



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/01.02.18.25-TDI

SIMULAÇÃO DO COMPORTAMENTO TÉRMICO DE SATÉLITES EM TEMPO REAL COM USO DE REDES NEURAIS ARTIFICIAIS

José Daniel Reis Júnior

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelos Drs. Fabiano Luis de Sousa, e Ana Maria Ambrosio, aprovada em 18 de dezembro de 2018.

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34R/3SFU6RH>

INPE
São José dos Campos
2019

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GBDIR)

Serviço de Informação e Documentação (SESID)

CEP 12.227-010

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/7348

E-mail: pubtc@inpe.br

COMISSÃO DO CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELECTUAL DO INPE (DE/DIR-544):

Presidente:

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CGCPT)

Membros:

Dra. Carina Barros Mello - Coordenação de Laboratórios Associados (COCTE)

Dr. Alisson Dal Lago - Coordenação-Geral de Ciências Espaciais e Atmosféricas (CGCEA)

Dr. Evandro Albiach Branco - Centro de Ciência do Sistema Terrestre (COCST)

Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia e Tecnologia Espacial (CGETE)

Dr. Hermann Johann Heinrich Kux - Coordenação-Geral de Observação da Terra (CGOBT)

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação - (CPG)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SESID)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SESID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SESID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SESID)

EDITORAÇÃO ELETRÔNICA:

Ivone Martins - Serviço de Informação e Documentação (SESID)

Murilo Luiz Silva Gino - Serviço de Informação e Documentação (SESID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21c/2019/01.02.18.25-TDI

SIMULAÇÃO DO COMPORTAMENTO TÉRMICO DE SATÉLITES EM TEMPO REAL COM USO DE REDES NEURAIS ARTIFICIAIS

José Daniel Reis Júnior

Tese de Doutorado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais, orientada pelos Drs. Fabiano Luis de Sousa, e Ana Maria Ambrosio, aprovada em 18 de dezembro de 2018.

URL do documento original:

<http://urlib.net/8JMKD3MGP3W34R/3SFU6RH>

INPE
São José dos Campos
2019

Dados Internacionais de Catalogação na Publicação (CIP)

Reis Júnior, José Daniel.

R277s Simulação do comportamento térmico de satélites em tempo real com uso de redes neurais artificiais / José Daniel Reis Júnior. – São José dos Campos : INPE, 2019.
xxiv + 141 p. ; (sid.inpe.br/mtc-m21c/2019/01.02.18.25-TDI)

Tese (Doutorado em Engenharia e Tecnologia Espaciais/Engenharia e Gerenciamento de Sistemas Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2018.

Orientadores : Drs. Fabiano Luis de Sousa, e Ana Maria Ambrosio.

1. Simulador operacional de satélites. 2. Simulação térmica. 3. Simulação em tempo real. 4. Redes neurais artificiais. I.Título.

CDU 629.78:536



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aluno (a): **José Daniel Reis Júnior**

Título: "SIMULAÇÃO DO COMPORTAMENTO TÉRMICO DE SATÉLITES EM TEMPO REAL COM USO DE REDES NEURAIS ARTIFICIAIS"

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de **Doutor(a)** em **Engenharia e Tecnologia Espaciais/Eng. Gerenc. de Sistemas Espaciais**

Dr. **Maurício Gonçalves Vieira Ferreira**

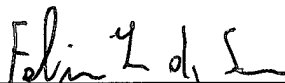


Presidente / INPE / SJCampos - SP

() **Participação por Video - Conferência**

Aprovado () **Reprovado**

Dr. **Fabiano Luis de Sousa**

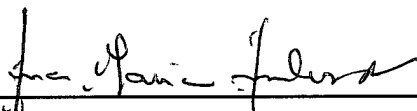


Orientador(a) / INPE / SJCampos - SP

() **Participação por Video - Conferência**

Aprovado () **Reprovado**

Dra. **Ana Maria Ambrosio**



Orientador(a) / INPE / São José dos Campos - SP

() **Participação por Video - Conferência**

Aprovado () **Reprovado**

Dr. **Fernando Manuel Ramos**



Membro da Banca // INPE / SJCampos - SP

() **Participação por Video - Conferência**

Aprovado () **Reprovado**

Este trabalho foi aprovado por:

() **maioria simples**

unanimidade

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Doutor(a)** em
**Engenharia e Tecnologia Espaciais/Eng.
Gerenc. de Sistemas Espaciais**

Dr. Marcos Gonçalves Quiles



Membro da Banca / INPE / São José dos Campos - SP

() Participação por Vídeo - Conferência

Aprovado () Reprovado

Dr. Ezio Castejon Garcia



Convidado(a) / ITA / São José dos Campos - SP

() Participação por Vídeo - Conferência

Aprovado () Reprovado

Dr. Paulo Marcelo Tasinaffo



Convidado(a) / ITA / São José dos Campos - SP

() Participação por Vídeo - Conferência

Aprovado () Reprovado

Este trabalho foi aprovado por:

() maioria simples

unanimidade

*“Caminhante, são tuas pegadas
o caminho e nada mais;
caminhante, não há caminho,
se faz caminho ao andar”.*

Trecho traduzido do poema Cantares de Antonio Machado

À minha esposa Luciana e aos amigos Nélia e Bogossian.

AGRADECIMENTOS

Agradeço, primeiramente, aos meus orientadores Ana Maria Ambrosio e Fabiano Luis de Sousa pelos ensinamentos, dedicação e generosidade.

Agradeço aos professores Mauricio Gonçalves Vieira Ferreira, Fernando Manuel Ramos, Marcos Gonçalves Quiles, Paulo Marcelo Tasinaffo e Ezio Castejon Garcia, pelos conhecimentos compartilhados e contribuições para melhoria do trabalho.

Agradeço aos amigos Nélia Ferreira Leite (*in memoriam*), Otávio Luiz Bogossian (*in memoriam*), Guilherme Sydow Nunes Bueno Brandão, Júlio Cesar Neves Juncioni, Mônica Elizabeth Rocha de Oliveira, Marcela Soares Servo e outros tantos que apoiaram, incentivaram e contribuíram para que este trabalho fosse realizado.

Por fim, agradeço à minha esposa pelo apoio e compreensão, e aos meus pais pela educação e incentivo.

RESUMO

Simuladores Operacionais de Satélites Artificiais são sistemas em software utilizados principalmente para apoiar as atividades de operação, treinar os operadores, testar o sistema de controle, além de permitir avaliar procedimentos operacionais normais e adversos, antes da execução destes no satélite real. Para atingir esses requisitos, tais Simuladores devem ser executados em tempo real, gerando respostas no mesmo tempo e com o mesmo conteúdo, como se fossem os próprios satélites em órbita da Terra. Dentre os diversos modelos que compõem um Simulador Operacional, o modelo térmico é um dos que apresenta maior custo computacional, pois a abordagem tradicional de análise térmica requer a solução de uma grande quantidade de equações diferenciais para determinar a distribuição de temperaturas no interior do satélite em função do ambiente espacial e do estado de funcionamento dos equipamentos que compõem o satélite. Com o objetivo de simular o comportamento térmico de satélites em tempo real, gerando valores de temperatura reais, foi proposta a utilização de Redes Neurais Artificiais, devido à sua faculdade de aprender relações entre conjuntos de dados de entrada e saída, e de generalizar a partir do conhecimento adquirido. Ao longo da pesquisa, foi investigada a aplicação de diferentes arquiteturas e algoritmos de treinamento de redes neurais e foram avaliados os efeitos de escala no sistema estudado. Os resultados, obtidos tanto para um CubeSat como para um satélite de médio porte, o Amazonia-1, evidenciaram o potencial e a adequação do uso de Redes Neurais para simular o comportamento térmico de satélites em tempo real com elevada fidelidade, corroborando a viabilidade da sua utilização como modelo térmico para um Simulador Operacional de Satélites.

Palavras-chave: Simulador Operacional de Satélites. Simulação Térmica. Simulação em Tempo Real. Redes Neurais Artificiais.

SIMULATION OF SATELLITE THERMAL BEHAVIOR IN REAL TIME WITH USE OF ARTIFICIAL NEURAL NETWORKS

ABSTRACT

Artificial Satellite Operational Simulators are software systems used primarily to support operating activities, train operators, test the control system, and allow the assessment of normal and adverse operational procedures prior to the execution of these on the actual satellite. To achieve these requirements, such simulators must run in real time, generating responses at the same time and with the same content, as if they were the real satellite in Earth's orbit. Among the several models that compose an Operational Simulator, the thermal model is one of those that presents the higher computational cost, since the traditional approach of thermal analysis requires the solution of a great quantity of differential equations to determine the distribution of temperatures inside the satellite in function of the space environment and the state of operation of the equipment that compose the satellite. In order to simulate the thermal behavior of satellites in real time, generating real temperature values, it was proposed the use of Artificial Neural Networks, due to its ability to learn relationships between input and output data sets, and to generalize from the acquired knowledge. Throughout the research, the application of different architectures and neural network training algorithms was investigated and the effects of scale in the studied system were evaluated. The results, obtained for both a CubeSat and a medium-sized satellite, the Amazonia-1, showed the potential and adequacy of the use of Neural Networks to simulate the thermal behavior of satellites in real time with high fidelity, corroborating the feasibility of its use as a thermal model for an Operational Satellite Simulator.

Keywords: Satellite Operational Simulator. Thermal Simulation. Real Time Simulation. Artificial Neural Networks.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1.1. Infraestruturas de simulação utilizadas no desenvolvimento de veículos espaciais.	3
Figura 2.1. Ambiente térmico de um satélite em órbita da Terra.....	14
Figura 2.2. Cenário de operação do Simulador dos Satélites CBERS 3&4.....	21
Figura 2.3. (a) Representação do fluxo radiativo incidente em um nó. (b) Discretização das direções (φ, ψ) de uma fonte radiativa.	27
Figura 2.4. Forma geral de um neurônio natural.	29
Figura 2.5. Modelo de um neurônio artificial.	30
Figura 2.6. Arquitetura geral de uma rede neural MLP.	32
Figura 2.7. Comparação entre os resultados simulados e os dados experimentais, para a temperatura ambiente de 50°C.	43
Figura 4.1. Arquitetura da rede utilizada para a esfera em órbita da Terra.	61
Figura 4.2. Comparação das curvas de temperatura $T(^{\circ}\text{C})$ em função do tempo $t(\text{s})$ para a esfera de alumínio.....	63
Figura 4.3. Modelo térmico do nano-satélite.	64
Figura 4.4. Curvas de temperatura em função do tempo, geradas pelo SINDA. 67	
Figura 4.5. Arquitetura da RNA para o CubeSat no estudo preliminar 1.	67
Figura 4.6. Curvas de temperatura geradas pela RNA e os pontos usados no treinamento.	68
Figura 4.7. Comparação entre as curvas geradas pela RNA e pelo SINDA. ...	69
Figura 4.8. Concepção artística do satélite Amazonia-1.	71
Figura 4.9. Arquitetura da RNA para o CubeSat no estudo preliminar 2.	73
Figura 4.10. Comparação das curvas de temperatura entre SINDA e RNA para quatro nós diferentes.....	74
Figura 4.11. Novo modelo térmico do CubeSat.....	76

Figura 4.12. Arquitetura da rede utilizada para o CubeSat nos estudos de generalização.	78
Figura 4.13. Comparação das curvas de temperatura entre SINDA e RNA para quatro cenários diferentes.	79
Figura 4.14. Comparação das curvas de temperatura entre SINDA e RNA para quatro cenários diferentes.	81
Figura 4.15. Arquitetura da RNA proposta para generalização das temperaturas do CubeSat.	85
Figura 4.16. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo GD.	87
Figura 4.17. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo GD.	88
Figura 4.18. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo GD.	89
Figura 4.19. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo GD.	90
Figura 4.20. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo LM.	91
Figura 4.21. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo LM.	92
Figura 4.22. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo LM.	93
Figura 4.23. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo LM.	94
Figura 4.24. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo GEO.	95
Figura 4.25. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo GEO.	96
Figura 4.26. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo GEO.	97

Figura 4.27. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo GEO.....	98
Figura 4.28. Comparação do erro médio de generalização para os três algoritmos.....	99
Figura 4.29. Arquitetura da RNA utilizada para a generalização para o satélite Amazonia-1.	104
Figura 4.30. Curvas de temperatura referentes ao o conjunto HotCold para o satélite Amazonia-1.....	105
Figura 4.31. Curvas de temperatura referentes ao o conjunto HotColdAB para o satélite Amazonia-1.....	106
Figura 4.32. Curvas de temperatura referentes ao o conjunto HotColdABCDE para o satélite Amazonia-1.....	107
Figura 4.33. Erro médio de generalização em função do número de cenários aleatórios adicionados para o Amazonia-1.	109
Figura 4.34. Comparação entre as curvas de temperatura obtidas pelo THERMICA-SYSTEMA e o CubeSat.	112
Figura 4.35. Curvas de temperatura calculadas pelo método RLM para o satélite Amazonia-1.....	114

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 2.1. Cargas térmicas médias para a órbita LEO.....	15
Tabela 2.2. Vantagens e desvantagens das abordagens de Integração e Interpolação.....	25
Tabela 4.1. Propriedades ópticas da esfera em órbita da Terra.....	60
Tabela 4.2. Cenários simulados para a esfera de alumínio.	62
Tabela 4.3. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.....	63
Tabela 4.4. Propriedades termo-físicas.....	65
Tabela 4.5. Propriedades ópticas.....	65
Tabela 4.6. Erro médio dos resultados da RNA em comparação com o SINDA. 70	
Tabela 4.7. Cenários simulados para o satélite Amazonia-1.	72
Tabela 4.8. Erros da comparação entre as curvas geradas pela RNA e pelo SINDA.	74
Tabela 4.9. Cenários simulados para o CubeSat.	77
Tabela 4.10. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.	80
Tabela 4.11. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.	81
Tabela 4.12. Cenários simulados para o CubeSat.....	86
Tabela 4.13. Erros médios de generalização para o algoritmo GD.	89
Tabela 4.14. Erros médios de generalização para o algoritmo LM.	93
Tabela 4.15. Erros médios de generalização para o algoritmo GEO.	97
Tabela 4.16. Erros médios de generalização para os três algoritmos.....	100
Tabela 4.17. Menores erros de generalização para os três algoritmos.....	101

Tabela 4.18. Erros médios de generalização para o satélite Amazonia-1, contendo apenas os cenários usados para teste.	108
Tabela 4.19. Erros médios de generalização para o satélite Amazonia-1, incluindo os cenários usados para validação.	108
Tabela 4.20. Menores erros médios e erros máximos de generalização para o melhor caso de treinamento para o satélite Amazonia-1.	110
Tabela 4.21. Erros médios e máximos de interpolação obtidos com o método RLM para o satélite Amazonia-1.	115
Tabela 4.22. Tempo médio para gerar 5 órbitas.	116

