhas(end+1,:) = { ...
    cenarios(c).nome, cenarios(c).descricao, ...
    metPID.RMSE_z, metRNA.RMSE_z, ...
    metPID.MAE_z, metRNA.MAE_z, ...
    metPID.Esf_u1, metRNA.Esf_u1, ...
    melhoriaRMSE, vencedor};
prefixo = cenarios(c).nome;
salvar_figura_variavel(ResPID.t, ResPID.X(:,1), ResRNA.X(:,1), R.z, ...
    'Altitude z (m)', fullfile(outDir, [prefixo '_altitude.png']))
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,3)),    rad2deg(ResRNA.X(:,3)),
rad2deg(R.phi), ...
    '\phi (graus)', fullfile(outDir, [prefixo '_roll.png']))
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,5)),    rad2deg(ResRNA.X(:,5)),
rad2deg(R.theta), ...
    '\theta (graus)', fullfile(outDir, [prefixo '_pitch.png']))
salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,7)),    rad2deg(ResRNA.X(:,7)),
rad2deg(R.psi), ...
    '\psi (graus)', fullfile(outDir, [prefixo '_yaw.png']))
salvar_figura_controle(ResPID.t, ResPID.U(:,1), ResRNA.U(:,1), ...
    'Empuxo u_1 (N)', fullfile(outDir, [prefixo '_empuxo.png']))
end
Tabela = cell2table(linhas, 'VariableNames', { ...
    'Cenario','Descricao', ...

```

```

'RMSE_z_PID','RMSE_z_PID_neural_residual', ...
'MAE_z_PID','MAE_z_PID_neural_residual', ...
'Esforco_u1_PID','Esforco_u1_PID_neural_residual', ...
'Melhoria_RMSE_percentual','Vencedor'});
disp(Tabela)
writetable(Tabela, fullfile(outDir, 'Tabela_retreino_RNA_residual.xlsx'))
writetable(Tabela, fullfile(outDir, 'Tabela_retreino_RNA_residual.csv'))
fprintf('\nProcesso concluido.\n')
fprintf('Resultados salvos em: %s\n', outDir)
%% Funcoes locais
function D = sortear_planta(Dnom)
    fm = 0.75 + 0.50*rand();
    fl = 0.80 + 0.50*rand();
    fk = 0.60 + 0.80*rand();
    D = alterar_planta(Dnom, fm, fl, fk, 1.00);
end
function D = alterar_planta(Dnom, fm, fl, fk, fTorque)
    D = Dnom;
    D.m = fm*Dnom.m;
    D.Ixx = fl*Dnom.Ixx;
    D.Iyy = fl*Dnom.Iyy;
    D.Izz = fl*Dnom.Izz;
    D.kz = fk*Dnom.kz;
    D.kphi = fk*Dnom.kphi;
    D.ktheta = fk*Dnom.ktheta;
    D.kpsi = fk*Dnom.kpsi;
    D.maxTorque = fTorque*Dnom.maxTorque;
end
function x = sortear_estado()
    z = 1.0*rand();
    dz = -0.8 + 1.6*rand();
    phi = deg2rad(-18 + 36*rand());
    p = deg2rad(-45 + 90*rand());
    theta = deg2rad(-18 + 36*rand());

```

```

q = deg2rad(-45 + 90*rand());
psi = deg2rad(-25 + 50*rand());
r = deg2rad(-60 + 120*rand());
x = [z dz phi p theta q psi r];
end
function entrada = montar_entrada_rede(x, R, Dplant, Dnom)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    erros = [R.z-z; -dz; R.phi-phi; -p; R.theta-theta; -q; R.psi-psi; -r];
    fatores = [
        Dplant.m/Dnom.m
        Dplant.Ixx/Dnom.Ixx
        Dplant.Iyy/Dnom.Iyy
        Dplant.Izz/Dnom.Izz
        Dplant.kz/Dnom.kz
        Dplant.kphi/Dnom.kphi
        Dplant.ktheta/Dnom.ktheta
        Dplant.kpsi/Dnom.kpsi
    ];
    entrada = [erros; fatores];
end
function u = pid_base_local(x, R, D)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    ez = R.z - z;
    ephi = R.phi - phi;
    etheta = R.theta - theta;
    epsi = R.psi - psi;
    Kpz = 10;    Kdz = 6;
    Kpphi = 0.8; Kdphi = 0.15;

```

```

Kptheta = 0.8; Kdtheta = 0.15;
Kppsi = 1.2; Kdpsi = 0.20;
u1 = D.m*(D.g + Kpz*ez - Kdz*dz)/(cos(phi)*cos(theta));
u2 = Kpphi*ephi - Kdphi*p;
u3 = Kptheta*etheta - Kdtheta*q;
u4 = Kppsi*epsi - Kdpsi*r;
u = saturar_local([u1 u2 u3 u4], D);
end

function u = controle_inverso_local(x, R, D)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    ez = R.z - z;
    ephi = R.phi - phi;
    etheta = R.theta - theta;
    epsi = R.psi - psi;
    Kpz = 10; Kdz = 6;
    Kpphi = 0.8; Kdphi = 0.15;
    Kptheta = 0.8; Kdtheta = 0.15;
    Kppsi = 1.2; Kdpsi = 0.20;
    azDesejada = Kpz*ez - Kdz*dz;
    dpDesejado = Kpphi*ephi - Kdphi*p;
    dqDesejado = Kptheta*etheta - Kdtheta*q;
    drDesejado = Kppsi*epsi - Kdpsi*r;
    u1 = D.m*(D.g + D.kz*dz + azDesejada)/(cos(phi)*cos(theta));
    u2 = D.Ixx*(dpDesejado + D.kphi*p);
    u3 = D.Iyy*(dqDesejado + D.ktheta*q);
    u4 = D.Izz*(drDesejado + D.kpsi*r);
    u = saturar_local([u1 u2 u3 u4], D);
end

function Res = simular_local(Dctrl, Dplant, R, Ts, Tf, redeResidual, alpha)
    t = (0:Ts:Tf)';
    N = numel(t);

```

```

X = zeros(N,8);
U = zeros(N,4);
E = zeros(N,4);
X(1,:) = [0 0 deg2rad(8) 0 deg2rad(-5) 0 deg2rad(10) 0];
for k = 1:N-1
    x = X(k,:);
    uPID = pid_base_local(x, R, Dctrl);
    if isempty(redeResidual)
        u = uPID;
    else
        entrada = montar_entrada_rede(x, R, Dplant, Dctrl);
        uCorr = redeResidual(entrada);
        u = uPID + alpha*uCorr(:)';
    end
    u = saturar_local(u, Dctrl);
    dx = planta_hover_local(x, u, Dplant);
    X(k+1,:) = x + Ts*dx(:)';
    U(k,:) = u;
    E(k,:) = [R.z-x(1), R.phi-x(3), R.theta-x(5), R.psi-x(7)];
end
U(N,:) = U(N-1,:);
E(N,:) = E(N-1,:);
Res.t = t;
Res.X = X;
Res.U = U;
Res.E = E;
Res.R = R;
Res.Dctrl = Dctrl;
Res.Dplant = Dplant;
end
function dx = planta_hover_local(x, u, D)
    dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);

```

```

    r = x(8);
    u1 = u(1);
    u2 = u(2);
    u3 = u(3);
    u4 = u(4);
    ddz = (u1/D.m)*cos(phi)*cos(theta) - D.g - D.kz*dz;
    dp = u2/D.Ixx - D.kphi*p;
    dq = u3/D.Iyy - D.ktheta*q;
    dr = u4/D.Izz - D.kpsi*r;
    dx = [dz; ddz; p; dp; q; dq; r; dr];
end

function u = saturar_local(u, D)
    u(1) = min(max(u(1), D.minThrust), D.maxThrust);
    u(2) = min(max(u(2), -D.maxTorque), D.maxTorque);
    u(3) = min(max(u(3), -D.maxTorque), D.maxTorque);
    u(4) = min(max(u(4), -D.maxTorque), D.maxTorque);
end

function met = metricas_locais(Res)
    Ts = Res.t(2) - Res.t(1);
    E = Res.E;
    U = Res.U;
    met.MAE_z = mean(abs(E(:,1)));
    met.RMSE_z = sqrt(mean(E(:,1).^2));
    met.ISE_z = sum(E(:,1).^2)*Ts;
    met.RMSE_phi = sqrt(mean(E(:,2).^2));
    met.RMSE_theta = sqrt(mean(E(:,3).^2));
    met.RMSE_psi = sqrt(mean(E(:,4).^2));
    met.Esf_u1 = mean(abs(U(:,1)));
end

function salvar_figura_variavel(t, yPID, yRNA, ref, ylabelTxt, nomeArquivo)
    f = figure('Color','w','Visible','off');
    plot(t, yPID, 'b', 'LineWidth', 2)
    hold on
    plot(t, yRNA, 'r--', 'LineWidth', 2)

```

```

    yline(ref, 'k:', 'LineWidth', 1.5)
    grid on
    xlabel('Tempo (s)')
    ylabel(ylabelTxt)
    legend('PID', 'PID-Neural residual', 'Referencia', 'Location', 'best')
    set(gca, 'FontSize', 12)
    exportgraphics(f, nomeArquivo, 'Resolution', 300)
    close(f)
end
function salvar_figura_controle(t, yPID, yRNA, ylabelTxt, nomeArquivo)
    f = figure('Color','w','Visible','off');
    plot(t, yPID, 'b', 'LineWidth', 2)
    hold on
    plot(t, yRNA, 'r--', 'LineWidth', 2)
    grid on
    xlabel('Tempo (s)')
    ylabel(ylabelTxt)
    legend('PID', 'PID-Neural residual', 'Location', 'best')
    set(gca, 'FontSize', 12)
    exportgraphics(f, nomeArquivo, 'Resolution', 300)
    close(f)
end

```

A partir dos resultados obtidos com o retreinamento, verificou-se que a atuação neural residual aplicada a todos os canais poderia comprometer as respostas angulares. Por esse motivo, a formulação final restringiu a contribuição neural ao canal de empuxo u_1 , conforme apresentado no item B.4.

B.4 Formulação final do PID-Neural Residual

A formulação final adotada restringe a atuação da rede neural residual ao canal de empuxo, preservando os torques de atitude sob responsabilidade do controlador PID. Essa configuração foi adotada porque a atuação neural em todos os canais melhorou a altitude, mas comprometeu as respostas angulares.

$$u_1 = u_{1,PID} + \alpha u_{1,RNA,res} \quad (109)$$

$$u_2 = u_{2,PID}, u_3 = u_{3,PID}, u_4 = u_{4,PID} \quad (110)$$

em que u_1 representa o empuxo total, $u_{1,PID}$ é o empuxo calculado pelo controlador PID, $u_{1,RNA,res}$ é a correção residual estimada pela rede neural e α é o fator de ponderação da contribuição neural.

APÊNDICE C — GERAÇÃO DE FIGURAS, TABELAS E MÉTRICAS

Este apêndice reúne os scripts utilizados para gerar as figuras, tabelas e indicadores quantitativos apresentados no Capítulo 6. Esses arquivos foram empregados para organizar os resultados das simulações, gerar os gráficos de resposta temporal e calcular métricas como MAE, RMSE, ISE e esforço médio de controle.

C.1 Script de geração do diagrama de blocos

O diagrama de blocos da estrutura de simulação em malha fechada foi gerado por meio do script Python `gerar_diagrama_blocos_hover.py`. A figura resultante foi salva no arquivo `figura_6x_diagrama_blocos_hover.svg`.

```
from pathlib import Path

ROOT = Path(__file__).resolve().parents[1]
OUTPUTS = ROOT / "outputs"
OUTPUTS.mkdir(exist_ok=True)
svg_path = OUTPUTS / "figura_6x_diagrama_blocos_hover.svg"
def box(x, y, w, h, title, subtitle="", fill="#F7F9FC", stroke="#2F3A4A"):
    lines = [
        f'<rect x="{x}" y="{y}" width="{w}" height="{h}" rx="8" '
        f'fill="{fill}" stroke="{stroke}" stroke-width="2"/>',
        f'<text x="{x + w / 2}" y="{y + 28}" text-anchor="middle" '
        f'font-family="Arial, Helvetica, sans-serif" font-size="16" '
        f'font-weight="700" fill="#172033">{title}</text>',
    ]
    if subtitle:
        for i, line in enumerate(subtitle.split("\n")):
            lines.append(
                f'<text x="{x + w / 2}" y="{y + 53 + 20 * i}" text-anchor="middle" '
                f'font-family="Arial, Helvetica, sans-serif" font-size="13" '
                f'fill="#334155">{line}</text>'
            )
```

```

    )
    return "\n".join(lines)
def arrow(x1, y1, x2, y2, label=""):
    mid_x = (x1 + x2) / 2
    mid_y = (y1 + y2) / 2
    label_svg = ""
    if label:
        label_svg = (
            f'<text x="{mid_x}" y="{mid_y - 10}" text-anchor="middle" '
            f'font-family="Arial, Helvetica, sans-serif" font-size="12" '
            f'fill="#475569">{label}</text>'
        )
    return (
        f'<line x1="{x1}" y1="{y1}" x2="{x2}" y2="{y2}" '
        f'stroke="#1E293B" stroke-width="2.2" marker-end="url(#arrowhead)"/>'
        f'\n{label_svg}'
    )
svg = f'<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" width="1180" height="520" viewBox="0 0 1180
520">
<defs>
<marker id="arrowhead" markerWidth="10" markerHeight="8" refX="9" refY="4"
orient="auto">
<polygon points="0 0, 10 4, 0 8" fill="#1E293B"/>
</marker>
<style>
.caption {{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 18px;
    font-weight: 700;
    fill: #111827;
}}
.note {{
    font-family: Arial, Helvetica, sans-serif;

```

```

    font-size: 13px;
    fill: #475569;
  }}
  .small {{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    fill: #334155;
  }}
</style>
</defs>
<rect x="0" y="0" width="1180" height="520" fill="#FFFFFF"/>
<text x="590" y="42" text-anchor="middle" class="caption">
  Estrutura da simulação em malha fechada para voo pairado
</text>
{box(60, 160, 180, 96, "Referências", "z_ref, phi_ref\n theta_ref, psi_ref", "#EEF6FF",
"#2563EB")}
{box(300, 160, 130, 96, "Erro", "e = ref - estado", "#F8FAFC", "#475569")}
{box(490, 138, 220, 140, "Controlador", "PID convencional\nou PID-Neural\nu1, u2, u3,
u4", "#F0FDF4", "#16A34A")}
{box(770, 138, 240, 140, "Planta reduzida", "Dinâmica de hover\naltitude e atitude",
"#FFF7ED", "#EA580C")}
{box(770, 328, 240, 98, "Estados simulados", "z, dz, phi, theta, psi\np, q, r", "#FEFCE8",
"#CA8A04")}
{arrow(240, 208, 300, 208)}
{arrow(430, 208, 490, 208)}
{arrow(710, 208, 770, 208, "sinais de controle")}
{arrow(890, 278, 890, 328, "resposta")}
<path d="M 770 377 L 365 377 L 365 256" fill="none" stroke="#1E293B" stroke-
width="2.2" marker-end="url(#arrowhead)"/>
<text x="560" y="365" text-anchor="middle" class="note">realimentação dos
estados</text>
<text x="60" y="472" class="note">

```

Fonte: elaborado pelo autor com base na arquitetura computacional de simulação do controlador PID/PID-Neural.

```

</text>
</svg>
"""
svg_path.write_text(svg, encoding="utf-8")
print(svg_path)

```

C.2 Scripts de geração das figuras de resposta temporal

Este item apresenta os scripts utilizados para gerar as figuras de resposta temporal da altitude, dos ângulos de rolagem, arfagem e guinada, além do sinal de empuxo. Recomenda-se incluir principalmente o script final correspondente ao controlador PID-Neural residual aplicado apenas ao canal de empuxo u_1 , uma vez que essa foi a formulação adotada na análise final da dissertação. O código correspondente é apresentado a seguir.

```

%% gerar_figuras_residual_apenas_u1.m
% Gera figuras comparativas entre PID e PID-Neural residual com correção
% neural aplicada apenas ao canal de empuxo u1.
% Lei de controle avaliada:
%  $u_1 = u_{1\_PID} + \alpha * u_{1\_RNA\_res}$ 
%  $u_2 = u_{2\_PID}$ 
%  $u_3 = u_{3\_PID}$ 
%  $u_4 = u_{4\_PID}$ 
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
%   /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:
%   gerar_figuras_residual_apenas_u1
clear
clc
close all
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)

```

```

end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual', 'figuras_residual_apenas_u1');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
redeFile = fullfile(pwd, 'resultados_retreino_rna_residual', 'rede_residual_pid_neural.mat');
if exist(redeFile, 'file') ~= 2
    error(['Arquivo da rede residual nao encontrado: ', redeFile, newline, ...
        'Execute primeiro o script testar_retreino_rna_residual.m'])
end
S = load(redeFile);
redeResidual = S.redeResidual;
if isfield(S, 'alpha')
    alpha = S.alpha;
else
    alpha = 1.0;
end
Dnom = parametrosDrone();
R = referencias();
Ts = 0.01;
Tf = 20;
Dnom.Ts = Ts;
cenarios = struct([]);
cenarios(1).id = 'C01';
cenarios(1).nome = 'massa_menor_20';
cenarios(1).descricao = 'Massa reduzida em 20%';
cenarios(1).D = alterar_planta(Dnom, 0.80, 1.00, 1.00);
cenarios(2).id = 'C02';
cenarios(2).nome = 'massa_maior_20';
cenarios(2).descricao = 'Massa aumentada em 20%';
cenarios(2).D = alterar_planta(Dnom, 1.20, 1.00, 1.00);
cenarios(3).id = 'C03';
cenarios(3).nome = 'inercias_maiores';
cenarios(3).descricao = 'Inercias aumentadas em 20%';

```