LISTA DE SIGLAS E ABREVIATURAS

ANN	Artificial Neural Network
AIT	Assembly, Integration and Test
CBERS	China-Brazil Earth Resources Satellite
CNES	Centre National d'Etudes Spatiales
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
ESOC	ESA Space Operations Center
ETRI	Electronics and Telecommunications Research Institute
GD	Gradiente Descendente
GEO	General Extremal Optimization
GPM	Global Precipitation Measurement
INPE	Instituto Nacional de Pesquisas Espaciais
LEO	Low Earth Orbit
LM	Levenberg-Marquardt
MCP	McCulloch-Pitts
MLI	Multi-Layer Insulation
MLP	Multi-Layer Perceptron
NASA	National Aeronautics and Space Administration
NOS	NASA Operational Simulator
OSR	Optical Solar Reflector
PCB	Printed Circuit Board
PMM	Plataforma Multi-Missão do INPE
REFA	ESA Reference Architecture
RLM	Regressão Linear Multivariada
RNA	Rede Neural Artificial
SIMULUS	ESA Simulation System Infrastructure
SIMSAT	ESA Satellite Simulator framework
SINDA	Systems Improved Numerical Differencing Analyzer
SMP	Simulation Modeling Platform
SysML	Systems Modeling Language
TBT	Thermal Balance Test
V&V	Verificação e Validação

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
1.1 Caracterização do Problema.....	1
1.2 Objetivo	7
1.3 Escopo da Tese	7
1.4 Metodologia da Pesquisa	8
1.5 Resultados Obtidos	9
1.6 Organização da Tese	10
2 REVISÃO BIBLIOGRÁFICA	13
2.1 Controle Térmico de Satélites	13
2.2 Simuladores de Satélites.....	16
2.2.1 Modelagem Térmica em Simuladores Operacionais	23
2.3 Redes Neurais Artificiais	28
2.3.1 Redes Neurais Aplicadas em Problemas de Engenharia Térmica	39
3 ABORDAGEM PROPOSTA PARA A SIMULAÇÃO EM TEMPO REAL DO COMPORTAMENTO TÉRMICO DE SATÉLITES.....	45
3.1 Formulação do Problema	45
3.2 Treinamento por Gradiente Descendente (GD)	46
3.3 Treinamento por Levenberg-Marquardt (LM)	49
3.4 Treinamento por Otimização Extrema Generalizada (GEO)	53
4 RESULTADOS E DISCUSSÃO	59
4.1 Estudos Preliminares	60
4.1.1 Esfera de alumínio.....	60
4.1.2 CubeSat.....	64
4.1.3 Satélite Amazonia-1.....	70
4.1.4 Inclusão de cenários intermediários	75
4.2 Generalização para Cenários não Treinados.....	82
4.2.1 CubeSat.....	84

4.2.2	Resultados para o algoritmo Gradiente Descendente	87
4.2.3	Resultados para o algoritmo Levenberg-Marquardt.....	90
4.2.4	Resultados para o algoritmo Otimização Extrema Generalizada..	94
4.2.5	Comparação entre os algoritmos.....	98
4.2.6	Satélite Amazonia-1.....	102
4.3	Comparação com Outras Metodologias.....	111
4.3.1	Comparação com o Método de Caracterização Térmica de Satélites	111
4.3.2	Comparação com Regressão Linear Multivariada.....	113
4.4	Análise do Tempo de Simulação.....	116
5	CONCLUSÕES E TRABALHOS FUTUROS	119
	REFERÊNCIAS BIBLIOGRÁFICAS	125
	APÊNDICE A – DADOS DE TREINAMENTO PARA O CUBESAT.....	133
	APÊNDICE B – CENÁRIOS SIMULADOS PARA O AMAZONIA-1.....	135
	APÊNDICE C – INTEGRAÇÃO NO SIMULADOR DO INPE.....	137
	APÊNDICE D – LISTA DE ARTIGOS PRODUZIDOS.....	141

1 INTRODUÇÃO

1.1 Caracterização do Problema

Sistemas Espaciais são formados, no mínimo, por um Segmento Espacial, um Segmento Solo e um Segmento Lançador (ECSS-S-ST-00-01C1, 2012).

O Segmento Espacial é composto por um ou mais veículos espaciais, que incluem satélites artificiais, sondas espaciais, estações espaciais, além de outros, como os já desativados ônibus espaciais. O objetivo dos veículos espaciais é carregar e manter cargas úteis embarcadas em funcionamento no espaço. As cargas úteis, por sua vez, consistem em um conjunto de instrumentos e/ou equipamentos, necessários para cumprir a missão estabelecida.

Os satélites artificiais (chamados daqui em diante simplesmente de satélites) podem ter diferentes finalidades: Sensoriamento Remoto, Meteorológicos, Comunicações, Navegação (localização por satélite), Científicos, Tecnológicos (para teste de novas tecnologias), etc.; e são classificados de acordo com sua massa: grandes (> 1000 kg), pequenos ou médios (500 a 1000 kg), mini (100 a 500 kg), micro (10 a 100 kg), nano (1 a 10 kg), etc. (FORTECUE et al., 2003).

O Segmento Solo é composto por pelo menos uma Estação de Solo, para recepção, armazenagem e distribuição de telemetrias e telecomandos; um Centro de Controle, para planejamento da missão, monitoramento e controle da dinâmica de voo, e realização de análises e relatórios de desempenho; a rede de comunicações, para distribuição dos dados; e, em geral, um Centro de Missão para tratamento, armazenamento e distribuição dos dados da Aplicação, isto é, dos dados da carga útil do satélite.

O Segmento Lançador, por sua vez, consiste no próprio veículo lançador (foguetes) e na infraestrutura necessária para manufatura, teste e entrega do lançador, cujo objetivo é transportar os veículos espaciais para o local desejado no espaço.

Além desses três segmentos, o desenvolvimento de Sistemas Espaciais requer também uma infraestrutura e serviços para o planejamento e definição do cronograma da missão; para o projeto, fabricação, integração e teste dos diferentes sistemas, subsistemas e equipamentos; para o lançamento, etc., que pode ser chamada de Segmento de Suporte.

O processo de desenvolvimento de sistemas espaciais é complexo e de alto risco. A complexidade advém da grande quantidade de requisitos que o sistema deve atender para cumprir os objetivos da sua missão e suportar os ambientes extremos encontrados no lançamento e no espaço. O alto risco, por sua vez, está relacionado com o impacto que falhas no sistema podem causar, podendo ser catastróficas e comprometer toda a missão. Além disso, os veículos espaciais geralmente não são produzidos em série, o que exige ainda mais planejamento e cuidado no seu desenvolvimento.

No espaço, os satélites enfrentam intensas variações de temperatura entre os momentos de exposição à luz do Sol e no eclipse. Além disso, a ausência de atmosfera implica na inexistência de fluxo convectivo de calor, o que dificulta muito a troca de calor entre as partes do satélite e a sua rejeição para o espaço (GILMORE, 2002). A radiação proveniente do meio espacial, por sua vez, pode interferir nos sensores e nos dados transmitidos, além de causar falhas em componentes eletrônicos. Por fim, a baixa gravidade, ou ausência de peso, afeta o comportamento de fluidos e tanques pressurizados.

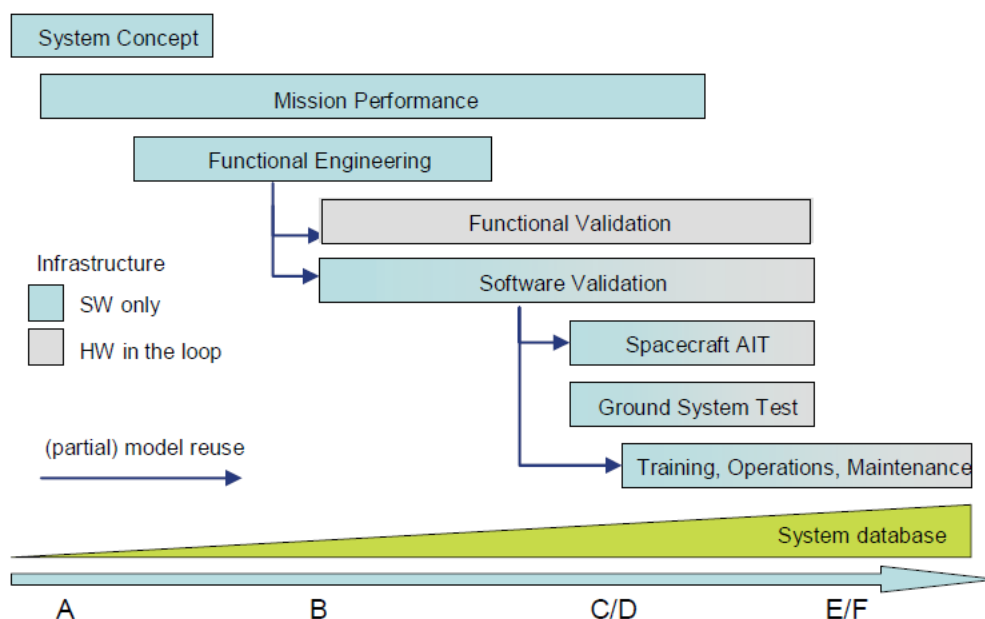
Reproduzir todos esses efeitos em um laboratório em solo é muito difícil. Normalmente, são realizados diversos testes físicos, porém, na maioria dos casos, apenas um dos efeitos pode ser reproduzido de cada vez (REGGESTAD et al., 2011).

A avaliação de falhas também é outro grande desafio, pois é inviável trazer um satélite à Terra e de volta ao laboratório para identificar a origem e corrigir ou substituir componentes.

Com o avanço da capacidade de processamento dos computadores, e também dos recursos de visualização gráfica, a modelagem e simulação computacionais vêm sendo utilizadas de forma cada vez mais frequente ao

longo de todo o ciclo de vida dos sistemas espaciais (ECSS-E-TM-10-21A, 2010). A Fig. 1.1 ilustra as diferentes infraestruturas de simulação e teste utilizadas ao longo do ciclo de vida de veículos espaciais.

Figura 1.1. Infraestruturas de simulação utilizadas no desenvolvimento de veículos espaciais.



Fonte: ECSS-E-TM-10-21A (2010).

Os blocos em azul claro representam as infraestruturas de simulação baseadas apenas em software, enquanto a cor cinza significa a utilização de equipamentos do veículo espacial conectados aos sistemas em software, chamado em inglês de *hardware in the loop*. As setas indicam o reuso dos modelos de uma fase para outra.

As simulações são usadas desde as fases iniciais com a definição do conceito do sistema; passando pelo estabelecimento dos requisitos de desempenho da missão; pelos processos de engenharia funcional; validação funcional; validação do software; montagem, integração e testes (AIT); teste do Sistema de Solo; e nas fases finais para treinamento, operações e manutenção.

Cada infraestrutura possui um modelo de sistema virtual, que evolui juntamente com a definição do sistema espacial durante cada fase; da mesma forma, o banco de dados do sistema é populado incrementalmente e validado ao longo das fases de desenvolvimento do satélite.

Em qualquer que seja a fase, o uso de modelagem e simulação computacionais tem contribuído significativamente para a redução de custos e prazos, ao mesmo tempo em que aumenta a confiabilidade do sistema ao auxiliar na identificação de riscos e prevenção de falhas.

Os simuladores usados para treinamento, operações e manutenção, chamados aqui simplesmente de Simuladores Operacionais de satélites, são usados extensivamente antes do lançamento para validar o sistema de controle de solo, treinar a equipe de operadores deste sistema e validar previamente os procedimentos operacionais antes de realizá-los com o satélite real (antes ou após o lançamento). O Simulador Operacional é um sistema complexo que permite executar cenários completos de forma tão realista que é virtualmente impossível notar a diferença entre um cenário simulado e a operação real do satélite (REGGESTAD et al., 2011).

Para atingir esse objetivo, o Simulador Operacional deve fornecer informações em tempo real, isto é, as informações devem estar disponíveis ao operador no mesmo tempo em que estariam caso o satélite real estivesse conectado ao sistema de controle do satélite. Isso implica que todos os módulos (ou modelos) internos do simulador devem ser executados em um tempo curto o suficiente para que a resposta do sistema como um todo seja dada em tempo real.

Dentre os diversos modelos que um simulador operacional deve executar, a simulação do comportamento térmico é um dos modelos que apresenta o maior custo computacional (consequentemente demorando mais tempo para ser processado), pois os métodos computacionais necessários para determinar a distribuição de temperaturas em um satélite, com alta fidelidade, requerem a resolução de uma grande quantidade de equações diferenciais para cada instante em que se deseja obter essa informação. Além disso, a informação

sobre o comportamento térmico influencia e também depende de outros modelos como de controle de potência, de controle de atitude e, em última instância, do estado de funcionamento de cada um dos equipamentos do satélite.

Tradicionalmente, são encontradas na literatura duas alternativas para simular o comportamento térmico de satélites em tempo real: a interpolação sobre cenários pré-definidos e a integração de um modelo bastante simplificado (PERPIÑÁN, 1994).

No primeiro caso, a interpolação é realizada entre cenários nos quais o comportamento térmico do satélite foi pré-determinado. Apesar de produzir bons resultados para esses cenários, o processo de interpolação pode produzir resultados com erros imprevisíveis para os cenários que não foram calculados previamente, pois a temperatura varia de forma não linear e depende da combinação de muitas variáveis do sistema.

No segundo caso, o modelo térmico do satélite é bastante simplificado de forma que seja possível integrar as equações de transferência de calor em tempo real. A desvantagem desta abordagem é a perda da fidelidade em comparação com o satélite real para qualquer cenário.

Considerando essas limitações, neste trabalho é proposta a utilização de Inteligência Computacional ou Aprendizado Computacional (*Machine Learning*), particularmente as Redes Neurais Artificiais (chamadas adiante simplesmente de redes neurais ou RNAs) do tipo *Multi-Layer Perceptron* (MLP), para modelagem e simulação em tempo real do comportamento térmico de satélites.

Ao longo das últimas décadas, as RNAs se tornaram um vasto campo de estudo e diversas arquiteturas e algoritmos de treinamento diferentes têm sido propostos com o objetivo de aprimorar o seu desempenho (PIOTROWSKI et al., 2014; OJHA et al., 2017). Além disso, as redes neurais têm sido aplicadas com sucesso em uma grande variedade de problemas, nas mais diversas áreas do conhecimento (YANG, 2008; YADAV et al., 2014; CHOJACZYK et al., 2015).

A escolha das RNAs para modelagem e simulação em tempo real do comportamento térmico de satélites se deve à sua capacidade de mapear relações complexas entre grandes conjuntos de dados de entrada e saída, bem como, de interpolar ou generalizar, isto é, fornecer saídas razoáveis para entradas não fornecidas no treinamento, a partir do conhecimento adquirido (HAYKIN, 2001; HAGAN, 2014). Além disso, depois de encerrado o processo de treinamento, as RNAs podem fornecer saídas com um custo computacional extremamente baixo, atendendo ao requisito de tempo real de um simulador operacional.