```

cenarios(3).D = alterar_planta(Dnom, 1.00, 1.20, 1.00);
cenarios(4).id = 'C04';
cenarios(4).nome = 'amortecimento_menor';
cenarios(4).descricao = 'Amortecimento reduzido em 30%';
cenarios(4).D = alterar_planta(Dnom, 1.00, 1.00, 0.70);
cenarios(5).id = 'C05';
cenarios(5).nome = 'combinado';
cenarios(5).descricao = 'Massa menor, inercias maiores e amortecimento menor';
cenarios(5).D = alterar_planta(Dnom, 0.80, 1.20, 0.70);
linhasFig = {};
linhasMet = {};
for c = 1:numel(cenarios)
    Dplant = cenarios(c).D;
    ResPID = simular_local(Dnom, Dplant, R, Ts, Tf, [], alpha, false);
    ResRNA = simular_local(Dnom, Dplant, R, Ts, Tf, redeResidual, alpha, true);
    metPID = metricas_locais(ResPID);
    metRNA = metricas_locais(ResRNA);
    melhoriaRMSE = 100*(metPID.RMSE_z - metRNA.RMSE_z)/metPID.RMSE_z;
    if metRNA.RMSE_z < metPID.RMSE_z
        vencedorAltitude = 'PID-Neural residual u1';
    else
        vencedorAltitude = 'PID';
    end
    linhasMet(end+1,:) = { ...
        cenarios(c).id, cenarios(c).nome, cenarios(c).descricao, ...
        metPID.RMSE_z, metRNA.RMSE_z, ...
        metPID.MAE_z, metRNA.MAE_z, ...
        metPID.RMSE_phi, metRNA.RMSE_phi, ...
        metPID.RMSE_theta, metRNA.RMSE_theta, ...
        metPID.RMSE_psi, metRNA.RMSE_psi, ...
        metPID.Esf_u1, metRNA.Esf_u1, ...
        melhoriaRMSE, vencedorAltitude};
    prefixo = [cenarios(c).id '_' cenarios(c).nome];
    arqAltitude = fullfile(outDir, [prefixo '_altitude.png']);
end

```

```

    arqRoll = fullfile(outDir, [prefixo '_roll.png']);
    arqPitch = fullfile(outDir, [prefixo '_pitch.png']);
    arqYaw = fullfile(outDir, [prefixo '_yaw.png']);
    arqEmpuxo = fullfile(outDir, [prefixo '_empuxo.png']);
    salvar_figura_variavel(ResPID.t, ResPID.X(:,1), ResRNA.X(:,1), R.z, ...
        'Altitude z (m)', arqAltitude)
    salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,3)),    rad2deg(ResRNA.X(:,3)),
rad2deg(R.phi), ...
        '\phi (graus)', arqRoll)
    salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,5)),    rad2deg(ResRNA.X(:,5)),
rad2deg(R.theta), ...
        '\theta (graus)', arqPitch)
    salvar_figura_variavel(ResPID.t,    rad2deg(ResPID.X(:,7)),    rad2deg(ResRNA.X(:,7)),
rad2deg(R.psi), ...
        '\psi (graus)', arqYaw)
    salvar_figura_controle(ResPID.t, ResPID.U(:,1), ResRNA.U(:,1), ...
        'Empuxo u_1 (N)', arqEmpuxo)
    linhasFig(end+1,:) = {cenarios(c).id, cenarios(c).nome, cenarios(c).descricao, ...
        arqAltitude, arqRoll, arqPitch, arqYaw, arqEmpuxo};
end
TabelaFiguras = cell2table(linhasFig, 'VariableNames', { ...
    'Cenario', 'Identificacao', 'Descricao', ...
    'Figura_altitude', 'Figura_roll', 'Figura_pitch', 'Figura_yaw', 'Figura_empuxo'});
TabelaMetricas = cell2table(linhasMet, 'VariableNames', { ...
    'Cenario', 'Identificacao', 'Descricao', ...
    'RMSE_z_PID', 'RMSE_z_PID_neural_u1', ...
    'MAE_z_PID', 'MAE_z_PID_neural_u1', ...
    'RMSE_phi_PID', 'RMSE_phi_PID_neural_u1', ...
    'RMSE_theta_PID', 'RMSE_theta_PID_neural_u1', ...
    'RMSE_psi_PID', 'RMSE_psi_PID_neural_u1', ...
    'Esforco_u1_PID', 'Esforco_u1_PID_neural_u1', ...
    'Melhoria_RMSE_z_percentual', 'Vencedor_altitude'});
writetable(TabelaFiguras,                                     fullfile(outDir,
'Tabela_arquivos_figuras_residual_apenas_u1.xlsx'))

```

```

writetable(TabelaFiguras, fullfile(outDir, 'Tabela_arquivos_figuras_residual_apenas_u1.csv'))
writetable(TabelaMetricas, fullfile(outDir, 'Tabela_metricas_residual_apenas_u1.xlsx'))
writetable(TabelaMetricas, fullfile(outDir, 'Tabela_metricas_residual_apenas_u1.csv'))
disp(TabelaMetricas)
fprintf('\nFiguras e tabelas salvas em: %s\n', outDir)
%%% Funcoes locais
function D = alterar_planta(Dnom, fm, fl, fk)
    D = Dnom;
    D.m = fm*Dnom.m;
    D.Ixx = fl*Dnom.Ixx;
    D.Iyy = fl*Dnom.Iyy;
    D.Izz = fl*Dnom.Izz;
    D.kz = fk*Dnom.kz;
    D.kphi = fk*Dnom.kphi;
    D.ktheta = fk*Dnom.ktheta;
    D.kpsi = fk*Dnom.kpsi;
end
function entrada = montar_entrada_rede(x, R, Dplant, Dnom)
    z = x(1); dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    psi = x(7); r = x(8);
    erros = [R.z-z; -dz; R.phi-phi; -p; R.theta-theta; -q; R.psi-psi; -r];
    fatores = [
        Dplant.m/Dnom.m
        Dplant.Ixx/Dnom.Ixx
        Dplant.Iyy/Dnom.Iyy
        Dplant.Izz/Dnom.Izz
        Dplant.kz/Dnom.kz
        Dplant.kphi/Dnom.kphi
        Dplant.ktheta/Dnom.ktheta
        Dplant.kpsi/Dnom.kpsi
    ];
    entrada = [erros; fatores];

```

end

```
function u = pid_base_local(x, R, D)
```

```
    z = x(1); dz = x(2);
```

```
    phi = x(3); p = x(4);
```

```
    theta = x(5); q = x(6);
```

```
    psi = x(7); r = x(8);
```

```
    ez = R.z - z;
```

```
    ephi = R.phi - phi;
```

```
    etheta = R.theta - theta;
```

```
    epsi = R.psi - psi;
```

```
    Kpz = 10;    Kdz = 6;
```

```
    Kpphi = 0.8; Kdphi = 0.15;
```

```
    Kptheta = 0.8; Kdtheta = 0.15;
```

```
    Kppsi = 1.2; Kdpsi = 0.20;
```

```
    u1 = D.m*(D.g + Kpz*ez - Kdz*dz)/(cos(phi)*cos(theta));
```

```
    u2 = Kpphi*ephi - Kdphi*p;
```

```
    u3 = Kptheta*etheta - Kdtheta*q;
```

```
    u4 = Kppsi*epsi - Kdpsi*r;
```

```
    u = saturar_local([u1 u2 u3 u4], D);
```

end

```
function Res = simular_local(Dctrl, Dplant, R, Ts, Tf, redeResidual, alpha,
    usarRNAapenasU1)
```

```
    t = (0:Ts:Tf)';
```

```
    N = numel(t);
```

```
    X = zeros(N,8);
```

```
    U = zeros(N,4);
```

```
    E = zeros(N,4);
```

```
    X(1,:) = [0 0 deg2rad(8) 0 deg2rad(-5) 0 deg2rad(10) 0];
```

```
    for k = 1:N-1
```

```
        x = X(k,:);
```

```
        uPID = pid_base_local(x, R, Dctrl);
```

```
        u = uPID;
```

```
        if ~isempty(redeResidual) && usarRNAapenasU1
```

```
            entrada = montar_entrada_rede(x, R, Dplant, Dctrl);
```

```

    uCorr = redeResidual(entrada);
    u(1) = uPID(1) + alpha*uCorr(1);
    u(2) = uPID(2);
    u(3) = uPID(3);
    u(4) = uPID(4);
end
u = saturar_local(u, Dctrl);
dx = planta_hover_local(x, u, Dplant);
X(k+1,:) = x + Ts*dx(:)';
U(k,:) = u;
E(k,:) = [R.z-x(1), R.phi-x(3), R.theta-x(5), R.psi-x(7)];
end
U(N,:) = U(N-1,:);
E(N,:) = E(N-1,:);
Res.t = t;
Res.X = X;
Res.U = U;
Res.E = E;
Res.R = R;
Res.Dctrl = Dctrl;
Res.Dplant = Dplant;
end
function dx = planta_hover_local(x, u, D)
    dz = x(2);
    phi = x(3); p = x(4);
    theta = x(5); q = x(6);
    r = x(8);
    u1 = u(1);
    u2 = u(2);
    u3 = u(3);
    u4 = u(4);
    ddz = (u1/D.m)*cos(phi)*cos(theta) - D.g - D.kz*dz;
    dp = u2/D.Ixx - D.kphi*p;
    dq = u3/D.Iyy - D.ktheta*q;

```

```

    dr = u4/D.Izz - D.kpsi*r;
    dx = [dz; ddz; p; dp; q; dq; r; dr];
end
function u = saturar_local(u, D)
    u(1) = min(max(u(1), D.minThrust), D.maxThrust);
    u(2) = min(max(u(2), -D.maxTorque), D.maxTorque);
    u(3) = min(max(u(3), -D.maxTorque), D.maxTorque);
    u(4) = min(max(u(4), -D.maxTorque), D.maxTorque);
end
function met = metricas_locais(Res)
    E = Res.E;
    U = Res.U;
    met.MAE_z = mean(abs(E(:,1)));
    met.RMSE_z = sqrt(mean(E(:,1).^2));
    met.RMSE_phi = sqrt(mean(E(:,2).^2));
    met.RMSE_theta = sqrt(mean(E(:,3).^2));
    met.RMSE_psi = sqrt(mean(E(:,4).^2));
    met.Esf_u1 = mean(abs(U(:,1)));
end
function salvar_figura_variavel(t, yPID, yRNA, ref, ylabelTxt, nomeArquivo)
    f = figure('Color','w','Visible','off');
    plot(t, yPID, 'b', 'LineWidth', 2)
    hold on
    plot(t, yRNA, 'r--', 'LineWidth', 2)
    yline(ref, 'k:', 'LineWidth', 1.5)
    grid on
    xlabel('Tempo (s)')
    ylabel(ylabelTxt)
    legend('PID', 'PID-Neural residual u_1', 'Referencia', 'Location', 'best')
    set(gca, 'FontSize', 12)
    exportgraphics(f, nomeArquivo, 'Resolution', 300)
    close(f)
end
function salvar_figura_controle(t, yPID, yRNA, ylabelTxt, nomeArquivo)

```

```

f = figure('Color','w','Visible','off');
plot(t, yPID, 'b', 'LineWidth', 2)
hold on
plot(t, yRNA, 'r--', 'LineWidth', 2)
grid on
xlabel('Tempo (s)')
ylabel(ylabelTxt)
legend('PID', 'PID-Neural residual u_1', 'Location', 'best')
set(gca, 'FontSize', 12)
exportgraphics(f, nomeArquivo, 'Resolution', 300)
close(f)
end

```

Os scripts `gerar_figuras_resultados_pid_neural_residual.m` e `testar_retreino_rna_residual.m` também foram utilizados em etapas intermediárias de análise, porém a geração final das figuras apresentadas no Capítulo 6 foi realizada a partir do script `gerar_figuras_residual_apenas_u1.m`.

C.3 Scripts De Geração Das Tabelas

Este item apresenta os scripts utilizados para gerar as tabelas de parâmetros físicos, cenários de simulação, configurações de retreinamento da RNA residual e indicadores quantitativos de desempenho dos controladores. Os códigos correspondentes são apresentados a seguir.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% gerar_tabela_variacoes_parametricas.m
% Gera a tabela de variacoes parametricas utilizadas nos testes complementares.
%
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
%   /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:

```

```

% gerar_tabela_variacoes_parametricas

clear
clc
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)
end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
D = parametrosDrone();
cenarios = {
    'massa_menor_20',    'Massa reduzida em 20%',          0.80, 1.00, 1.00
    'massa_maior_20',   'Massa aumentada em 20%',          1.20, 1.00, 1.00
    'inerCIAS_maiores', 'InerCIAS aumentadas em 20%',      1.00, 1.20, 1.00
    'amortecimento_menor', 'Amortecimento reduzido em 30%', 1.00, 1.00, 0.70
    'combinado',        'Massa menor, inerCIAS maiores e amortecimento menor', 0.80, 1.20, 0.70
};
linhas = {};
for i = 1:size(cenarios,1)
    nome = cenarios{i,1};
    descricao = cenarios{i,2};
    fatorMassa = cenarios{i,3};
    fatorInercia = cenarios{i,4};
    fatorAmort = cenarios{i,5};
    massa = fatorMassa*D.m;
    Ixx = fatorInercia*D.Ixx;
    Iyy = fatorInercia*D.Iyy;
    Izz = fatorInercia*D.Izz;
    kz = fatorAmort*D.kz;
    kphi = fatorAmort*D.kphi;
    ktheta = fatorAmort*D.ktheta;

```

```

kpsi = fatorAmort*D.kpsi;
linhas(end+1,:) = { ...
    nome, descricao, ...
    fatorMassa, massa, ...
    fatorInercia, Ixx, Iyy, Izz, ...
    fatorAmort, kz, kphi, ktheta, kpsi};
end
Tabela = cell2table(linhas, 'VariableNames', { ...
    'Cenario', 'Descricao', ...
    'Fator_massa', 'Massa_kg', ...
    'Fator_inercia', 'Ixx_kg_m2', 'Iyy_kg_m2', 'Izz_kg_m2', ...
    'Fator_amortecimento', 'kz', 'kphi', 'ktheta', 'kpsi'});
disp(Tabela)
arquivoXlsx = fullfile(outDir, 'Tabela_variacoes_parametricas_testes_complementares.xlsx');
arquivoCsv = fullfile(outDir, 'Tabela_variacoes_parametricas_testes_complementares.csv');
arquivoTxt = fullfile(outDir,
'Tabela_variacoes_parametricas_testes_complementares_para_Word.txt');
writetable(Tabela, arquivoXlsx)
writetable(Tabela, arquivoCsv)
fid = fopen(arquivoTxt, 'w');
fprintf(fid, 'Tabela 6.X - Variacoes parametricas utilizadas nos testes complementares\n');
fprintf(fid, 'Fonte: Elaborado pelo autor com base nos cenarios de simulacao
implementados.\n\n');
fprintf(fid, 'Cenario\tDescricao\tFator massa\tMassa (kg)\tFator inercia\tIxx (kg.m2)\tIyy
(kg.m2)\tIzz (kg.m2)\tFator amortecimento\tkz\tkphi\tktheta\tkpsi\n');
for i = 1:height(Tabela)
    fprintf(fid, '%s\t%s\t%.2f\t%.4f\t%.2f\t%.6f\t%.6f\t%.2f\t%.4f\t%.4f\t%.4f\n',
...
    Tabela.Cenario{i}, ...
    Tabela.Descricao{i}, ...
    Tabela.Fator_massa(i), ...
    Tabela.Massa_kg(i), ...
    Tabela.Fator_inercia(i), ...
    Tabela.Ixx_kg_m2(i), ...

```

```

Tabela.Iyy_kg_m2(i), ...
Tabela.Izz_kg_m2(i), ...
Tabela.Fator_amortecimento(i), ...
Tabela.kz(i), ...
Tabela.kphi(i), ...
Tabela.ktheta(i), ...
Tabela.kpsi(i));
end
fclose(fid);
fprintf('\nArquivos salvos em:\n')
fprintf('%s\n', arquivoXlsx)
fprintf('%s\n', arquivoCsv)
fprintf('%s\n', arquivoTxt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% gerar_tabela_config_retreino_rna_residual.m
% Gera a tabela de configuracoes utilizadas no retreinamento da RNA residual.
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
%   /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:
%   gerar_tabela_config_retreino_rna_residual
clear
clc
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)
end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
itens = {
    'Tipo de rede',          'Rede neural feedforward para ajuste de funcao'
    'Funcao MATLAB utilizada', 'fitnet'

```

```

'Arquitetura',          'Duas camadas ocultas com 24 e 16 neuronios'
'Algoritmo de treinamento', 'Levenberg-Marquardt (trainlm)'
'Numero de amostras',    '6000'
'Numero de entradas',    '16'
'Numero de saidas',      '4'
'Entradas da rede',      'Erros de altitude e atitude, velocidades associadas e fatores
relativos da planta'
'Saidas da rede',        'Correcoes residuais para u1, u2, u3 e u4'
'Divisao dos dados',     '70% treinamento, 15% validacao e 15% teste'
'Numero maximo de epocas', '600'
'Meta de desempenho',    '1e-7'
'Fator de ponderacao alpha', '1,0'
'Lei de controle avaliada', 'u = u_PID + alpha*u_RNA,res'
'Cenarios considerados',  'Variacoes de massa, inercias, amortecimento e cenario
combinado'
'Arquivo da rede treinada', 'rede_residual_pid_neural.mat'
};
Tabela = cell2table(itens, 'VariableNames', {'Parametro','Valor'});
disp(Tabela)
arquivoXlsx = fullfile(outDir, 'Tabela_configuracoes_retreino_RNA_residual.xlsx');
arquivoCsv = fullfile(outDir, 'Tabela_configuracoes_retreino_RNA_residual.csv');
arquivoTxt = fullfile(outDir, 'Tabela_configuracoes_retreino_RNA_residual_para_Word.txt');
writetable(Tabela, arquivoXlsx)
writetable(Tabela, arquivoCsv)
fid = fopen(arquivoTxt, 'w');
fprintf(fid, 'Tabela 6.X - Configuracoes utilizadas no retreinamento da RNA residual\n');
fprintf(fid, 'Fonte: Elaborado pelo autor com base na implementacao computacional do
retreinamento da RNA residual.\n\n');
fprintf(fid, 'Parametro\tValor\n');
for i = 1:height(Tabela)
    fprintf(fid, '%s\t%s\n', Tabela.Parametro{i}, Tabela.Valor{i});
end
fclose(fid);
fprintf('\nArquivos salvos em:\n')

```

```

fprintf('%s\n', arquivoXlsx)
fprintf('%s\n', arquivoCsv)
fprintf('%s\n', arquivoTxt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% gerar_tabela_cenarios_pid_neural_residual.m
% Gera a tabela de cenarios avaliados para o controlador PID-Neural residual.
% Como usar no MATLAB Online:
% 1) Salve este arquivo em:
%   /MATLAB Drive/ARDrone_Simulacao_PID_Neural
% 2) Execute:
%   gerar_tabela_cenarios_pid_neural_residual
clear
clc
projDir = '/MATLAB Drive/ARDrone_Simulacao_PID_Neural';
if exist(projDir, 'dir') == 7
    cd(projDir)
end
outDir = fullfile(pwd, 'resultados_retreino_rna_residual');
if exist(outDir, 'dir') ~= 7
    mkdir(outDir)
end
dados = {
    'C01', 'massa_menor_20',    'Massa reduzida em 20%',          'm = 0,80 m_n',
    'Avaliar robustez frente a reducao de massa'
    'C02', 'massa_maior_20',    'Massa aumentada em 20%',          'm = 1,20 m_n',
    'Avaliar robustez frente a aumento de massa'
    'C03', 'inercias_maiores',   'Inercias aumentadas em 20%',      'Ixx, Iyy, Izz =
1,20 I_n', 'Avaliar resposta angular com maior resistencia rotacional'
    'C04', 'amortecimento_menor', 'Amortecimento reduzido em 30%',    'k = 0,70
k_n',      'Avaliar desempenho com menor dissipacao dinamica'
    'C05', 'combinado',          'Massa menor, inercias maiores e amortecimento menor', 'm =
0,80 m_n; I = 1,20 I_n; k = 0,70 k_n', 'Avaliar efeito simultaneo das variacoes parametricas'
    };
Tabela = cell2table(dados, 'VariableNames', { ...