Existem diferentes arquiteturas e metodologias de RNAs na literatura tais como, Redes de Funções de Base Radial (*Radial Basis Functions*), Máquinas de Vetor de Suporte (*Support Vector Machines*), Redes Neurais Convolucionais (*Convolutional Neural Networks*), Redes Neurais de Aprendizado Profundo (*Deep Learning Neural Networks*), etc. Dentre as arquiteturas existentes, as redes MLPs são as mais utilizadas para aproximação de funções e identificação de sistemas dinâmicos (HAYKIN, 2001; HAGAN, 2014; CHOJACZYK et al., 2015; OJHA et al., 2018). Por essa razão, neste trabalho optou-se por utilizar redes MLPs canônicas e a sua forma recorrente, otimizando os seus parâmetros para o problema específico.

As redes do tipo *Multi-Layer Perceptron* possuem três características distintivas: uma função de ativação não linear, uma ou mais camadas de neurônios ocultos e um alto grau de conectividade. A combinação dessas características e a habilidade de “aprender” por meio do processo de treinamento, conferem a essas redes um poder computacional significativo. As redes MLP recorrentes, por sua vez, são aproximadores universais de sistemas dinâmicos não lineares (LO et al., 1993). Sendo estas características as que melhor satisfazem os objetivos desta Tese.

1.2 Objetivo

O objetivo principal desta Tese é verificar a aplicabilidade de Redes Neurais Artificiais, para determinar o comportamento térmico de Satélites Artificiais em órbita da Terra, com elevada fidelidade em comparação com o sistema real e em um tempo de processamento compatível com tempo real, atendendo aos requisitos de um Simulador Operacional de Satélites.

1.3 Escopo da Tese

No projeto de doutorado descrito nesta Tese foi estudada a aplicação de RNAs para a simulação em tempo real do comportamento térmico de satélites. Com o objetivo de obter resultados simulados de temperatura com alta fidelidade, foram testados diferentes algoritmos para treinamento das redes neurais. Além disso, foram realizados diferentes estudos de caso para verificar a aplicabilidade dessa solução, bem como, avaliar os efeitos da escala no sistema estudado.

O escopo deste trabalho está na solução para o problema da simulação em tempo real do comportamento térmico de satélites artificiais com alta fidelidade, para utilização em Simuladores Operacionais. Sendo uma pesquisa no estado da arte de Engenharia de Sistemas Espaciais. A contribuição para a área de Redes Neurais Artificiais ou Aprendizagem de Máquina limita-se ao conhecimento adquirido para sintonizar as redes a este tipo de aplicação espacial.

A técnica de redes neurais foi escolhida aqui por ser amplamente utilizada e recomendada para a solução de problemas de regressão em aprendizado de máquina (HAYKIN, 2001; HAGAN, 2014). Sendo, regressão o mapeamento da relação de dependência entre variáveis contínuas. Além disso, uma vez identificado o sistema pela rede, a resposta desta a variações nas suas

entradas se dá de maneira praticamente imediata, cumprindo assim o requisito de tempo real de simuladores operacionais.

Posteriormente, espera-se que o resultado deste trabalho seja incorporado aos simuladores operacionais que vêm sendo desenvolvidos no INPE para os seus programas de satélites. Uma vez que, em praticamente todas as missões de satélites deve ser desenvolvido um Simulador Operacional específico e, de preferência, que faça uso de soluções comuns (AZEVEDO et al., 2012).

1.4 Metodologia da Pesquisa

A metodologia geral da pesquisa envolveu, inicialmente, a aquisição de conhecimentos referentes à engenharia de sistemas espaciais, ao controle térmico de satélites, aos simuladores de sistemas espaciais e às redes neurais artificiais.

Em seguida, foram iniciados os estudos com a aplicação de RNAs para simulação do comportamento térmico de satélites. Os padrões de treinamento de entrada e saída da rede MLP utilizados nesta Tese foram obtidos a partir de simulações realizadas com o software SINDA/FLUINT (C&R TECHNOLOGIES, 2013). A complexidade dos problemas resolvidos por redes neurais neste trabalho consiste em arquiteturas com 4 a 95 entradas e 1 a 50 saídas, e conjuntos de dados variando de 202 a 1414 padrões de treinamento.

Primeiramente, foram realizados alguns estudos preliminares para reprodução das temperaturas de veículos espaciais em órbita da Terra.

O primeiro estudo foi reproduzir a temperatura de uma esfera de alumínio em órbita da Terra. Para treinamento da rede foi utilizado o algoritmo do gradiente descendente.

O segundo estudo foi aplicar a mesma metodologia para um *CubeSat* (um pequeno satélite formado por uma ou mais unidades em forma de cubo; WOELLERT et al. 2011), que foi modelado com três equipamentos dissipativos em seu interior.

Na sequência, foi treinada uma rede neural com o objetivo de reproduzir as temperaturas do satélite Amazonia-1 (um satélite de médio porte atualmente em desenvolvimento no INPE) (INPE, 2018), para os estados quente e frio. Visando avaliar a escalabilidade da solução.

Em seguida, o modelo do CubeSat é retomado para testar a generalização da rede. Como esse sistema permite a variação de diversos modos de funcionamento, com a combinação de diferentes valores para a potência dos equipamentos, foram simulados diversos cenários combinando variações nas variáveis ambientais e nos modos de funcionamento. As RNAs foram então, utilizadas para armazenar conhecimento sobre alguns cenários e, em seguida, foi estudada a sua capacidade de generalizar para cenários não apresentados no treinamento.

Para contornar as limitações da rede neural utilizada, foram propostas mudanças na arquitetura da rede e a aplicação de diferentes algoritmos de treinamento, para o modelo do CubeSat, com o objetivo de aprimorar sua capacidade de generalização e também de torná-la menos suscetível a mudanças drásticas nas variáveis que representam o cenário sendo simulado.

Na sequência, essa metodologia foi aplicada ao satélite Amazonia-1 (SILVA et al., 2014), com o objetivo de obter a distribuição de temperaturas para cenários não apresentados à RNA durante o treinamento.

1.5 Resultados Obtidos

Os resultados desta Tese, obtidos a partir da utilização de redes neurais artificiais para simulação do comportamento térmico de satélites, apresentaram precisão e custo computacional compatível com os requisitos de simulação em tempo real do comportamento térmico de um satélite.

Os diversos experimentos computacionais realizados em modelos de veículos espaciais de diferentes complexidades e escalas evidenciaram que as

RNAs podem não somente reproduzir o seu comportamento térmico, mas também generalizar para cenários não apresentados durante o treinamento.

Em todos os casos aplicados, erros médios de temperatura abaixo do limite de 5°C, estabelecido como requisito para o Simulador Operacional, foram atingidos. Os melhores resultados de generalização foram obtidos utilizando redes MLP Recorrentes treinadas com o algoritmo do Gradiente Descendente (GD), incluindo diferentes cenários aleatórios no conjunto de treinamento, além de modificações nas variáveis de entrada da rede.

No caso do satélite Amazonia-1, foi observado um importante ganho de tempo de processamento, partindo de um tempo de cerca de 10h para simulação de 5 órbitas para um cenário específico utilizando o software SINDA, para um tempo médio de simulação com a RNA de cerca de 0,088s para gerar as mesmas 5 órbitas, o que representa um ganho de três ordens de grandeza.

1.6 Organização da Tese

O conteúdo desta Tese está organizado em cinco capítulos principais, com a seguinte denominação:

- Introdução;
- Revisão Bibliográfica;
- Abordagem Proposta para a Simulação em Tempo Real do Comportamento Térmico de Satélites;
- Resultados e Discussão;
- Conclusões e Trabalhos Futuros;
- Referências Bibliográficas;
- Apêndices.

Este primeiro Capítulo consiste, na introdução dos conceitos essenciais, apresentação do problema objeto da pesquisa, proposição da solução estudada, delimitação do escopo do projeto, descrição da metodologia de pesquisa utilizada, dos resultados obtidos e da organização desta Tese.

A revisão bibliográfica, apresentada no Capítulo 2, está dividida nos seguintes temas: controle térmico de satélites, simuladores de satélites e, por fim, redes neurais artificiais, que inclui uma subseção específica sobre o uso de redes neurais aplicadas a problemas térmicos de engenharia.

O Capítulo 3 contém a estrutura conceitual e metodológica desenvolvida ao longo do projeto para a aplicação de redes neurais na simulação do comportamento térmico de satélites. Na Seção 3.1, são descritos a formulação e a modelagem do problema e nas seções seguintes os três diferentes algoritmos de treinamento utilizados nesta Tese: Gradiente Descendente, Levenberg-Marquardt e Otimização Extrema Generalizada.

Os resultados e as discussões são descritos no Capítulo 4, no qual são apresentados os estudos preliminares e os resultados obtidos para estudar a capacidade de generalização das redes neurais ao problema sendo atacado.

O primeiro estudo preliminar consiste na reprodução da variação da temperatura de uma esfera em órbita da Terra. Na sequência, são apresentados os resultados para a reprodução da variação da temperatura de um CubeSat e também para o satélite Amazonia-1.

Na segunda parte do Capítulo 4, são apresentados os resultados da pesquisa realizada com o objetivo de obter a generalização da rede neural para cenários não apresentados no treinamento tanto para o modelo do CubeSat, utilizando três algoritmos diferentes para treinamento da rede neural, quanto para o satélite Amazonia-1. Adicionalmente, são também apresentadas comparações com metodologias encontradas na literatura.

As conclusões finais são apresentadas no Capítulo 5, considerando os diferentes casos, algoritmos e estudos. Neste capítulo, são discutidas também as possibilidades de extensão e continuidade do projeto em trabalhos futuros.

No Apêndice A, consta um exemplo de conjunto utilizado para treinamento da rede neural. No Apêndice B, é mostrada a tabela dos cenários simulados para o satélite Amazonia-1, utilizada na Subseção 4.2.6. No Apêndice C é discutida a integração do Modelo Térmico baseado em Rede Neural nos

futuros Simuladores Operacionais do INPE. Por fim, no Apêndice D consta a lista dos artigos escritos ao longo do projeto.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados conceitos relacionados aos temas Controle Térmico de Satélites, Simuladores de Satélites e Redes Neurais Artificiais.

Os livros citados foram selecionados de acordo com a sua relevância para cada tema específico e também foram utilizadas as edições mais recentes, quando disponíveis.

Os artigos científicos foram selecionados primeiramente em função da relevância do tema para o escopo do presente projeto. Em segundo lugar, foram considerados a data de publicação e o número de citações.

2.1 Controle Térmico de Satélites

No projeto de sistemas espaciais, o subsistema de controle térmico tem a função de manter a temperatura de todos os equipamentos e componentes dentro de limites aceitáveis para o seu funcionamento adequado, durante toda a vida do satélite.

O controle térmico de satélites deve ser planejado de tal forma que o sistema esteja protegido das variações de temperatura em diferentes ambientes ao longo do seu ciclo de vida (GILMORE, 2002), tanto no lançamento quanto em órbita.

O ambiente de lançamento pode ser subdividido em três etapas:

- I. pré-lançamento, onde o satélite é geralmente mantido em um ambiente com temperatura controlada de 10°C a 25°C;
- II. fase inicial do lançamento, onde na superfície da coifa do lançador, a temperatura pode atingir a faixa dos 200°C; e
- III. após a abertura da coifa, onde o satélite fica sujeito ao aquecimento decorrente do atrito com moléculas livres da atmosfera superior. Embora sujeito a essas variações abruptas, o ambiente de

lançamento usualmente não impõe restrições importantes para o controle térmico, já que o tempo para o satélite atingir a órbita é muito curto (em torno de 10 minutos), não havendo tempo para que as cargas térmicas aplicadas ao satélite durante o lançamento tenham impacto na maioria dos seus equipamentos.

É no ambiente orbital que o satélite passa a maior parte da sua vida útil e, por essa razão, deve receber maior atenção. As principais fontes de cargas térmicas de um satélite em órbita baixa são a radiação solar direta, a radiação solar refletida pela Terra (albedo) e a radiação infravermelha emitida pela Terra. O controle térmico de um satélite em órbita é normalmente realizado por meio do balanceamento entre essas cargas, a energia emitida pelo satélite em infravermelho e a energia dissipada internamente pelos seus componentes elétricos. A Fig. 2.1 ilustra as diferentes fontes de radiação a que um satélite está sujeito em órbita terrestre.

Figura 2.1. Ambiente térmico de um satélite em órbita da Terra.



Fonte: Adaptado do vídeo INPE-CBERS (2011).

A intensidade dessas grandezas varia em função da órbita do satélite. As principais órbitas terrestres utilizadas são a órbita baixa, chamada em inglês de

Low Earth Orbit (LEO), utilizada principalmente para observação da Terra; a órbita média, em inglês *Medium Earth Orbit*, utilizada pelos satélites de geoposicionamento; a Geoestacionária, utilizada para comunicação e meteorologia; e a órbita *Molniya*, utilizada principalmente para comunicação em países próximos dos polos.

As cargas térmicas originárias da Terra (albedo e infravermelha) são mais intensas para órbitas baixas, devido à maior proximidade da superfície terrestre. Conforme a altitude da órbita aumenta, essas cargas decrescem rapidamente. A Tabela 2.1 apresenta valores médios de radiação térmica para órbita LEO.

Tabela 2.1. Cargas térmicas médias para a órbita LEO.

Tipo de Radiação	Valor Médio (W/m²)
Radiação Solar	1366,1
Albedo	409,8*
Radiação Infravermelha	230

* considerando um fator de Albedo de 0,3

Fonte: ECSS (2008).

Outro fator importante na determinação das cargas térmicas de um satélite é o tipo de estabilização da atitude. Nos satélites estabilizados por rotação, as cargas se distribuem mais homoganeamente nas superfícies paralelas ao eixo de rotação. Já na estabilização em três eixos, há uma maior variação na distribuição das cargas térmicas em função da diferença na exposição das superfícies.

No interior do satélite, as transferências térmicas ocorrem principalmente por radiação e condução. A convecção ocorre apenas em equipamentos ou componentes que possuam gás ou fluido no seu interior, como os tubos de calor. O calor produzido pela dissipação dos equipamentos elétricos é

transmitido por meio da emissão de radiação e pela condução térmica para a estrutura. O excesso desse calor gerado internamente é conduzido para fora do satélite por meio de radiadores.

O subsistema de controle térmico pode ser constituído por elementos passivos e ativos. Os itens passivos incluem mantas multicamada superisolantes (MLI), revestimentos, materiais colocados nas interfaces para isolar ou melhorar o acoplamento térmico (*fillers, washers, doublers*), radiadores, superfícies ópticas refletoras (OSR), etc. Os elementos de controle ativo incluem aquecedores elétricos resistivos, circuitos com bombeamento mecânico, resfriadores termoelétricos, *cryocoolers*, etc. Alguns elementos, como tubos de calor e persianas térmicas, podem ser passivos ou ativos em função da sua dependência ou não de energia elétrica.

Para o desenvolvimento do projeto térmico devem ser considerados, para cada equipamento: os intervalos de temperatura operacionais e não operacionais, se existe uma grande variação nas cargas térmicas externas ou na dissipação do equipamento ao longo da órbita, se o equipamento apresenta alta densidade de dissipação de calor, se exige estrito controle de temperatura, se é necessário isolamento térmico e se o equipamento necessita trabalhar em temperatura criogênica.