```

```

    'Cenario', 'Identificacao', 'Descricao', 'Variacao_parametrica', 'Objetivo'}));
disp(Tabela)
arquivoXlsx = fullfile(outDir, 'Tabela_cenarios_PID_neural_residual.xlsx');
arquivoCsv = fullfile(outDir, 'Tabela_cenarios_PID_neural_residual.csv');
arquivoTxt = fullfile(outDir, 'Tabela_cenarios_PID_neural_residual_para_Word.txt');
writetable(Tabela, arquivoXlsx)
writetable(Tabela, arquivoCsv)
fid = fopen(arquivoTxt, 'w');
fprintf(fid, 'Tabela 6.X - Cenarios avaliados para o controlador PID-Neural residual\n');
fprintf(fid, 'Fonte: Elaborado pelo autor com base nos cenarios de simulacao
implementados.\n\n');
fprintf(fid, 'Cenario\tIdentificacao\tDescricao\tVariacao parametrica\tObjetivo\n');
for i = 1:height(Tabela)
    fprintf(fid, '%s\t%s\t%s\t%s\t%s\n', ...
        Tabela.Cenario{i}, ...
        Tabela.Identificacao{i}, ...
        Tabela.Descricao{i}, ...
        Tabela.Variacao_parametrica{i}, ...
        Tabela.Objetivo{i});
end
fclose(fid);
fprintf('\nArquivos salvos em:\n')
fprintf('%s\n', arquivoXlsx)
fprintf('%s\n', arquivoCsv)
fprintf('%s\n', arquivoTxt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%gerar_tabela13_indicadores_pid_neural_u1
clear
clc
% Diretório principal do projeto
cd('/MATLAB Drive/ARDrone_Simulacao_PID_Neural')
% Pasta de saída
pastaSaida = fullfile(pwd, 'resultados_retreino_rna_residual', ...
    'figuras_residual_apenas_u1');

```

```

if ~exist(pastaSaida, 'dir')
    mkdir(pastaSaida)
end
% Tabela 13 - Indicadores quantitativos finais
Cenario = {
    'C01'
    'C02'
    'C03'
    'C04'
    'C05'
};
Descricao = {
    'Massa reduzida em 20%'
    'Massa aumentada em 20%'
    'Inercias aumentadas em 20%'
    'Amortecimento reduzido em 30%'
    'Massa menor, inercias maiores e amortecimento menor'
};
RMSE_z_PID = [
    0.22580
    0.24860
    0.14073
    0.14025
    0.22574
];
RMSE_z_PID_neural_u1 = [
    0.13907
    0.13913
    0.13908
    0.13907
    0.13910
];
MAE_z_PID = [
    0.20484

```

```

0.22151
0.031234
0.030860
0.20489
];
MAE_z_PID_neural_u1 = [
0.030129
0.030178
0.030333
0.030162
0.030200
];
RMSE_phi_PID = [
0.010081
0.010081
0.010158
0.010080
0.010156
];
RMSE_phi_PID_neural_u1 = RMSE_phi_PID;
RMSE_theta_PID = [
0.0063798
0.0063798
0.0064427
0.0063786
0.0064413
];
RMSE_theta_PID_neural_u1 = RMSE_theta_PID;
RMSE_psi_PID = [
0.012262
0.012262
0.012420
0.012260
0.012418

```

```

];
RMSE_psi_PID_neural_u1 = RMSE_psi_PID;
Esforco_u1_PID = [
    3.3016
    4.9498
    4.1259
    4.1244
    3.3001
];
Esforco_u1_PID_neural_u1 = [
    3.3008
    4.9511
    4.1259
    4.1244
    3.2995
];
Melhoria_RMSE_z_percentual = [
    38.410
    44.033
    1.1737
    0.83973
    38.384
];
Vencedor = {
    'PID-Neural residual u1'
    'PID-Neural residual u1'
    'PID-Neural residual u1'
    'PID-Neural residual u1'
    'PID-Neural residual u1'
};
Tabela13 = table( ...
    Cenario, ...
    Descricao, ...
    RMSE_z_PID, ...

```

```

RMSE_z_PID_neural_u1, ...
MAE_z_PID, ...
MAE_z_PID_neural_u1, ...
RMSE_phi_PID, ...
RMSE_phi_PID_neural_u1, ...
RMSE_theta_PID, ...
RMSE_theta_PID_neural_u1, ...
RMSE_psi_PID, ...
RMSE_psi_PID_neural_u1, ...
Esforco_u1_PID, ...
Esforco_u1_PID_neural_u1, ...
Melhoria_RMSE_z_percentual, ...
Vencedor);
% Exibir no MATLAB
disp(Tabela13)
% Salvar arquivos
arquivoXLSX = fullfile(pastaSaida, ...
    'Tabela13_indicadores_PID_PID_neural_residual_u1.xlsx');
arquivoCSV = fullfile(pastaSaida, ...
    'Tabela13_indicadores_PID_PID_neural_residual_u1.csv');
arquivoTXT = fullfile(pastaSaida, ...
    'Tabela13_indicadores_PID_PID_neural_residual_u1_para_Word.txt');
writetable(Tabela13, arquivoXLSX)
writetable(Tabela13, arquivoCSV)
% Versão em texto para copiar no Word
fid = fopen(arquivoTXT, 'w');
fprintf(fid, 'Tabela 13 - Indicadores quantitativos do PID e do PID-Neural residual aplicado ao
canal de empuxo.\n\n');
fprintf(fid, ['Cenario\tDescricao\tRMSE z PID\tRMSE z PID-Neural residual u1\t' ...
    'MAE z PID\tMAE z PID-Neural residual u1\tRMSE phi PID\t' ...
    'RMSE phi PID-Neural residual u1\tRMSE theta PID\t' ...
    'RMSE theta PID-Neural residual u1\tRMSE psi PID\t' ...
    'RMSE psi PID-Neural residual u1\tEsforco u1 PID\t' ...
    'Esforco u1 PID-Neural residual u1\tMelhoria RMSE z (%%)\tVencedor\n']);

```

```

for i = 1:height(Tabela13)
    fprintf(fid,
'%s\t%s\t%.5ft%.5ft%.5ft%.5ft%.6ft%.6ft%.7ft%.7ft%.6ft%.6ft%.4ft%.4ft%.3ft%s\
n', ...
        Tabela13.Cenario{i}, ...
        Tabela13.Descricao{i}, ...
        Tabela13.RMSE_z_PID(i), ...
        Tabela13.RMSE_z_PID_neural_u1(i), ...
        Tabela13.MAE_z_PID(i), ...
        Tabela13.MAE_z_PID_neural_u1(i), ...
        Tabela13.RMSE_phi_PID(i), ...
        Tabela13.RMSE_phi_PID_neural_u1(i), ...
        Tabela13.RMSE_theta_PID(i), ...
        Tabela13.RMSE_theta_PID_neural_u1(i), ...
        Tabela13.RMSE_psi_PID(i), ...
        Tabela13.RMSE_psi_PID_neural_u1(i), ...
        Tabela13.Esforco_u1_PID(i), ...
        Tabela13.Esforco_u1_PID_neural_u1(i), ...
        Tabela13.Melhoria_RMSE_z_percentual(i), ...
        Tabela13.Vencedor{i});
end
fprintf(fid, '\nFonte: Elaborado pelo autor com base nos dados obtidos nas simulacoes.\n');
fclose(fid);
fprintf('\nArquivos salvos em:\n')
fprintf('%s\n', arquivoXLSX)
fprintf('%s\n', arquivoCSV)
fprintf('%s\n', arquivoTXT)

```

C.4 Cálculo das métricas quantitativas

Este item apresenta os indicadores utilizados para avaliar quantitativamente o desempenho dos controladores. As métricas principais foram o erro absoluto médio, o erro quadrático médio, a integral do erro quadrático e o esforço médio de empuxo. Esses indicadores permitiram comparar o desempenho do controlador PID e do controlador PID-

Neural residual aplicado ao canal de empuxo u_1 . As equações correspondentes são apresentadas a seguir.

$$MAE = \frac{1}{N} \sum_{k=1}^N |e(k)| \quad (91)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N e(k)^2} \quad (92)$$

$$ISE = \sum_{k=1}^N e(k)^2 T_s \quad (93)$$

em que $e(k)$ representa o erro no instante de amostragem k , N é o número total de amostras, T_s é o tempo de amostragem e $u_1(k)$ representa o sinal de empuxo aplicado ao quadrotor.

APÊNDICE D — RESULTADOS COMPLEMENTARES

Este apêndice apresenta resultados complementares obtidos durante o desenvolvimento da estratégia PID-Neural. Embora esses resultados não componham a análise principal do Capítulo 6, eles são relevantes para demonstrar o processo de avaliação, ajuste e escolha da formulação final do controlador. Assim, são incluídos testes intermediários envolvendo a comparação inicial entre PID e PID-Neural, a aplicação da RNA residual em todos os canais de controle e a formulação final com atuação neural restrita ao canal de empuxo u_1 .