2.2 Simuladores de Satélites

O desenvolvimento de satélites em todo o mundo mudou significativamente nas últimas décadas e tem sido acelerado e otimizado pelas modernas ferramentas de simulação. O método clássico de desenvolver e testar diversos modelos físicos de um satélite e seus subsistemas e equipamentos está sendo substituído cada vez mais por essa nova metodologia, consideravelmente mais rápida e econômica. Essa nova abordagem não mais inclui uma grande quantidade de modelos físicos para testes estruturais, térmicos e funcionais, sendo uma boa parte destes substituída por modelos e simulações

computacionais. Dessa forma, não só os tempos de execução do projeto podem ser reduzidos, como também podem ser realizados testes de integração e de software simulados, contribuindo para o desenvolvimento de sistemas cada vez mais complexos (HENDRICKS; EICKHOFF, 2005; EICKHOFF, 2009).

A aplicação de infraestruturas modernas de simulação durante o desenvolvimento de satélites permite a verificação consistente da funcionalidade de todos os subsistemas isoladamente ou em conjunto. Além disso, as simulações contribuem para a avaliação de diferentes cenários e modos de operação relacionados a projetos específicos. Similarmente, é possível checar a confiabilidade e a segurança do sistema, além de avaliar questões como degradação e tempo de vida. De uma forma geral, os riscos podem ser reduzidos de maneira significativa e os sistemas podem ser produzidos de forma mais eficiente em termos de custo, qualidade, confiabilidade e em um prazo mais curto (PIDGEON; BÉGIN, 2000; IRVINE et al., 2013).

O memorando técnico ECSS-E-TM-10-21A – System Modeling and Simulation (ECSS, 2010) define diversos tipos de simuladores de acordo com a sua função: concepção do sistema, desempenho da missão, simulador de engenharia funcional, simulador de montagem, integração e testes, simulador de teste do sistema de solo e simulador de treinamento, operações e manutenção.

No Simulador de Treinamento, Operações e Manutenção, chamado aqui apenas de Simulador Operacional, a simulação do sistema é necessária para garantir que o segmento solo e a equipe de operações estejam prontos para apoiar as atividades de operação nominais e de contingência.

O desafio principal desse tipo de simulador é a concepção de um modelo de alta fidelidade do satélite e suas interfaces com o segmento solo, com ênfase em prover uma simulação altamente representativa da plataforma do satélite e controle da carga útil, gerenciando telemetrias e telecomandos.

Esse simulador deve representar o comportamento do satélite e da sua carga útil, de forma que, para a equipe de controle de voo, seus efeitos nas

telemetrias sejam indistinguíveis (tanto quanto possível) do satélite real. O simulador deve executar o software de bordo sem modificação. Também deve modelar as estações de solo e a interface de rede para permitir a conexão direta com o sistema de controle da missão, além de permitir injeção de falhas pré-definidas pelo condutor da simulação.

Para atingir esses objetivos, o Simulador Operacional deve reproduzir o funcionamento e o comportamento do satélite com elevada fidelidade e, também, deve ser capaz de produzir dados em tempo real, para se comunicar com o sistema de solo e para que sua operação ocorra sem atrasos ou interrupções. Atender a esses requisitos é um grande desafio, dada a complexidade dos sistemas espaciais e o custo computacional necessário para simular os diversos subsistemas.

O simulador completo é composto por um conjunto de módulos de software que representam os subsistemas espaciais e terrestres e as especificidades da missão que irá apoiar. Todos os subsistemas do satélite devem estar presentes no simulador operacional e seus modelos computacionais precisam não apenas representar o comportamento e o desempenho do subsistema real, mas também devem ter um baixo custo computacional, de tal modo que o simulador possa produzir informações em tempo real ou quase real.

Na Agência Espacial Europeia (*European Space Agency – ESA*), um novo simulador é desenvolvido para cada missão fazendo extensivo reuso do software de infraestrutura genérica de simulação, chamado SIMULUS (REGGESTAD et al., 2012). O Simulador Operacional de Veículos Espaciais da ESA é composto por modelos de alta fidelidade do veículo espacial real e das suas interfaces com o Segmento Solo. O seu principal objetivo é prover uma simulação representativa do comportamento do veículo espacial, o que é obtido por meio de um emulador do processador de bordo, capaz de executar o software de bordo real.

O SIMULUS é composto por uma estrutura de tempo de execução, chamada SIMSAT, emuladores em software, um conjunto de modelos genéricos reutilizáveis e um conjunto de modelos de equipamentos de estações terrenas.

Ele permite obter ciclos de desenvolvimento mais curtos e, portanto, mais eficientes, por meio de reuso, enquanto produz resultados de maior qualidade.

O SIMSAT inclui um mecanismo de agendamento e execução baseado em um conceito típico de simulação de eventos discretos. Os eventos são gerados na execução dos modelos de simulação e são etiquetados com o tempo de simulação específico. O agendador de simulação serializa toda a execução do evento e mantém o tempo da execução geral usando um mecanismo de relógio global.

Para o desenvolvimento de simuladores operacionais, a ESA estabeleceu o padrão SMP (Simulation Modeling Platform); (ECSS, 2008), que estabelece interfaces padronizadas entre os modelos de simulação e o ambiente em tempo de execução para permitir serviços de simulação padronizados, bem como mecanismos de apoio à comunicação entre modelos.

Desenvolvimentos de simuladores recentes são baseados na Arquitetura de Referência (REFA), que foi introduzida com o objetivo de identificar, definir, desenvolver e integrar uma arquitetura de simulador operacional de referência. A REFA, e todos os simuladores baseados nesta, estão em conformidade com o padrão SMP.

O Simulador Operacional da NASA (NOS; ZEMERICK, 2013), possui uma arquitetura de software de simulação de alta fidelidade capaz de executar o software de voo do veículo espacial sem modificações. Ele pode ser executado em laptops e a partir de um pen drive. A sua arquitetura em camadas permite uma solução genérica para simulações distribuídas, provém uma solução para sincronização do tempo, permite a modificação dos dados de teste durante a execução e facilita o desenvolvimento de camadas de aplicações adicionais.

A capacidade de injeção de erros permite a execução de atividades de verificação e validação (V&V) para a redução do risco do software de voo. O NOS inclui um conjunto reutilizável de modelos em software que emulam placas de hardware de voo, de forma que falhas de hardware podem ser simuladas e testes não nominais podem ser executados. Além disso, inclui extensões genéricas reutilizáveis que permitem o uso das interfaces de

comando e telemetria do sistema operacional de solo, usando os mesmos sistemas, bancos de dados e *scripts* usados para o desenvolvimento do veículo espacial e as atividades de V&V.

A arquitetura NOS foi inicialmente aplicada no desenvolvimento de um simulador operacional para a missão *Global Precipitation Measurement (GPM)* do *Goddard Space Flight Center*, uma missão de satélite internacional com o objetivo de fornecer medidas de precipitação a partir do Espaço, avançar a compreensão dos ciclos de água e energia e estender o uso de dados de precipitação para benefício direto da sociedade. O Simulador Operacional GPM inclui o sistema solo, bancos de dados de comandos e telemetria, o software executável de voo e o simulador do veículo espacial.

A arquitetura NOS foi também utilizada no desenvolvimento de um simulador para o telescópio espacial James Webb. A arquitetura reutilizável permitiu que a equipe de desenvolvimento focasse em componentes de simulação específicos para a missão, reduzindo o risco ao cronograma e custos associados com o desenvolvimento do ambiente. O tempo de desenvolvimento foi reduzido em 50% em comparação com o que foi necessário para o Simulador Operacional GPM.

Os autores ressaltam ainda que, enquanto a arquitetura NOS tem sido utilizada primariamente em missões da NASA, sua aplicabilidade pode ser facilmente estendida para outros domínios como defesa e transportes.

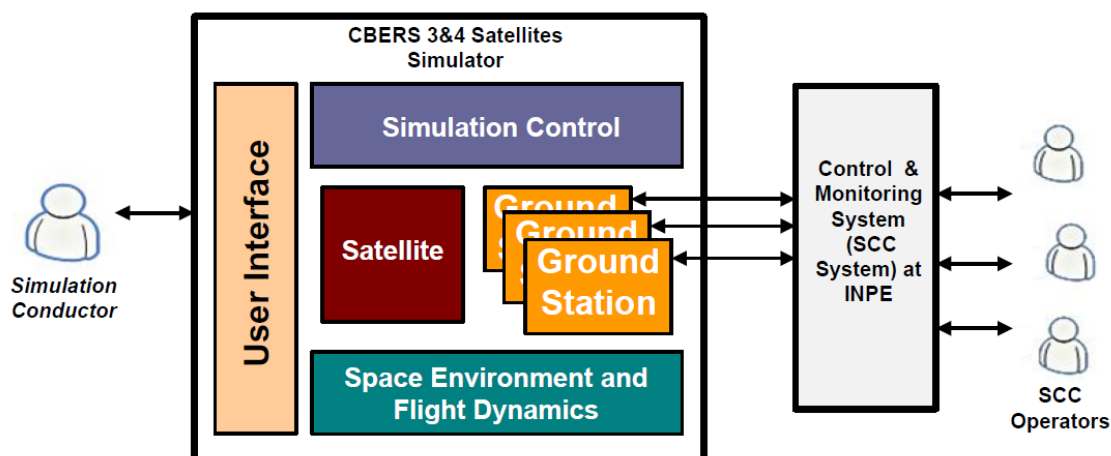
Há mais de uma década, o INPE vem desenvolvendo e aperfeiçoando Simuladores de Treinamento, Operações e Manutenção. Em especial, o Simulador Operacional desenvolvido para os satélites CBERS 3 e 4 (AMBROSIO et al., 2006; TOMINAGA et al., 2012), que poderá ser adaptado também para simuladores de satélites de outras plataformas como a PMM (Plataforma Multi-Missão; INPE, 2018).

Os principais objetivos do Simulador dos Satélites CBERS-3&4 são: apoiar as atividades de operação do satélite; treinar os operadores; testar o software de controle; avaliar procedimentos operacionais normais e adversos, antes da execução no satélite real (INPE, 2016a e 2016b).

Entre as características desse Simulador destacam-se a capacidade de prover: (i) o comportamento de cada subsistema; (ii) a dinâmica de voo; (iii) o ambiente espacial; e (iv) as estações terrenas. Adicionalmente, o Simulador deve reproduzir as reações de um subsistema causadas pelas mudanças de estado em outros subsistemas, bem como as alterações nas condições ambientais do satélite em órbita da Terra.

A Fig. 2.2 ilustra um cenário de operação do Simulador dos Satélites CBERS 3&4. À esquerda está o Condutor da Simulação, responsável por definir os parâmetros, operar e monitorar o simulador. Ao centro (bloco maior) está o Simulador propriamente dito, com os seus módulos principais: Controle da Simulação, Interface com o Usuário, Satélite (que contém todos os modelos dos subsistemas), Ambiente Espacial e Dinâmica de Voo e as Estações Terrenas. À direita do Simulador está o Sistema de Controle e, na sequência, estão os Operadores do Sistema de Controle do Satélite. O Simulador se comunica diretamente com o Sistema de Controle imitando o próprio satélite em órbita e as estações terrenas. O modelo do comportamento térmico, objeto de estudo desta Tese, está inserido dentro do bloco denominado Satélite.

Figura 2.2. Cenário de operação do Simulador dos Satélites CBERS 3&4.



Fonte: INPE (2016a).

Com o aprimoramento dos seus diversos modelos (incluindo a contribuição desta Tese), esse simulador também poderá ser utilizado para avaliar a situação do satélite em órbita, para acompanhar a degradação dos componentes e também contribuir para a análise de falhas que podem ocorrer no seu funcionamento durante o tempo de vida do satélite. Além disso, os resultados das simulações permitirão definir procedimentos operacionais que podem maximizar o uso dos recursos do satélite, prolongando seu tempo de vida.

Sobre uma infraestrutura ou núcleo, são agregados os modelos que representam as partes específicas da missão espacial que serão simuladas. Dentre os modelos que são utilizados para o projeto e análise de um satélite, o modelo térmico é um dos que apresenta o maior custo computacional, pois para calcular a distribuição de temperaturas e a dinâmica de transferência de calor é necessário resolver um sistema de equações diferenciais que modelam o comportamento térmico do satélite.

Na concepção do projeto térmico de satélites comumente são utilizados softwares especializados (geralmente pacotes comerciais), nos quais o satélite é discretizado em um conjunto de nós que representam os vários componentes do satélite. Este conjunto pode chegar a milhares de nós em um satélite médio como o Amazônia-1 (SILVA et al., 2014). O conjunto discretizado de equações é resolvido utilizando-se principalmente os métodos de diferenças finitas e de elementos finitos, e deve ser resolvido a cada instante que se deseja saber o estado térmico do satélite. Esse tipo de simulação, aplicada para um satélite completo, em diferentes cenários de operação e condições ambientais, apresenta um alto custo computacional (cerca de 10h para cada cenário no caso do Amazonia-1), o que torna inviável sua utilização direta em um Simulador Operacional.

O Simulador dos Satélites CBERS 3&4 utiliza um modelo térmico simplificado baseado em regras (INPE, 2016b). Neste modelo, as temperaturas são limitadas aos seus valores máximos e mínimos nominais. Os valores são incrementados ou decrementados caso o satélite esteja sendo iluminado ou em

eclipse, respectivamente, e também de acordo com o status de funcionamento dos equipamentos.

Uma forma alternativa de obter esses dados com maior fidelidade, a um custo computacional reduzido e, conseqüentemente, em um tempo compatível com os requisitos de um Simulador Operacional, é realizar uma interpolação sobre um conjunto finito de cenários pré-selecionados, para os quais o estado térmico é calculado de antemão. A dificuldade de realizar essa interpolação é que o comportamento térmico do satélite se modifica de maneira não linear e complexa em função da variação dos parâmetros ambientais e do seu estado de operação. A consequência é a redução na precisão dos resultados obtidos em comparação com o sistema real. Além disso, as técnicas tradicionais de interpolação (polinomial, trigonométrica, etc.), quando aplicadas a um sistema simulador, requerem o armazenamento e a manipulação de uma grande quantidade de dados referentes aos cenários pré-determinados.

2.2.1 Modelagem Térmica em Simuladores Operacionais

Nesta seção, é apresentada uma revisão bibliográfica sobre as abordagens utilizadas para a modelagem do comportamento térmico em simuladores operacionais de satélites.

Eickhoff (2009) argumenta em seu livro *Simulating Spacecraft Systems*, que simuladores elaborados durante o desenvolvimento do sistema para testes funcionais, podem ser aplicados ao suporte das operações na estação de controle em solo e treinamento de operadores. A vantagem seria que tais simuladores já teriam passado por um extenso processo de verificação. Entretanto, a aceitação desses simuladores para apoio à operação varia de agência para agência.

O Centro de Operações Espaciais da ESA (*ESA Space Operations Center – ESOC*) normalmente não utiliza simuladores oriundos do ciclo de desenvolvimento devido a sua filosofia de utilizar apenas ferramentas

desenvolvidas independentemente (EICKHOFF, 2009). Esta abordagem minimiza o risco de erros potenciais inerentes ao processo de desenvolvimento.