D.1 Comparação Inicial Entre PID E PID-Neural

Este item apresenta os resultados da comparação inicial entre o controlador PID e o controlador PID-Neural originalmente implementado. Essa etapa mostrou que os dois controladores apresentaram respostas praticamente equivalentes, indicando que a configuração inicial da rede neural não produziu melhoria significativa em relação à estratégia PID utilizada como referência.

Tabela D.1 – Indicadores da comparação inicial entre PID e PID-Neural.

Estado inicial	Métrica	PID	PID-Neural
Altitude z	MAE	0,030741	0,030741
Altitude z	RMSE	0,13171	0,13171
Altitude z	ISE	0,34693	0,34693
Rolagem ϕ	RMSE	0,010084	0,010084
Arfagem θ	RMSE	0,0063814	0,0063814
Guinada ψ	RMSE	0,012265	0,012265
Empuxo u_1	Valor médio absoluto	4,126	4,126

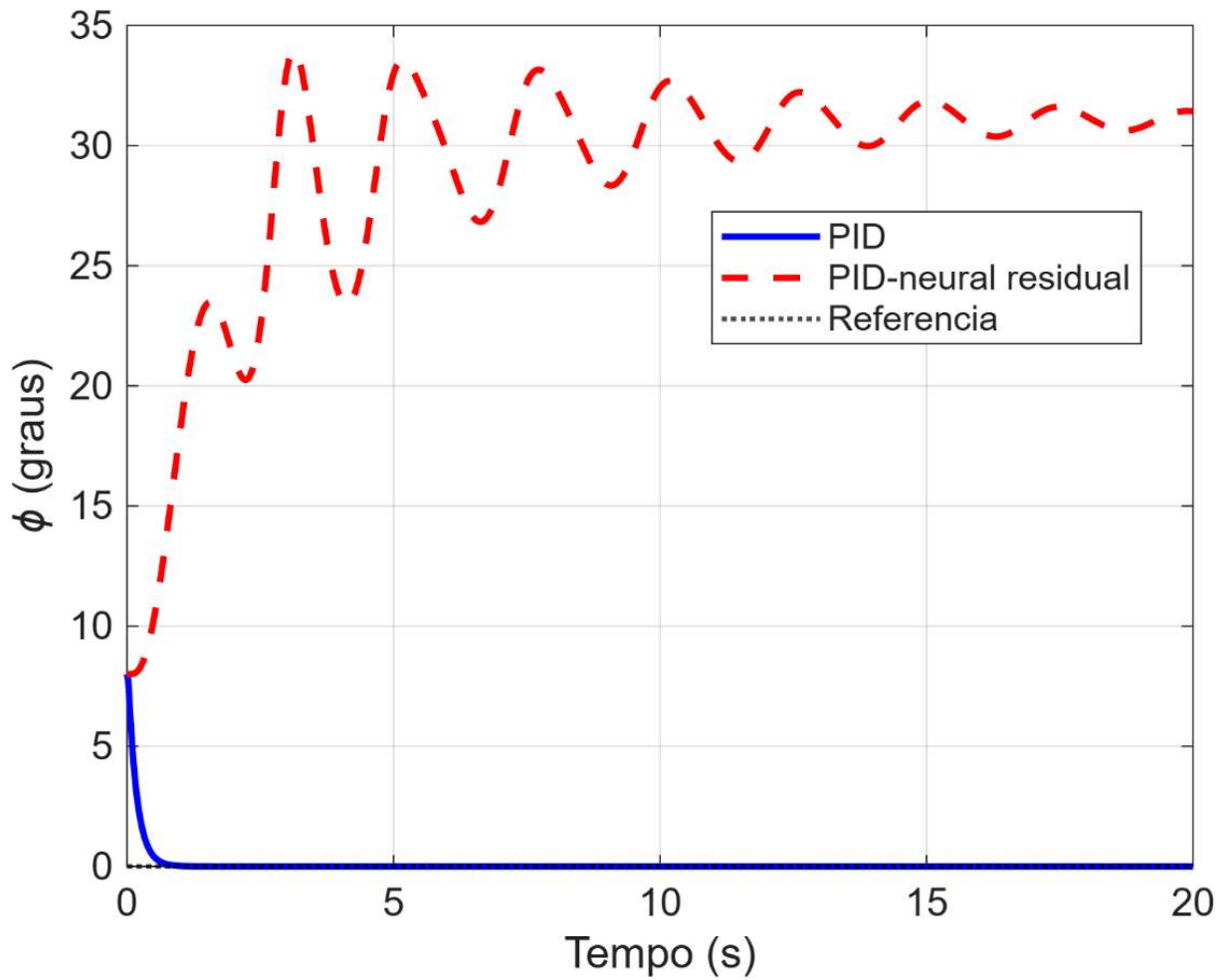
Fonte: Elaborado pelo autor (2014).

D.2 PID-Neural Residual Aplicado A Todos Os Canais

Este item apresenta os resultados obtidos quando a correção neural residual foi aplicada simultaneamente aos canais de controle u_1 , u_2 , u_3 e u_4 . Essa configuração apresentou

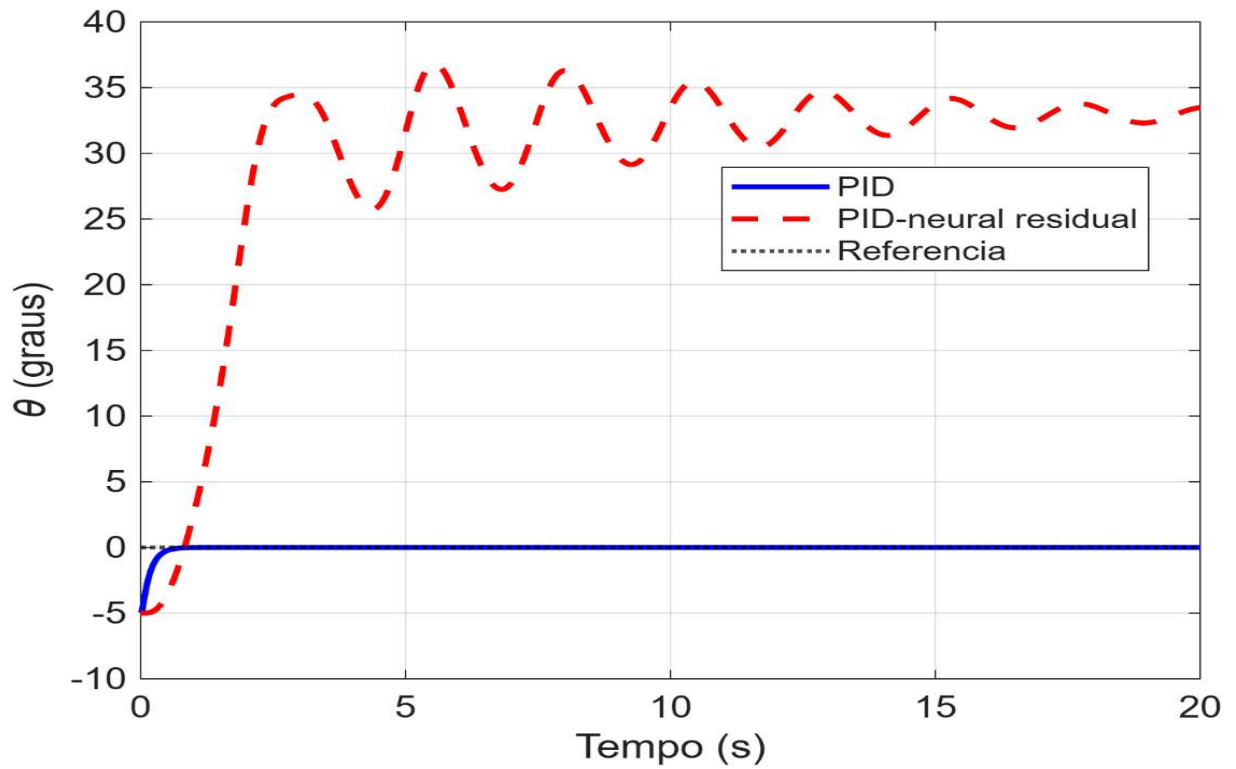
melhoria na resposta de altitude; entretanto, provocou degradação nas respostas angulares, especialmente nos ângulos de rolagem, arfagem e guinada. Por esse motivo, essa formulação foi mantida apenas como resultado complementar e não foi adotada como configuração final do controlador.

Figura D.1 – Resposta de rolagem com PID-Neural residual aplicado a todos os canais.