De acordo com Eickhoff (2009), no caso de simuladores que apoiam o processo de desenvolvimento de veículos espaciais e não a sua operação e manutenção, a modelagem térmica pode conter somente os aspectos térmicos na ordem correta de magnitude. Contudo, como citado na Seção 2.2, essa precisão é insuficiente para o simulador operacional que é a motivação deste trabalho.

Kang et al. (1995) desenvolveram um simulador em tempo real para apoiar as atividades de operação de solo do sistema de controle de satélites do ETRI (*Electronics and Telecommunications Research Institute* – República da Coreia). As aplicações incluem: teste do sistema de solo, validação dos procedimentos de controle de voo, verificação dos comandos de satélites e treinamento dos operadores de solo. O modelo desenvolvido simula as características de um satélite de comunicações geoestacionário. Segundo os autores, trata-se de uma ferramenta de simulação de alta fidelidade, que pode ser usada em computadores desktop de baixo custo.

O modelo do subsistema de controle térmico simula o estado de comutação dos aquecedores e calcula a temperatura dos nós no satélite de acordo com os ângulos do Sol. O modelo exporta dados de telemetria para a memória compartilhada. A maioria dos comandos térmicos está na categoria de habilitar/desabilitar aquecedores e unidades de controle de aquecedores. Os autores não informam maiores detalhes sobre como é feito o cálculo das temperaturas.

Raif et al. (2009) desenvolveram uma abordagem modular de simulação dinâmica, baseada em SysML (*Systems Modeling Language*) para modelar o projeto de um satélite pequeno (nano-satélite), com relação à composição do sistema e seu comportamento dinâmico. O método descrito é utilizado para prever o desempenho do sistema na fase de design inicial do projeto. Portanto, utiliza modelos simples para limitar a complexidade. O método é modular e

adaptável, podendo ser utilizado em fases posteriores. O objetivo é construir uma biblioteca de módulos, a partir da qual o projeto do satélite pode ser montado e simulado.

O comportamento térmico descreve a distribuição de temperatura no interior do satélite. Como eles queriam evitar a utilização de equações diferenciais parciais, o número de nós foi limitado, minimizando o esforço computacional. Cada nó foi caracterizado por sua temperatura (estado) e a capacidade térmica.

No passado, a abordagem de simulação térmica utilizada tradicionalmente no ESOC para simuladores em tempo real era a interpolação (PERPIÑÁN, 1994), geralmente usando redes de aquecedores (*heaters*) e termistores. Essa escolha se deve à velocidade de computação e ao fato de não ser necessária uma grande precisão. No entanto, ele afirma que o poder computacional disponível aumentou de tal forma que os métodos de integração não são mais proibitivos. No artigo, essas duas metodologias de simulação são comparadas, em termos de eficiência, precisão, flexibilidade e facilidade da construção do modelo, e são apontadas as vantagens e desvantagens dessas duas metodologias, como mostrado na Tabela 2.2.

Tabela 2.2. Vantagens e desvantagens das abordagens de Integração e Interpolação.

Abordagem	Vantagens	Desvantagens
Integração	alta precisão, válido também para situações não nominais (ex: falhas)	consome tempo de execução e o desenvolvimento do modelo térmico é uma tarefa complexa que requer perícia e tempo
Interpolação	tempo de computação bem curto, a computação dos estados estacionários é fácil, o modelo é simples e a sua implementação também	baixa precisão, extrapolação irá produzir previsões incertas, é necessário armazenar tabelas que podem ficar muito grandes se houver muitos contextos

Fonte: Perpiñán (1994).

A implementação do modelo consiste de uma rede feita por dois tipos de nós: *heaters*, que são elementos que produzem ou absorvem calor; e termistores, elementos que medem a temperatura. O número de termistores é da ordem de 100.

Em um primeiro passo, o conjunto de aquecedores e termistores, juntamente com seus coeficientes, são extraídos de um modelo detalhado do satélite. Estes dados são então ajustados por um engenheiro térmico, por meio de um procedimento de tentativa e erro até que os resultados sejam aceitáveis. O modelo final fica disponível, geralmente, em um estágio bem avançado do ciclo de vida do simulador. Até então, os desenvolvedores de software trabalham com suposições. Em sua conclusão, ele propõe o uso de um método híbrido: integração numérica de um modelo térmico reduzido com a interpolação de uma tabela para os acoplamentos radiativos e fluxos externos. Por fim, ele afirma que essa metodologia irá melhorar a precisão da telemetria térmica gerada.

Mais recentemente, Manon et al. (2018) apresentaram um novo conceito de simulação térmica desenvolvido na agência espacial francesa CNES (*Centre National d'Etudes Spatiales*) para ser aplicado em simuladores de veículos espaciais.

O objetivo da solução investigada, denominada de Método de Caracterização Térmica de Satélites, é calcular os fluxos externos e as condutâncias radiativas sem recorrer a métodos intensivos, mas mantendo uma precisão da mesma ordem de magnitude desses métodos.

Inicialmente, é realizado um estágio preparatório para configurar o simulador. Este passo é realizado apenas uma vez utilizando o software THERMICA (um pacote comercial de análise térmica; AIRBUS, 2018) e envolve as seguintes ações:

- I. Migração do modelo térmico nodal satélite para modelos de nós integráveis no simulador;

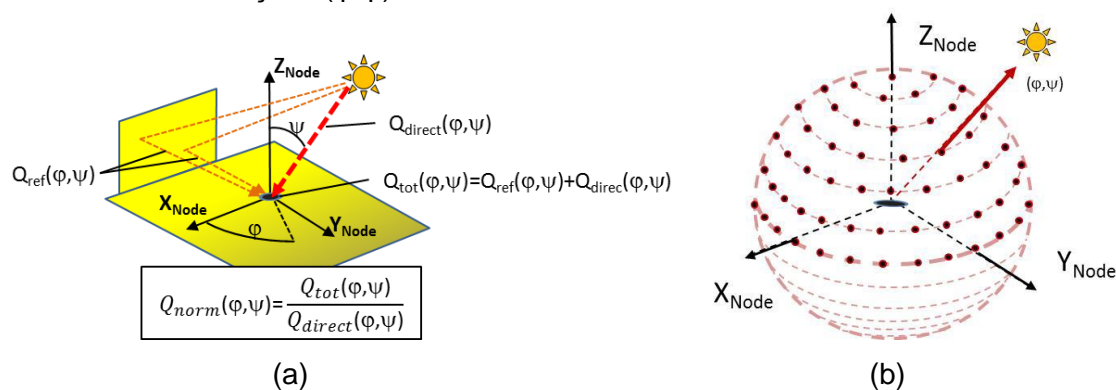
- II. Caracterização na forma de tabelas de coeficientes das trocas radiativas de calor entre nós, fontes e dissipadores térmicos do sistema que inclui satélite e ambiente.

Essas tabelas de coeficientes são específicas para cada satélite e são independentes de quaisquer cenários de simulação (posição orbital, atitude, dissipação de calor de peças de equipamentos, programação do satélite).

O fluxo total de calor recebido Q_{tot} é calculado de forma precisa pelo THERMICA para cada nó do modelo e cada fonte definida por dois ângulos (φ, ψ) em coordenadas esféricas em referência a cada nó. Esse cálculo inclui não só o fluxo direto Q_{direct} mas também o fluxo refletido Q_{ref} , correspondente a todos os fenômenos como reflexão múltipla, sombras e inclusive as propriedades especulares e a forma do nó (plano, esférico ou cilíndrico). O fluxo total é normalizado a partir da divisão pelo fluxo direto, ficando independente da potência da fonte e do próprio Q_{direct} . O fluxo normalizado Q_{norm} representa então um fator de escala que inclui todos fenômenos já citados (Fig. 2.3a).

Essa operação é realizada sobre todas as direções (φ, ψ) de forma a varrer uniformemente o espaço representado por uma esfera centrada no nó (Fig. 2.3b). Dessa forma, é obtida uma tabela de fluxos normalizados que fornecem os coeficientes de troca de calor em função da fonte (Sol ou Terra).

Figura 2.3. (a) Representação do fluxo radiativo incidente em um nó. (b) Discretização das direções (φ, ψ) de uma fonte radiativa.



Fonte: Manon (2018).

A atitude e a posição orbital do satélite podem ser determinadas a cada passo de integração na simulação, e então a direção da fonte radiativa em relação ao nó pode ser deduzida e, conseqüentemente, o fluxo direto. O cálculo do fluxo direto depende apenas da potência da fonte e das características físicas do nó (área, absortividade, etc.). A partir da determinação do valor do fluxo normalizado por interpolação na tabela da fonte em questão, pode ser deduzido o valor do fluxo total recebido.

Uma vez que os modelos de nós térmicos tenham sido incorporados no simulador e configurados com as tabelas dos coeficientes, o simulador pode ser executado para qualquer tipo de cenário de missão para calcular as temperaturas e o seu comportamento dinâmico com grande velocidade e precisão equivalente ao THERMICA.

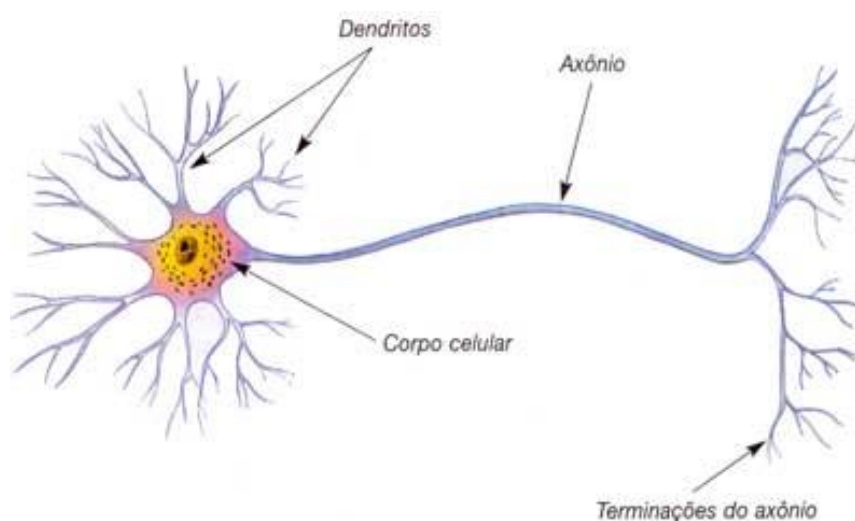
Para fins de validação, o método foi aplicado em um modelo de um CubeSat com cerca de 30 nós térmicos. Para esse modelo, observaram ganhos de cerca de 150 vezes no tempo de execução e erro máximo de 5K (ou 5°C) nas temperaturas geradas, em comparação com o THERMICA.

2.3 Redes Neurais Artificiais

Na sua forma mais geral, uma Rede Neural Artificial (RNA) é uma máquina virtual desenhada para modelar a forma como o cérebro biológico executa uma tarefa particular ou função de interesse (HAYKIN, 2001). No momento do nascimento, o cérebro humano possui uma extensa estrutura e a habilidade de desenvolver suas próprias regras por meio do que é usualmente denominado “experiência”. Na prática, a experiência vai sendo acumulada com o tempo, sendo que o desenvolvimento mais drástico do cérebro humano (por ligações físicas) ocorre durante os dois primeiros anos de vida, embora esse desenvolvimento continue para muito além desse estágio.

A Figura 2.4 ilustra a forma geral de um neurônio natural. Os dendritos são as conexões pelas quais os impulsos elétricos chegam ao neurônio. Esses impulsos são combinados no corpo celular, que é o núcleo da célula, e são transmitidos pelo axônio até suas terminações, também chamadas de sinapses. Essas terminações, por sua vez, podem se conectar a diversos outros neurônios.

Figura 2.4. Forma geral de um neurônio natural.



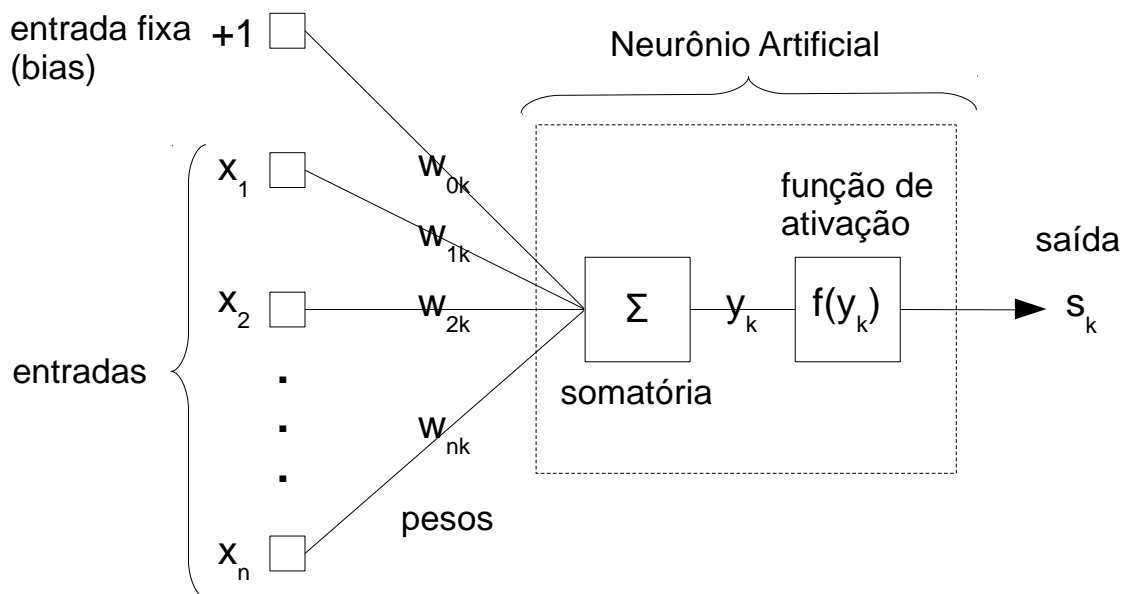
Fonte: SOGAB (2015).

Para atingirem seu objetivo, as redes neurais artificiais empregam uma vasta interconexão de células computacionais simples ou unidades de processamento, chamadas de neurônios. Em outras palavras, uma rede neural é um processador paralelo distribuído que possui a tendência natural de armazenar conhecimento experiencial e torná-lo disponível para uso (ROSENBLATT, 1958).

Um neurônio artificial é a unidade de processamento de informação fundamental para a operação de uma rede neural. A Fig. 2.5 ilustra o modelo de um neurônio artificial. Nesse diagrama, podem ser identificados três elementos básicos do modelo neuronal: um conjunto de sinapses ou elos de conexão, cada um caracterizado por uma intensidade própria, chamada de

peso sináptico (w); uma entrada fixa, chamada de bias; uma junção somatória, que realiza uma combinação linear dos sinais de entrada (x) com os pesos sinápticos; e uma função de ativação (f) que restringe a amplitude da saída do neurônio a um valor finito.

Figura 2.5. Modelo de um neurônio artificial.



Fonte: Produção do autor.

As equações 2.1 e 2.2 definem como é calculada a combinação linear na junção (y) e a saída do neurônio (s):

$$y_k = \sum_{j=0}^m w_{kj} x_j \quad (2.1)$$

$$s_k = f(y_k) \quad (2.2)$$

A função de ativação (f) pode ser qualquer função que limite os valores da saída entre 0 e 1 ou entre -1 e 1, como a função limiar ou de Heaviside, a função sigmoide ou a função tangente hiperbólica (HAYKIN, 2001). Além disso,

mais recentemente está sendo utilizada a função de ativação ReLU (*Rectified Linear Unit*), que não é limitada por assíntotas horizontais.

O modelo de neurônio com a função limiar ou de Heaviside é também chamado de modelo de McCulloch-Pitts (1943), ou simplesmente de neurônio MCP, e pode assumir apenas os valores de saída 0 ou 1.

O primeiro modelo de rede neural foi proposto por Rosenblatt (1958 e 1962) para classificação de padrões linearmente separáveis. Esse modelo, também chamado de Perceptron, consiste de um ou mais neurônios com pesos sinápticos ajustáveis, como o ilustrado na Fig. 2.5.

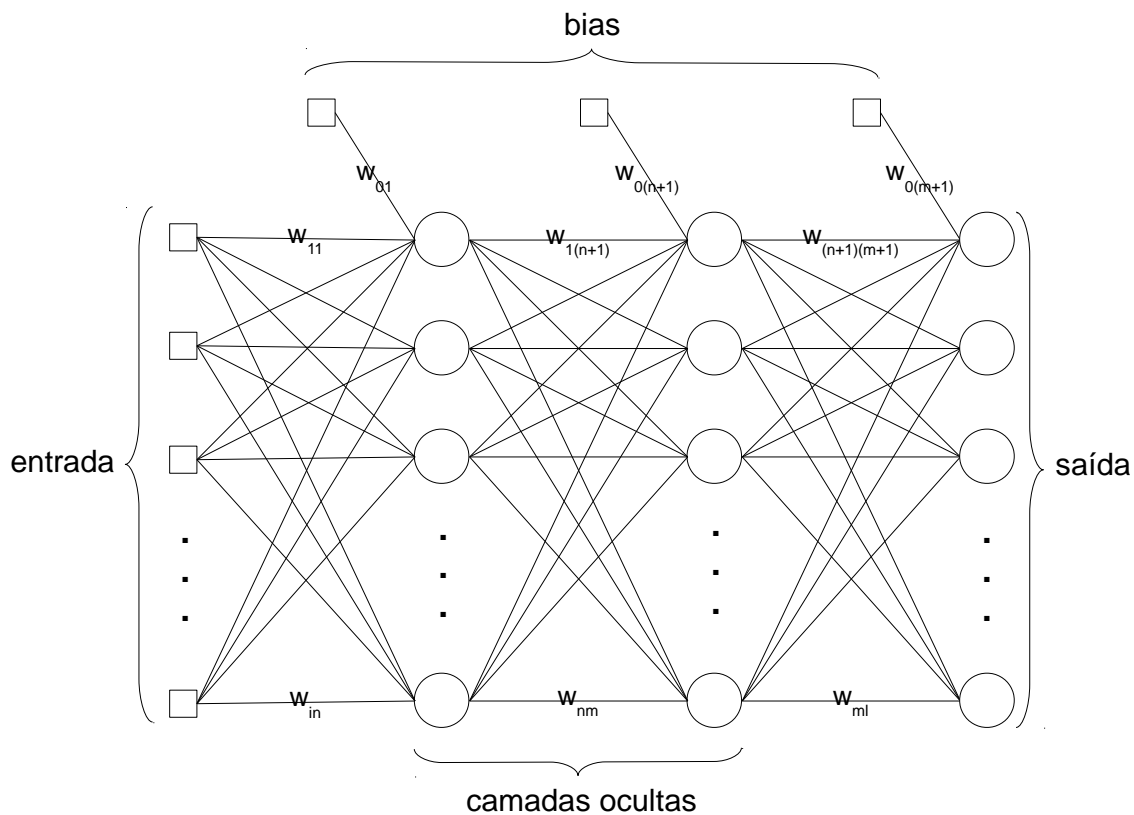
As redes neurais modernas são formadas por uma combinação de neurônios formando o que é chamada de arquitetura. Basicamente, podem ser identificadas três classes de arquiteturas de redes alimentadas para frente (do inglês *feedforward*): redes com uma única camada, com múltiplas camadas e redes recorrentes. A Fig. 2.6 ilustra uma rede de múltiplas camadas, chamada de *Multi-Layer Perceptron* (MLP), ou perceptron de múltiplas camadas (MINSKY; PAPERT, 1969). Redes MLP possuem uma ou mais camadas internas de neurônios artificiais, chamadas de camadas ocultas, pois não têm contato direto com o meio externo à rede. Os círculos na figura representam os neurônios artificiais.

O conhecimento é adquirido pela rede por meio de um processo de treinamento, no qual uma série de dados, também chamados de exemplos de treinamento, é apresentada à rede. Esses exemplos podem ser rotulados ou não rotulados.

Nos exemplos rotulados, cada sinal ou valor de entrada é associado a uma resposta desejada correspondente. Esse processo de treinamento, com exemplos rotulados, é chamado de Aprendizagem Supervisionada. Esta é a forma de aprendizagem utilizada neste trabalho.

De outro lado, os exemplos não rotulados consistem em diferentes ocorrências dos próprios sinais de entrada. Neste caso, chamada de Aprendizagem Não-Supervisionada. Outro paradigma também bastante conhecido sobre aprendizagem de máquina é a Aprendizagem por Reforço.

Figura 2.6. Arquitetura geral de uma rede neural MLP.



Fonte: Produção do autor.

De uma forma simplificada, o processo de treinamento de uma rede neural com exemplos rotulados consiste em inserir na rede um valor de entrada que é transmitido pelos neurônios, por meio do cálculo das Eqs. 2.1 e 2.2, até produzir um valor de saída. Esse valor de saída é então comparado com a resposta desejada e o erro associado é calculado. Esse erro, por sua vez, é usado para atualizar os pesos sinápticos da rede, de forma que a saída se aproxime mais do valor desejado. Em seguida, é apresentado o exemplo seguinte e assim por diante, até que o último exemplo de treinamento seja apresentado à rede. Caso o erro médio total seja maior do que um determinado valor, chamado de tolerância, então todo o conjunto de treinamento é novamente apresentado à rede. Este procedimento se repete até que o erro

médio total seja menor do que o valor de tolerância especificado. Neste momento, a rede é considerada treinada.

Esse processo de aprendizagem permite que a rede possa não só reproduzir o conhecimento adquirido, mas também generalizar, isto é, produzir saídas razoáveis para entradas não fornecidas durante o treinamento.

As redes neurais também possuem a vantagem de fornecer resultados muito rapidamente após o treinamento, tendo em vista que após este processo, o "conhecimento" fica armazenado em sua estrutura e nos seus parâmetros. No entanto, na prática, as redes neurais não podem prover a solução trabalhando por elas mesmas. Ao invés disso, elas precisam estar integradas em uma abordagem consistente, o que exige um grande esforço de teste, adaptação, verificação e validação dos procedimentos e resultados.

Uma das principais propriedades das RNAs é a capacidade de aproximar qualquer função matemática não linear (CYBENKO, 1988, 1989; HORNIK et al., 1989; COTTER, 1990). Este aspecto é útil particularmente quando as relações entre as variáveis não é conhecida ou é muito complexa para se descrever matematicamente. É possível distinguir entre duas classes principais de problemas de aproximação de funções:

- I. funções alvo conhecidas – consiste em um ramo da análise numérica que estuda como certas funções conhecidas podem ser aproximadas por uma classe específica de funções, como, por exemplo, funções polinomiais ou racionais, que apresentam propriedades desejáveis, como cálculo pouco dispendioso, continuidade, etc.;
- II. funções alvo desconhecidas – que, ao invés de uma fórmula explícita, possui-se apenas um conjunto de pontos na forma $[x, y(x)]$. Se y é uma operação no conjunto de números reais, técnicas como interpolação, extrapolação, regressão ou ajuste de curva, podem ser utilizadas. Por outro lado, se o conjunto alvo é finito, trata-se de um problema de classificação.

Os problemas da segunda classe consistem no mapeamento entrada-saída que, conforme citado anteriormente, é uma das principais propriedades das

RNAs. De acordo com Cybenko (1989) e também com Hornik et al. (1989), sempre existe uma rede neural de três camadas (entrada, oculta e saída) capaz de estimar uma função não linear arbitrária.

O modelo de rede neural mais comumente utilizado para essa tarefa é o MLP (Haykin, 2001). No projeto de redes MLP, é construído um modelo não-linear do fenômeno físico responsável pela geração dos exemplos de entrada e saída utilizados para treinar a rede, que é justamente o que é buscado no presente projeto.

Apesar da popularidade, a abordagem tradicional de treinamento de redes MLP, baseada no gradiente, apresenta algumas deficiências, como a tendência em ficar aprisionada em mínimos locais não desejáveis ao tentar alcançar o mínimo global em um espaço de busca complexo.

Para produzir uma melhora significativa no desempenho de convergência de um perceptron de múltiplas camadas, pode ser utilizada informação de ordem mais elevada no processo de treinamento. Isso pode ser obtido ao utilizar uma aproximação quadrática da superfície de erro em torno do ponto atual.

O método de Newton realiza essa aproximação por meio da matriz Hessiana invertida. A Hessiana é uma matriz quadrada constituída por derivadas de segunda ordem. Se a função de custo (erro médio) for quadrática, o método de Newton converge para a solução ótima em apenas uma iteração. No entanto, se a função custo não for quadrática, não há garantia de convergência. Além disso, o cálculo da matriz Hessiana invertida pode ter um alto custo computacional. Por essas razões, o método de Newton não é adequado para o treinamento da MLP.

Para superar algumas dessas dificuldades, pode ser usado um método quasi-Newton, que requer apenas uma estimativa do vetor gradiente, sem necessitar da inversão da matriz Hessiana. Por meio dessa estimativa, a descida na superfície de erro é garantida. Contudo, o custo computacional ainda é alto, o que dificulta a utilização desta abordagem para redes neurais de larga escala.

Outro método pertencente à classe dos métodos de otimização de segunda ordem é o método do gradiente conjugado, cuja principal propriedade é minimizar uma função quadrática sobre um espaço vetorial linear que é expandido progressivamente a cada iteração, até eventualmente incluir o mínimo global dessa função. Esta abordagem permite determinar a direção de busca sem o conhecimento específico da matriz Hessiana, o que torna este método mais interessante do que o quasi-Newton em termos computacionais.

O método de Levenberg-Marquardt (LM; LEVENBERG, 1944; MARQUARDT, 1963) combina os métodos de Newton e do Gradiente Descendente (GD). Quando os valores dos parâmetros estão distantes dos valores ótimos, o método atua como o GD. Por outro lado, quando os parâmetros estão próximos dos seus valores ideais ou, em outras palavras, conforme a RNA se aproxima do mínimo, o algoritmo alterna gradualmente para o método de Newton.

Hagan e Menhaj (1994) compararam a aplicação do método LM com os métodos do gradiente conjugado e com um algoritmo com taxa de aprendizagem variável. Esses métodos foram testados em diversos problemas de aproximação de funções. Eles concluem que o método LM é muito mais eficiente que as duas outras técnicas, especialmente quando é necessária uma alta precisão.

As RNAs vêm sendo aplicadas em uma grande variedade de campos diferentes e diversos algoritmos e meta-heurísticas vem sendo utilizados para aprimorar o seu treinamento e, conseqüentemente, os seus resultados (OJHA et al., 2017).

Uma meta-heurística é uma estratégia de busca, não específica para um determinado problema, que tenta explorar de forma eficiente o espaço das soluções viáveis desse problema. Em outras palavras, são algoritmos aproximados que incorporam mecanismos para evitar confinamento em mínimos ou máximos locais (NETO, 2016).

Na aplicação de algoritmos e meta-heurísticas é importante considerar um balanço entre busca global e busca local. Em inglês é comum o uso dos termos

exploration e *exploitation*, para se referir a essa relação. Como não existem termos diretamente análogos em português, o termo *exploration* pode ser referido como exploração diversificada (ou diversificação) e o termo *exploitation* como exploração focada (ou intensificação).

O processo de busca global ou exploração diversificada está relacionado com a procura de áreas promissoras do espaço de busca de forma ampla. De outro lado, a busca local ou exploração focada se refere à investigação mais detalhada de uma região localizada.

Ojha et al. (2017) realizaram uma revisão abrangente, de cerca de duas décadas, relacionada com a aplicação de meta-heurísticas para otimização de redes neurais. O artigo inclui uma introdução aos algoritmos convencionais de otimização baseados em gradiente, uma análise de diversas meta-heurísticas, uma discussão sobre meta-heurísticas multiobjetivo e métodos de amostragem.

O artigo classifica as meta-heurísticas em três categorias básicas:

- I. Algoritmos baseados em solução única: partem de uma única solução (candidata), aprimorando essa solução buscando em torno da sua vizinhança, até que uma solução satisfatória seja obtida. Ex.: Recozimento Simulado, Busca Tabu, etc.
- II. Algoritmos baseados em populações: operam simultaneamente com múltiplas soluções candidatas, até encontrar uma solução quase ótima. Ex.: Algoritmos Genéticos, Otimização por Enxame de Partículas, Evolução Diferencial, etc.
- III. Algoritmos híbridos: uma combinação de duas ou mais meta-heurísticas com o objetivo de explorar de forma mais eficiente o espaço de busca e encontrar uma solução ótima global.

Algoritmos convencionais baseados em gradiente possuem o problema de convergirem para mínimos locais, porém são rápidos e eficientes nessas buscas locais. Por outro lado, as meta-heurísticas são boas para buscas globais, porém sofrem de convergência prematura. Por essa razão, a combinação dessas duas classes de otimização pode oferecer uma solução melhor para otimização das RNAs (OJHA et al., 2017).

Como a minimização do erro de treinamento não garante sozinha uma boa generalização, a otimização de uma rede neural pode ser vista sob a perspectiva de um problema multiobjetivo. O procedimento de otimização multiobjetivo envolve a otimização de duas ou mais funções objetivo simultaneamente.

O aspecto de generalização é essencial para a aplicação de redes neurais, pois mesmo que uma RNA apresente baixo erro de treinamento, ela pode não responder bem para dados não apresentados no treinamento. Por essa razão, a minimização do erro de generalização é tão importante quanto a do erro de treinamento.

Entre as abordagens indicadas para melhorar a capacidade de generalização de redes neurais estão a validação cruzada (uso de um conjunto de validação durante o treinamento para verificar a resposta de rede a dados não apresentados), a inclusão de um termo de regularização ao erro de treinamento (para evitar o crescimento dos valores dos pesos) e a redução da complexidade da rede (relacionada com o número de neurônios e conexões da rede) (HAYKIN, 2001; HAGAN, 2014).

Dessa forma, as abordagens multiobjetivo buscam otimizar o erro de treinamento em conjunto com outros objetivos adicionais para atingir uma melhor generalização.