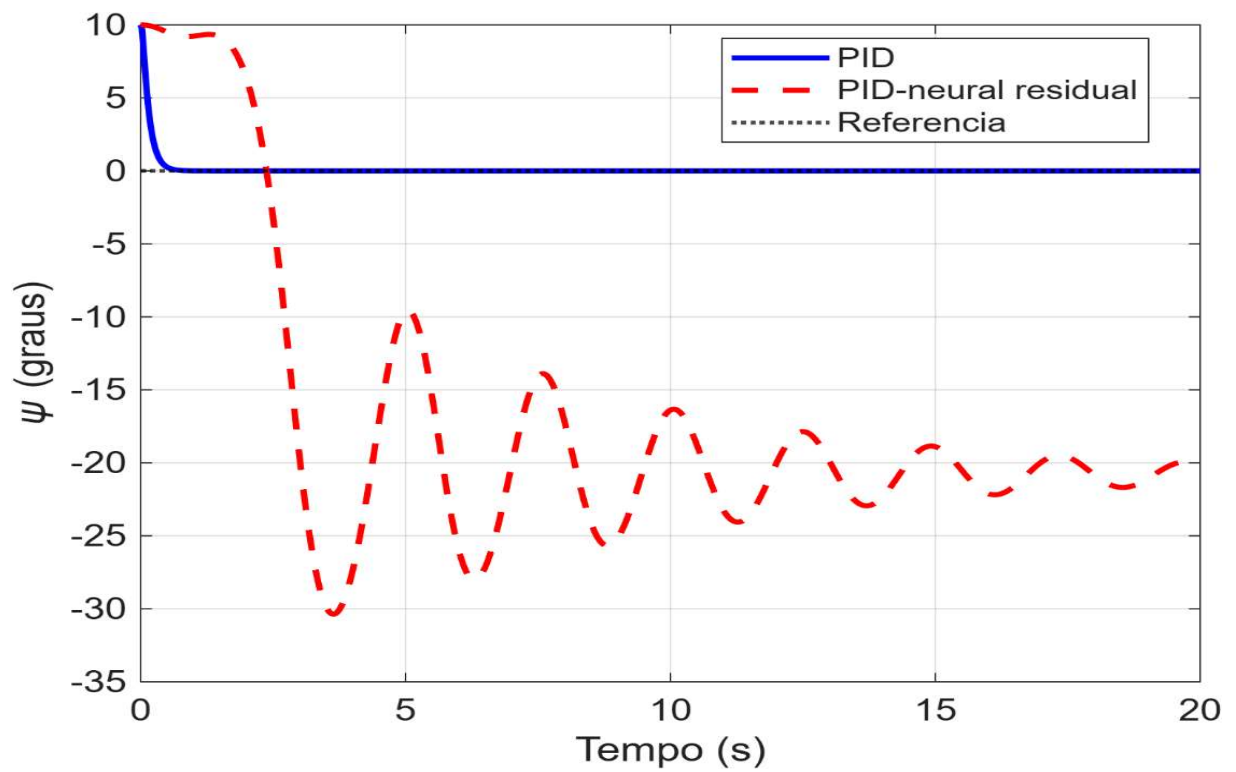
Fonte: Elaborado pelo autor (2014).

Figura D.2 – Resposta de arfagem com PID-Neural residual aplicado a todos os canais.



Fonte: Elaborado pelo autor (2014).

Figura D.3 – Resposta de arfagem com PID-Neural residual aplicado a todos os canais.



Fonte: Elaborado pelo autor (2014).

As Figuras D.2 a D.4 mostram que a aplicação da correção neural residual em todos os canais de controle provocou desvios significativos nas respostas angulares. Esse comportamento indicou que a atuação da RNA sobre os torques u_2 , u_3 e u_4 poderia comprometer a estabilização de atitude. Por esse motivo, a formulação final restringiu a contribuição neural ao canal de empuxo u_1 .

D.3 PID-Neural Residual Aplicado Apenas Ao Canal De Empuxo

Este item apresenta os resultados complementares da formulação final do controlador PID-Neural residual, na qual a rede neural atua apenas no canal de empuxo u_1 . Essa configuração apresentou melhor equilíbrio entre a redução do erro de altitude, a preservação da estabilidade de atitude e a manutenção do esforço de controle em níveis próximos aos observados com o controlador PID.

Figura D.4 – Resposta de altitude com PID-Neural residual aplicado ao empuxo no cenário de inércias aumentadas.

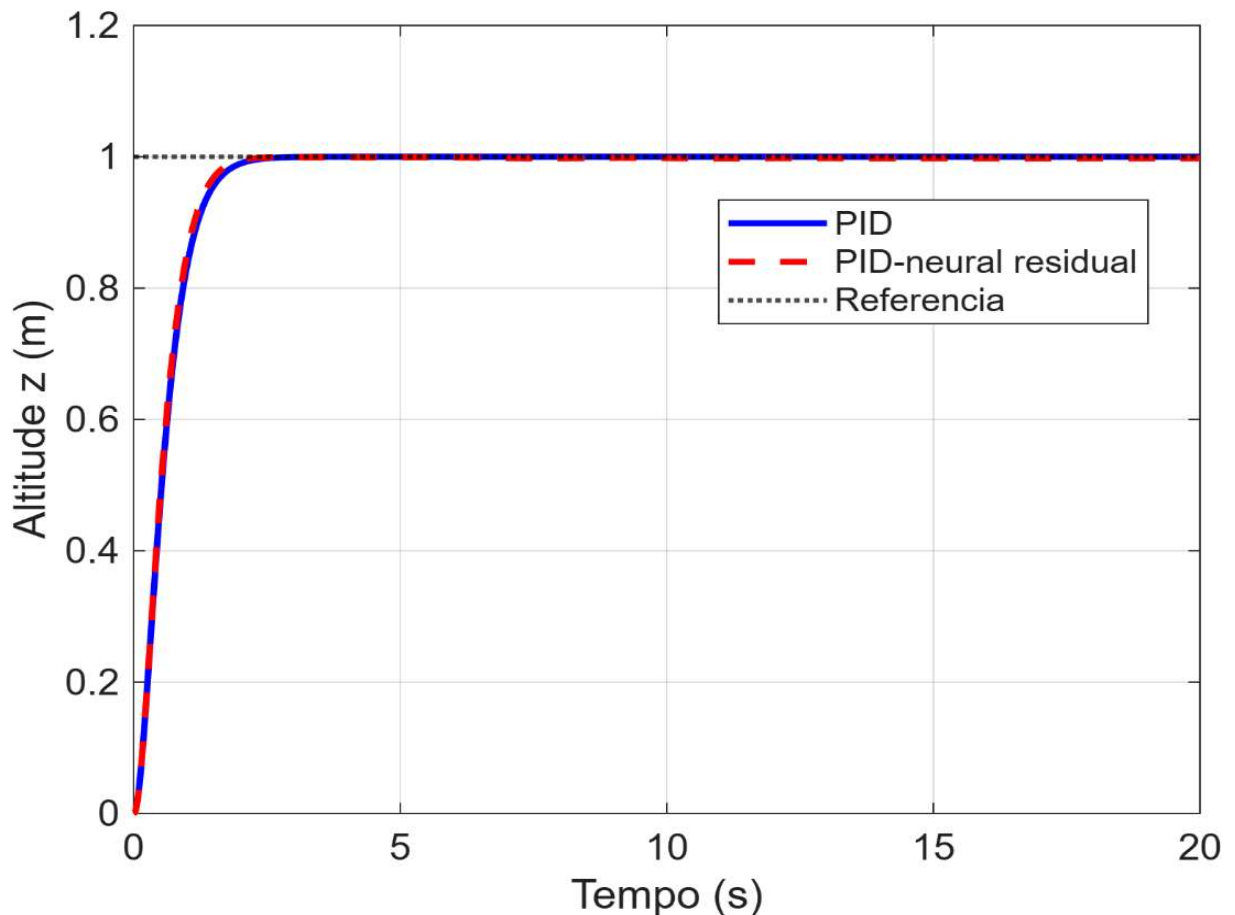
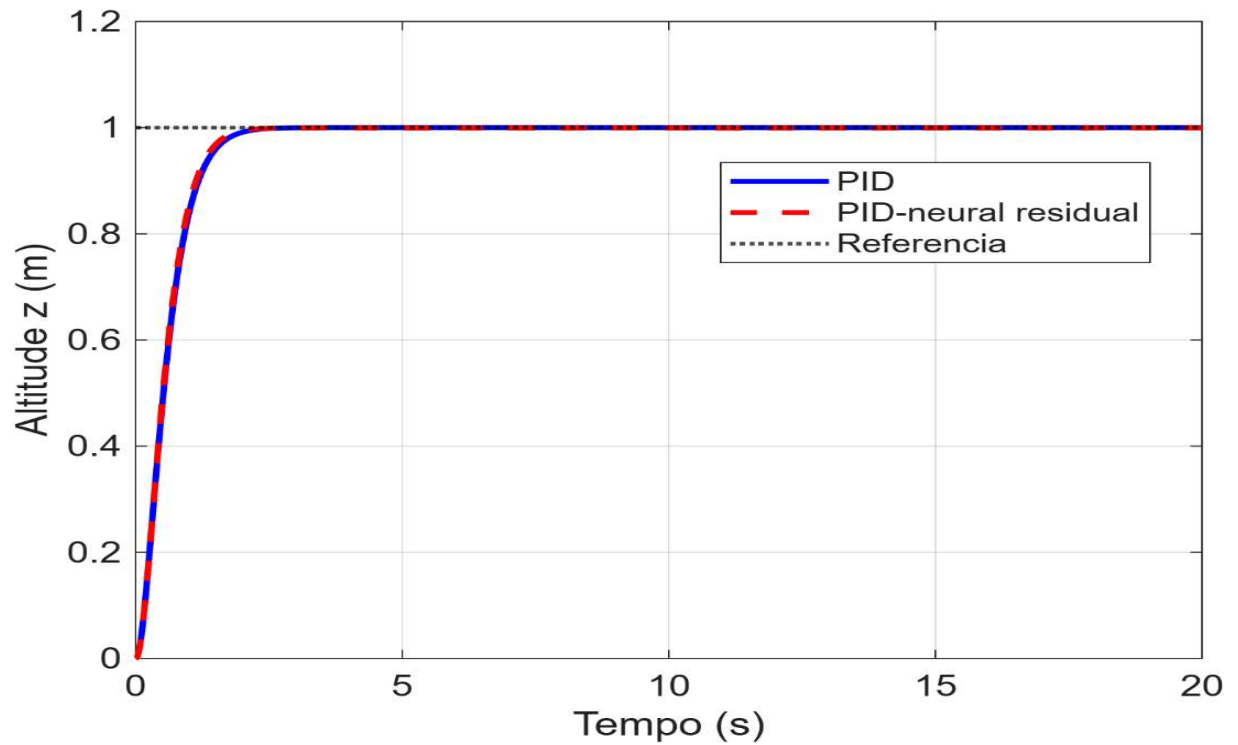
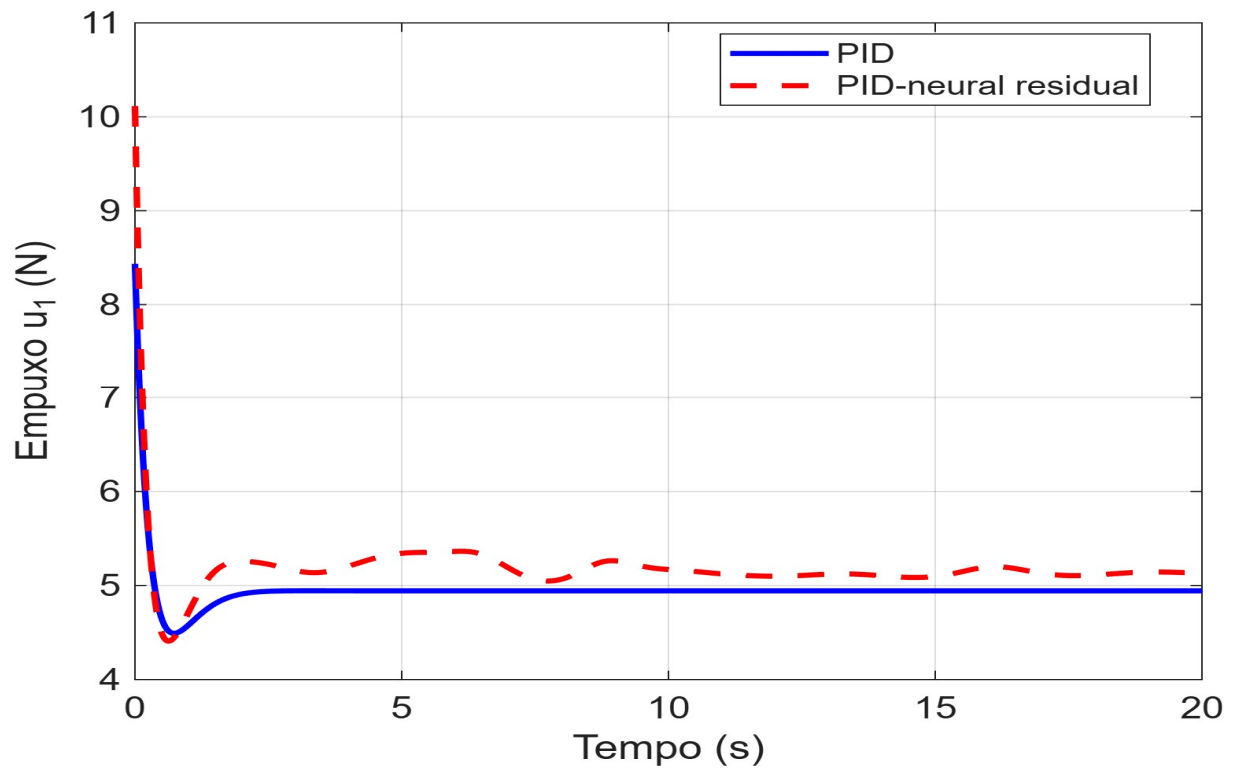