De acordo com Lo et al. (1993), as Redes Neurais Recorrentes, são aproximadores universais de sistemas dinâmicos não lineares. As arquiteturas das redes recorrentes podem assumir diferentes formas como: modelo recorrente de entrada-saída, conhecido como NARMAX (*Nonlinear AutoRegressive Moving Average with exogenous inputs*); Modelo de Espaço de Estados; Perceptron de Múltiplas Camadas Recorrente; e Redes de Segunda Ordem (HAYKIN, 2001).

O modelo NARMAX fornece uma descrição de sistemas não lineares em termos de uma expansão funcional não linear de entradas, saídas e erros de previsão (CHEN et al. 1990). Chen et al. (1990) derivaram um algoritmo para a utilização do modelo NARMAX em redes neurais. O algoritmo foi aplicado em

alguns exemplos, incluindo uma série temporal de manchas solares e dados gerados a partir de um trocador de calor, e os resultados obtidos sugeriram que a abordagem poderia ser utilizada de forma efetiva para modelagem de sistemas não lineares. Carrara (1997) aplicou o modelo NARMAX com redes neurais para o controle de atitude de satélites com geometria variável. Os resultados demonstraram que o controle da atitude de satélites por meio de redes neurais é viável, e que as RNAs são robustas no que se refere a perturbações no sistema, desconhecimento de parâmetros e ausência de alguns sinais.

O uso de modelos mais simples e/ou com custo computacional reduzido em substituição de modelos mais complexos e/ou com custo computacional mais alto é muitas vezes referenciado na literatura como *surrogate modeling* ou modelos substitutos.

Yondo et al. (2018) realizaram uma revisão abrangente do estado da arte das técnicas mais comuns de *surrogate modeling* utilizadas no setor aeroespacial. Eles classificaram as diferentes técnicas em quatro grupos: modelos de ajuste de dados (que incluem as redes neurais artificiais), modelos de ordem reduzida, modelos de multi-fidelidade e modelos híbridos.

Sobre RNAs, os autores destacam que estas têm sido utilizadas largamente em aerodinâmica de aeronaves e se mostraram eficientes em vários contextos, porém alguns trabalhos apresentaram limitações quando lidaram com um grande número de variáveis de projeto. Entre as vantagens apontadas estão a redução da dimensionalidade e a análise de sensibilidade implícita. Quanto às desvantagens, foram apontadas a tendência de adaptação excessiva aos dados de treinamento (*overfitting*) e a natureza de “caixa preta” que previne a quantificação eficiente das incertezas e a avaliação da correlação entre as variáveis de entrada e as respostas.

Em outro trabalho recente, Freitag et al. (2018) utilizaram uma estratégia de *surrogate modeling* para realizar previsões de intervalos de campos de assentamento de superfície em tempo real durante a construção de túneis conduzida por máquinas. Eles combinaram o uso de redes neurais recorrentes

com a técnica *Proper Orthogonal Decomposition* para aproximar e prever os campos de assentamento de superfície calculados por um modelo de simulação de elementos finitos.

Durante a fase de construção do túnel, após a fase de treinamento da RNA para cenários selecionados, o modelo substituto foi utilizado em tempo real para realizar as previsões de assentamento em cada estágio de avanço do túnel para apoiar as decisões do controlador da máquina. Em comparação com a abordagem de otimização para análise de intervalo, o tempo computacional foi reduzido significativamente pela estratégia proposta. O que permite que ela seja utilizada para realizar previsões em tempo real para apoiar o controlador, já que ela permite a rápida investigação das consequências de certos parâmetros do processo nos assentamentos esperados nos estágios de escavação subsequentes.

Todas as técnicas citadas podem contribuir para a obtenção de melhores resultados com a aplicação de redes neurais. Porém, não existe um método único que seja melhor para todos os problemas. Por essa razão, é necessário adaptar e/ou combinar diferentes técnicas, de forma específica para cada problema.

2.3.1 Redes Neurais Aplicadas em Problemas de Engenharia Térmica

Com o aumento da complexidade dos problemas térmicos e com a necessidade lidar com questões como dinâmica, desempenho do sistema, otimização e controle, novas metodologias são necessárias para tratá-los. Entre essas metodologias, as redes neurais têm sido aplicadas com êxito em uma ampla variedade de problemas térmicos (YANG, 2008).

De acordo com Yang (2008), as RNAs têm a capacidade de reconhecer com precisão a relação inerente entre qualquer conjunto de entrada e saída sem um modelo físico, e ainda assim os resultados respondem por toda a física que

relaciona a saída à entrada. Em segundo lugar, a metodologia é inerentemente tolerante a falhas, devido ao grande número de unidades de processamento da rede que realizam um processamento paralelo massivo de dados. Além disso, sua habilidade de aprendizagem também permite que a metodologia se adapte a mudanças nos parâmetros.

Adicionalmente, as RNAs possuem muita flexibilidade para incorporar outras metodologias para melhorar ainda mais sua capacidade de lidar com complexidade adicional em problemas térmicos.

Yang (2008) analisou uma grande variedade de trabalhos relacionados com a aplicação de redes neurais em problemas de ciência e engenharia térmica em três grandes áreas de aplicação, a saber, problemas térmicos estacionários, modelagem térmica dinâmica e controle térmico adaptativo. Os atributos dos resultados das RNAs foram avaliados em termos da sua precisão e flexibilidade de uso, e também suas validações computacionais e experimentais. Também foram discutidos algoritmos híbridos nos quais as redes neurais representam um papel mais importante, juntamente com outras metodologias de inteligência artificial associadas como algoritmos evolucionários, lógica nebulosa (*fuzzy logic*), etc., para melhoria do desempenho das aplicações. Entre as deficiências na aplicação de redes neurais, destaca a necessidade de dados experimentais confiáveis para treinamento e a incerteza na escolha dos parâmetros livres da rede. Com a presença sempre constante de ruído, imperfeições e informação incerta em problemas do mundo real, ele argumenta que as RNAs sozinhas podem não fornecer as respostas completas para problemas complicados, devendo, neste caso, ser combinadas com outras metodologias do universo de inteligência artificial. Ele conclui que a aplicação de metodologias de redes neurais representa uma forma promissora para abordar e solucionar problemas térmicos difíceis.

Martínez-Heras e Donati (2004) investigaram os benefícios e as desvantagens em utilizar RNAs, no suporte ao processo de modelagem do

comportamento de veículos espaciais. Essa abordagem foi aplicada às missões CLUSTER e ROSETTA da ESA.

Na missão CLUSTER, os autores supuseram que um dos sensores térmicos falhou repentinamente, de forma que seus dados não estariam mais disponíveis. O objetivo então foi reconstruir os valores que este sensor teria gerado, considerando a disponibilidade de dados anteriores da telemetria gerada, e fazendo uso da redundância implícita da informação gerada pelos sensores restantes. É sabido que a temperatura em diferentes locais no interior do satélite é correlacionada de alguma forma. No entanto, essa correlação é difícil de descobrir e até de modelar matematicamente, considerando o satélite real.

Os dados utilizados dos sensores selecionados referem-se a 10 dias, em intervalos de 1 min, resultando no total de 15.338 amostras. A rede utilizada foi composta por uma camada de entrada, uma oculta e uma de saída. Contendo 13 neurônios na camada de entrada (referente aos 13 sensores em funcionamento) e apenas 1 neurônio na camada de saída, correspondendo ao sensor que falhou. Foram testados de 1 a 40 neurônios na camada oculta (a melhor foi a de 32).

No caso ROSETTA, foram usadas RNAs para prever as leituras de certos sensores térmicos importantes, em função da distância do Sol e da atitude. Para o treinamento, foram utilizados dados da simulação de Balanço Térmico/Vácuo Térmico. Também foi assumido, para simplificação, que todas as fontes internas de calor estavam constantemente disponíveis e apenas fatores externos afetavam o comportamento térmico do veículo, tais como a distância do Sol e a atitude. O tempo foi incluído como uma entrada adicional para permitir a aprendizagem da evolução térmica. As 4 entradas identificadas foram: distância do Sol, dois ângulos para determinar a atitude e tempo, considerando 250 sensores térmicos. Foram excluídos termistores que dependem claramente de outras fontes, como os localizados muito próximos aos propulsores, pois seria necessária informação sobre o status dos propulsores para que a previsão fosse precisa. A topologia utilizada foi 4:M:1 (4

valores de entrada, M neurônios intermediários e 1 saída), com M variando de 1 a 15. A melhor topologia depende do sensor individual que se quer prever, no entanto, os melhores resultados foram obtidos para $10 \leq M \leq 15$. O erro médio na previsão de dados nunca vistos foi de $\pm 5,56^\circ \text{C}$.

Eles concluem que as RNAs têm capacidade para replicar o comportamento do subsistema térmico com dados de telemetria reais ou simulados e, com design e topologia apropriados, é possível implementar modelos confiáveis, capazes de generalizar o comportamento do sistema considerado durante a sessão de treinamento.

Fang et al. (2012) empregaram redes neurais para construir um modelo de previsão para simular as mudanças na temperatura superficial de uma bateria de hidreto metálico de níquel (Ni-MH na sigla em inglês), do tipo utilizado em veículos elétricos, durante o carregamento sob várias temperaturas ambientais.

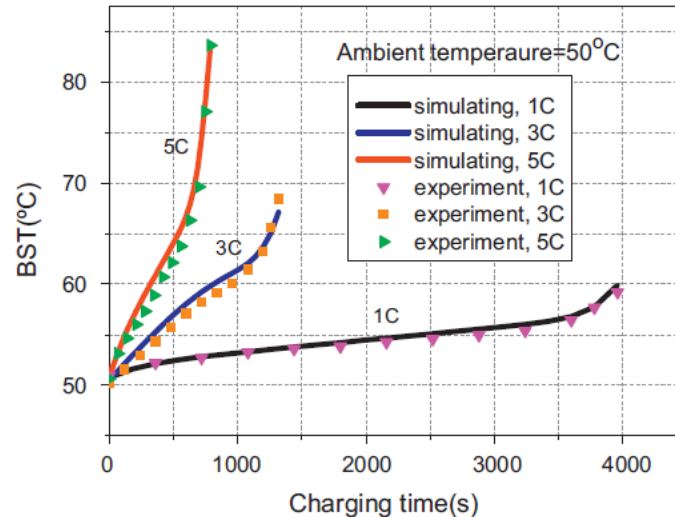
Em experimentos, é observado que a temperatura de superfície T_s é influenciada pela corrente de carregamento, temperatura ambiental e tempo de carregamento, se o calor dissipado para o ambiente for desconsiderado.

O modelo consiste em uma rede do tipo MLP com três camadas, sendo três entradas, 5 neurônios na camada oculta e um na saída. As variáveis de entrada são as três variáveis citadas acima e a saída da rede é a temperatura superficial da bateria. O treinamento foi realizado por meio do algoritmo de Levenberg-Marquardt.

Para obter os dados de entrada, a bateria foi carregada com diferentes valores de corrente e temperatura ambiental. As temperaturas foram medidas por um sistema imageador infravermelho. Para fins de segurança, os experimentos foram realizados para temperaturas abaixo de 50°C . Para temperaturas mais elevadas foi utilizado um modelo em MATLAB para estimar as temperaturas superficiais. Uma comparação entre os resultados simulados e os dados experimentais pode ser vista na Fig. 2.7.

Os resultados mostram que as curvas simuladas têm um bom acordo com os dados medidos, com uma diferença máxima de temperatura de $1,3^\circ\text{C}$, o que demonstra sua efetividade na simulação.

Figura 2.7. Comparação entre os resultados simulados e os dados experimentais, para a temperatura ambiente de 50°C.



Fonte: Fang et al. (2012).

Trocadores de calor são dispositivos que facilitam a transferência efetiva de calor entre dois materiais, em razão da sua diferença de temperatura, sendo utilizados em uma grande variedade de indústrias. A aplicação de RNAs para a análise térmica de trocadores de calor foi revisada por Mohanraj et al. (2015). Os estudos analisados foram caracterizados em quatro principais aplicações: modelagem, estimação dos parâmetros, estimação das características das mudanças de fase e controle dos trocadores de calor.

Redes MLP foram amplamente utilizadas para prever o desempenho de trocadores de calor, devido a sua simplicidade. No entanto, foram observadas limitações na otimização da configuração das redes. Neste caso, técnicas de otimização como Algoritmos Genéticos, Otimização por Enxame de Partículas, Recozimento Simulado, Algoritmo de Colônia de Formigas e formas híbridas foram recomendadas para otimizar as configurações. Sendo que as abordagens híbridas parecem ser as mais eficientes.

Poucas investigações abordaram outras arquiteturas como Redes de Funções de Base Radial e Redes Neurais de Regressão Generalizada devido a desvantagens na estimação e previsão de desempenho. As redes híbridas

ANFIS (*Adaptive Neuro Fuzzy Interface Systems*) apresentaram precisão aceitável para modelagem e simulação, mas carecem de mais estudos relacionados com controle e aplicações para diagnóstico de falhas.

A literatura avaliada confirmou que as RNAs podem ser usadas com sucesso para a modelagem, simulação e análise de trocadores de calor. A partir da revisão detalhada, Mohanraj et al. (2015) recomendam o uso de redes MLP com um algoritmo de retropropagação para a análise térmica de trocadores de calor.

3 ABORDAGEM PROPOSTA PARA A SIMULAÇÃO EM TEMPO REAL DO COMPORTAMENTO TÉRMICO DE SATÉLITES

Neste Capítulo, são descritos a formulação do problema, particularmente os modelos de veículos espaciais e as arquiteturas de redes neurais que foram utilizadas, bem como, os diferentes algoritmos e meta-heurísticas que foram aplicados para treinamento das redes neurais nesta Tese, como Gradiente Descendente, Levenberg-Marquardt e Otimização Extrema Generalizada.

3.1 Formulação do Problema

O problema geral a ser solucionado é como desenvolver um sistema computacional que seja capaz de fornecer o comportamento térmico de um satélite artificial em órbita da Terra, com tempo de processamento compatível com os requisitos de um Simulador Operacional e elevada fidelidade em comparação com o sistema real.

Para solucionar esse problema, é proposta a utilização de Redes Neurais Artificiais (RNAs) do tipo MLP, devido à sua capacidade de armazenar conhecimento, mapeando relações complexas entre dados de entradas e saídas, e generalizar a partir do conhecimento adquirido (RUMELHART et al., 1986). Além disso, as redes MLP recorrentes são aproximadores universais de sistemas dinâmicos não lineares (LO et al., 1993). Por fim, o baixo custo computacional para o processamento da RNA após o treinamento é uma característica fundamental para a escolha de aplicação em um simulador em tempo real.

De forma geral, a metodologia de trabalho consistiu em construir modelos tridimensionais, simular cenários orbitais, utilizar parte dos dados gerados para treinar as RNAs e, em seguida, comparar os resultados analisando a sua capacidade de reprodução dos dados utilizados no treinamento e também a

capacidade de generalização, comparando com o restante dos dados não aplicados no treinamento.