Figura D.5 – Resposta de altitude com PID-Neural residual aplicado ao empuxo no cenário de amortecimento reduzido.



Fonte: Elaborado pelo autor (2014).

Figura D.6 – Sinal de empuxo com PID-Neural residual aplicado ao empuxo no cenário de massa aumentada em 20%.



Fonte: Elaborado pelo autor (2014).

Os resultados principais dessa formulação foram apresentados no Capítulo 6. Neste apêndice, são incluídos apenas resultados complementares, com o objetivo de documentar a análise completa realizada durante a avaliação da estratégia proposta.

D.4 Tabelas Complementares de resultados

Este item apresenta tabelas complementares contendo as métricas completas geradas no *MATLAB/Simulink*. São incluídos indicadores que foram resumidos no corpo da dissertação para melhorar a legibilidade, mas que são relevantes para documentar de forma mais abrangente os resultados obtidos nas simulações.

Tabela D.2 – Indicadores completos do PID-Neural residual aplicado ao canal de empuxo

Cenário	Descrição	RMSE z PID	RMSE z PID- Neural residual u_1	MAE z PID	MAE z PID- Neural residual u_1	RMSE ϕ PID	RMSE ϕ PID- Neural residual u_1	RMSE θ PID	RMSE θ PID- Neural residual u_1	RMSE ψ PID	RMSE ψ PID- Neural residual u_1	Esforço u_1 PID	Esforço u_1 PID- Neural residual u_1	Melhoria RMSE z (%)	Vencedor
C01	Massa reduzida em 20%	0,22580	0,13907	0,20484	0,03013	0,010081	0,010081	0,0063798	0,0063798	0,012262	0,012262	3,3016	3,3008	38,410	PID- Neural residual u_1
C02	Massa aumentada em 20%	0,24860	0,13913	0,22151	0,03018	0,010081	0,010081	0,0063798	0,0063798	0,012262	0,012262	4,9498	4,9511	44,033	PID- Neural residual u_1
C03	Inércias aumentadas em 20%	0,14073	0,13908	0,03123	0,03033	0,010158	0,010158	0,0064427	0,0064427	0,012420	0,012420	4,1259	4,1259	1,174	PID- Neural residual u_1
C04	Amortecimento reduzido em 30%	0,14025	0,13907	0,03086	0,03016	0,010080	0,010080	0,0063786	0,0063786	0,012260	0,012260	4,1244	4,1244	0,840	PID- Neural residual u_1
C05	Massa menor, inércias maiores e amortecimento menor	0,22574	0,13910	0,20489	0,03020	0,010156	0,010156	0,0064413	0,0064413	0,012418	0,012418	3,3001	3,2995	38,384	PID- Neural residual u_1

Fonte: Elaborado pelo autor (2014).

Tabela D.3 – Cenários avaliados para o controlador PID-Neural residual

Cenário	Identificação	Descrição	Variação paramétrica	Objetivo
C01	massa_menor_20	Massa reduzida em 20%	$m = 0,80m_n$	Avaliar robustez frente à redução de massa
C02	massa_maior_20	Massa aumentada em 20%	$m = 1,20m_n$	Avaliar robustez frente ao aumento de massa
C03	inercias_maiores	Inércias aumentadas em 20%	$I_{xx}, I_{yy}, I_{zz} = 1,20I_n$	Avaliar resposta angular com maior resistência rotacional
C04	amortecimento_menor	Amortecimento reduzido em 30%	$k = 0,70k_n$	Avaliar desempenho com menor dissipação dinâmica
C05	Combinado	Massa menor, inércias maiores e amortecimento menor	$m = 0,80m_n;$ $I = 1,20I_n;$ $k = 0,70k_n$	Avaliar efeito simultâneo das variações paramétricas

Fonte: Elaborado pelo autor (2014).

Tabela D.4 – Variações paramétricas utilizadas nos testes complementares

Cenário	Descrição	Fator de massa	Massa (kg)	Fator de inércia	I_{xx} (kg.m ²)	I_{yy} (kg.m ²)	I_{zz} (kg.m ²)	Fator de amortecimento	k_z	k_ϕ	k_θ	k_ψ
massa_menor_20	Massa reduzida em 20%	0,8	0,336	1,0	0,002240	0,002980	0,004800	1,0	0,250	0,080	0,080	0,050
massa_maior_20	Massa aumentada em 20%	1,2	0,504	1,0	0,002240	0,002980	0,004800	1,0	0,250	0,080	0,080	0,050
inercias_maiores	Inércias aumentadas em 20%	1,0	0,420	1,2	0,002688	0,003576	0,005760	1,0	0,250	0,080	0,080	0,050
amortecimento_menor	Amortecimento reduzido em 30%	1,0	0,420	1,0	0,002240	0,002980	0,004800	0,7	0,175	0,056	0,056	0,035
combinado	Massa menor, inércias maiores e amortecimento menor	0,8	0,336	1,2	0,002688	0,003576	0,005760	0,7	0,175	0,056	0,056	0,035

Fonte: Elaborado pelo autor (2014).



Universidade Federal Rural do Semi-Árido (UFERSA)
Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG)
6ª Reunião Ordinária de 2026 do CPPGIT

4º PONTO

Apreciação e deliberação sobre a pauta da 6ª Reunião Ordinária do Consepe.

Segue link para acessar os documentos da reunião do Consepe:

<https://conselhos.ufersa.edu.br/convocacoes-pastas-e-atas-consepe-2026/>



Universidade Federal Rural do Semi-Árido (UFERSA)
Pró-Reitoria de Pesquisa e Pós-Graduação (PROPPG)
6ª Reunião Ordinária de 2026 do CPPGIT

5º PONTO

Outras ocorrências.