A modelagem 3D e os dados dos cenários orbitais foram gerados a partir do software *AutoCAD*, utilizando a extensão *Thermal Desktop* e o software SINDA/FLUINT (C&R TECHNOLOGIES, 2013). Essa extensão permite a criação de modelos tridimensionais com propriedades termo-físicas e ópticas (para emissão e absorção de radiação), além da inclusão de elementos térmicos como condutores, cargas térmicas, aquecedores, etc. O software SINDA/FLUINT (SINDA) é uma ferramenta para o cálculo da transferência de calor e modelagem do fluxo de fluidos em sistemas complexos, que permite a geração de análises e simulações térmicas, incluindo simulações orbitais.

Para gerar os dados de treinamento, primeiramente, foram calculados os estados de equilíbrio (*steady state*) e, em seguida, foram calculadas as temperaturas transientes ao longo de algumas órbitas para estabilizar as temperaturas. Por fim, os dados da última órbita foram extraídos para que pudessem ser utilizados para treinar a RNA.

Todas as implementações de redes neurais desta Tese foram realizadas utilizando a linguagem de programação C e para os treinamentos foi utilizada uma workstation com processador Intel Xenon com 8 núcleos.

De forma geral, no projeto de um subsistema de controle térmico, uma diferença de até 5°C em relação ao sistema real é considerada dentro dos padrões aceitáveis para a maior parte dos equipamentos. Por essa razão, foi estabelecido como requisito para as temperaturas geradas no Simulador Operacional um limite de 5°C para o erro em comparação com os dados gerados no SINDA.

3.2 Treinamento por Gradiente Descendente (GD)

O algoritmo de treinamento mais comumente utilizado para a aprendizagem supervisionada de redes MLP é o algoritmo de Gradiente Descendente,

também chamado de Descida Mais Íngreme, de Retropropagação do Erro ou simplesmente de Retropropagação (RUMELHART et al., 1986). Considerando um neurônio k sendo alimentado por um conjunto de sinais produzidos por uma camada de neurônios à sua esquerda, o campo local induzido y_k é dado pela Eq. 3.1 e a correção no peso w_{ji} é definida pela regra delta:

$$\Delta w_{ji} = \eta \delta_j y_i \quad (3.1)$$

onde η é a taxa de aprendizagem e o gradiente local é definido por:

$$\delta_j = e_j f'(y_j) \quad (3.2)$$

onde f' é a derivada da função de ativação e o erro e_j é dado pela diferença entre o valor desejado (d_j) e valor resultante do neurônio de saída (s_j):

$$e_j = d_j - s_j \quad (3.3)$$

O gradiente local aponta para as mudanças necessárias nos pesos sinápticos. Das Eqs. 3.1 e 3.2 nota-se que o sinal do erro é um fator chave no cálculo do ajuste do peso. Neste caso, é possível identificar duas situações distintas, dependendo da localização do neurônio na rede.

Na primeira situação, o neurônio é um nó de saída. Este caso é mais simples de tratar, pois cada neurônio de saída possui uma resposta desejada própria, de forma que basta calcular o erro associado, por meio da Eq. 3.3 e o gradiente local pela Eq. 3.2.

Na situação 2, o neurônio é oculto e, portanto, não possui uma resposta desejada específica. Mesmo não sendo acessados diretamente, os neurônios das camadas intermediárias compartilham responsabilidade pelo erro gerado na saída da rede. A questão então é como penalizar ou recompensar os neurônios ocultos por sua parcela de responsabilidade.

Neste caso, o sinal do erro para um neurônio oculto deve ser determinado recursivamente e trabalhando de trás para frente em termos dos erros de todos os neurônios aos quais ele está conectado diretamente. Dessa forma, o gradiente local pode ser redefinido como:

$$\delta_j = f'(y_j) \sum_k \delta_k w_{kj} \quad (3.4)$$

onde o índice k se refere aos neurônios conectados ao neurônio j pelo lado direito. O fator $f'(y_j)$ depende apenas da função de ativação associada ao neurônio oculto j . O fator restante (soma sobre k) depende do gradiente δ_k , que requer conhecimento sobre os sinais de erro e_k para todos os neurônios da camada imediatamente à direita que estão conectados diretamente ao neurônio j , e também dos pesos w_{kj} , associados a essas conexões.

A partir desse processo os pesos da rede podem ser adaptados com o objetivo de reduzir o erro na saída até um valor mínimo de tolerância pré-definido. Quando esse valor é atingido, a RNA é considerada treinada.

O treinamento pode ser realizado de duas formas diferentes. Em uma delas os padrões de entrada são inseridos na rede um a um e os pesos são adaptados para cada padrão (treinamento sequencial ou *online*). Na outra forma, os padrões também são inseridos um a um, porém os pesos são atualizados somente a partir do erro médio, que é calculado após a apresentação de todos os padrões de treinamento (treinamento em lote ou *batch*). Nos dois casos, após a inserção de todos os padrões, caso o erro não tenha convergido para o valor de tolerância, o processo de apresentação dos padrões é reiniciado. Cada ciclo de apresentação de todos os padrões de entrada é chamado de “época”.

O algoritmo de Gradiente Descendente, apesar de apresentar bons resultados para o treinamento de redes MLP, possui alguns problemas de convergência, sendo comum ficar preso em um mínimo local da superfície de erro. Além disso, existem diversos parâmetros livres, tais como o número de

camadas, o número de neurônios em cada camada, a taxa de aprendizagem, entre outros.

3.3 Treinamento por Levenberg-Marquardt (LM)

O método de Levenberg-Marquardt (LEVENBERG, 1944; MARQUARDT, 1963) pertence à categoria das técnicas de otimização de segunda ordem, assim como os métodos de Gradiente Conjugado e quasi-Newton.

O LM é um método de busca local, ou de exploração intensificada, que consiste em uma variação do método de Newton para minimizar funções que são somas de quadrados de outras funções não lineares e é indicado para o treinamento de redes neurais nas quais o índice de desempenho é o erro quadrático médio (HAGAN, 1994, 2014).

Considerando o método de Newton para otimizar um índice de desempenho $F(\mathbf{x})$, temos (as variáveis em negrito minúsculas representam vetores e as maiúsculas matrizes):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k, \quad (3.5)$$

$$\text{onde } \mathbf{A}_k \equiv \nabla^2 F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k} \text{ e } \mathbf{g}_k \equiv \nabla F(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}.$$

Se for assumido que $F(\mathbf{x})$ é uma função de soma de quadrados:

$$F(\mathbf{x}) = \sum_{i=1}^N v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad (3.6)$$

então o elemento j do gradiente seria

$$[\nabla F(\mathbf{x})]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j}. \quad (3.7)$$

O gradiente pode ser escrito na forma matricial da seguinte forma:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad (3.8)$$

onde

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (3.9)$$

é a matriz Jacobiana.

A seguir, é definida a matriz Hessiana, cujo elemento k,j pode ser definido como

$$[\nabla^2 F(\mathbf{x})]_{k,j} = \frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(\mathbf{x})}{\partial x_k} \frac{\partial v_i(\mathbf{x})}{\partial x_j} + v_i(\mathbf{x}) \frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \right\}. \quad (3.10)$$

A matriz Hessiana pode então ser expressa na forma matricial:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}), \quad (3.11)$$

onde

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}). \quad (3.12)$$

Se for considerado que $\mathbf{S}(\mathbf{x})$ é pequeno, a matriz Hessiana pode ser aproximada como

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}). \quad (3.13)$$

Se então as Eqs. 3.13 e 3.8 forem substituídas na Eq. 3.5, é obtido o método de Gauss-Newton:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - [2\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} 2\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k) \\ &= \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k). \end{aligned} \quad (3.14)$$

A vantagem do método de Gauss-Newton sobre o método de Newton padrão é que o primeiro não requer o cálculo de derivadas de segunda ordem. Porém, um problema com o método de Gauss-Newton é que a matriz $\mathbf{H}=\mathbf{J}^T\mathbf{J}$ pode não ser inversível. Isso pode ser superado utilizando a seguinte modificação à matriz Hessiana aproximada:

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I}. \quad (3.15)$$

onde \mathbf{I} é a matriz identidade e μ é um parâmetro ajustável.

Isso leva ao algoritmo Levenberg-Marquardt:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k). \quad (3.16)$$

Este algoritmo possui a útil característica de que com o aumento de μ_k ele se aproxima do algoritmo de Gradiente Descendente com taxa de aprendizado pequena. Enquanto que, com a redução de μ_k o algoritmo se torna o Gauss-Newton.

O algoritmo é iniciado com μ_k definido com um valor baixo (p.ex. 0,1). Se um passo não produz um valor menor para $F(\mathbf{x})$, então o passo é repetido com μ_k multiplicado por um fator $\nu > 1$. Eventualmente, $F(\mathbf{x})$ deverá decrescer, pois está sendo dado um passo pequeno na direção da descida mais íngreme. Se, por outro lado, um passo produz um valor menor para $F(\mathbf{x})$, então μ_k é dividido por ν para o próximo passo, de forma que o algoritmo irá aproximar o Gauss-Newton, o que deveria prover uma convergência mais rápida.

Para aplicar o algoritmo ao problema de treinamento de redes neurais o índice de desempenho deve ser definido como o erro quadrático médio:

$$F(\mathbf{x}) = \sum_{p=1}^{N^p} \sum_{j=1}^{N^s} (e_{p,j})^2, \quad (3.17)$$

onde $e_{p,j}$ é o elemento j do erro para o padrão de entrada p , N^p é o número de padrões de treinamento e N^s é o número de neurônios na camada de saída.

O passo mais importante no Levenberg-Marquardt é o cálculo da matriz Jacobiana. Para realizar esse cálculo é usada uma variação do algoritmo de Gradiente Descendente. Neste último, as derivadas são calculadas sobre os erros quadráticos, com relação aos pesos e *biases* da rede. Enquanto que para construir a matriz Jacobiana é necessário calcular as derivadas dos erros, ao invés das derivadas dos erros quadráticos.

Após a realização das substituições adequadas, a matriz Jacobiana para o treinamento de redes MLP pode ser escrito como:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_1} & \frac{\partial e_{1,1}}{\partial w_2} & \dots & \frac{\partial e_{1,1}}{\partial w_{N^w}} \\ \frac{\partial e_{1,2}}{\partial w_1} & \frac{\partial e_{1,2}}{\partial w_2} & \dots & \frac{\partial e_{1,2}}{\partial w_{N^w}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{1,N^s}}{\partial w_1} & \frac{\partial e_{1,N^s}}{\partial w_2} & \dots & \frac{\partial e_{1,N^s}}{\partial w_{N^w}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{2,1}}{\partial w_1} & \frac{\partial e_{2,1}}{\partial w_2} & \dots & \frac{\partial e_{2,1}}{\partial w_{N^w}} \\ \frac{\partial e_{2,2}}{\partial w_1} & \frac{\partial e_{2,2}}{\partial w_2} & \dots & \frac{\partial e_{2,2}}{\partial w_{N^w}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_{N^p,N^s}}{\partial w_1} & \frac{\partial e_{N^p,N^s}}{\partial w_2} & \dots & \frac{\partial e_{N^p,N^s}}{\partial w_{N^w}} \end{bmatrix}, \quad (3.18)$$

onde N^w é o número de pesos da rede e N^p o número de padrões de treinamento.

Em resumo, a matriz Jacobiana é construída da seguinte forma:

- I. uma linha para cada neurônio na camada de saída (N^s);
- II. uma coluna para cada peso na rede (N^w);
- III. para cada padrão de treinamento (N^p) os passos i e ii são repetidos.

Dessa forma, a matriz Jacobiana completa tem como dimensões N^w colunas por $N^p \times N^s$ linhas.

Os termos dessa matriz podem ser computados por um processo semelhante ao definido para o algoritmo de Gradiente Descendente na Seção 3.2, apenas com algumas modificações importantes.

Primeiramente, no algoritmo de Gradiente Descendente apenas um processo de retropropagação é necessário. Enquanto no Levenberg-Marquardt o processo de retropropagação deve ser repetido para cada saída separadamente. Além disso, no LM os parâmetros δ são calculados para cada neurônio j e cada saída k .

Se o neurônio j é um neurônio de saída, tem-se que:

$$\delta_{k,j} = \begin{cases} -f'(y_j), & j = k \\ 0, & j \neq k \end{cases} \quad (3.19)$$

onde k indica qual saída está sendo retropropagada.

Já para as camadas ocultas é usada a Eq. 3.4, da mesma forma como no algoritmo de Gradiente Descendente.

Dessa forma, os elementos da matriz Jacobiana podem ser calculados como:

$$\frac{\partial e_{p,k}}{\partial w_i} = -\delta_k y_i \quad (3.20)$$

Por fim, os pesos da rede devem ser atualizados pela Eq. 3.16 que, adaptada para as redes neurais, pode ser escrita como:

$$\Delta \mathbf{w} = -[\mathbf{J}^T \mathbf{J} + \mu_i \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}. \quad (3.21)$$

3.4 Treinamento por Otimização Extrema Generalizada (GEO)

O algoritmo de Otimização Extrema Generalizada (Generalized Extremal Optimization, GEO), proposto por De Sousa e Ramos (2002), foi desenvolvido para ser utilizado em problemas de otimização complexos nos quais os métodos tradicionais baseados em gradiente muitas vezes não são eficientes.

Pertencente à classe dos algoritmos evolutivos, assim como o Algoritmo Genético, o GEO consiste em uma generalização do método de Otimização Extrema (BOETTCHER; PERCUS, 2001), baseado no princípio da seleção natural, porém sem utilizar a estrutura de reprodução de populações do Algoritmo Genético. O GEO, assim como o método de Otimização Extrema, tem como inspiração o modelo simplificado de evolução natural proposto por Bak e Sneppen (1993), desenvolvido com o objetivo de demonstrar que ecossistemas naturais podem ser descritos por um estado de Criticalidade Auto-Organizada.

O GEO é uma meta-heurística de busca global ou de exploração diversificada, que não utiliza derivadas e pode ser utilizado em problemas com ou sem restrições, em espaços não convexos ou desconexos e na presença de qualquer combinação de variáveis contínuas, discretas ou inteiras (DE SOUSA, 2003).

No GEO canônico, L espécies são alinhadas lado a lado e para cada espécie é atribuído um valor de adaptabilidade que irá definir qual espécie é propensa a sofrer mutação. A cada espécie é associado um bit, que pode assumir o valor de 0 ou 1. De forma que toda a população consiste de uma cadeia binária, na qual são codificadas as variáveis de projeto.

O número de adaptabilidade, associado a cada espécie, é proporcional ao ganho (ou perda) que o valor da função objetivo tem ao mutar (trocar) o bit. Todos os bits são então ranqueados de 1, para o bit menos adaptado, a N para o mais adaptado. Um bit é então trocado (de 0 para 1 ou de 1 para 0) de acordo com a distribuição de probabilidade (P). Esse processo é repetido até que um dado critério de parada é alcançado e a melhor configuração de bits encontrada (a que resulta no melhor valor para a função objetivo) é retornada.

A implementação prática do algoritmo GEO para um problema de otimização é a seguinte:

- I. Inicializar aleatoriamente uma cadeia binária de comprimento L , que codifica N variáveis de projeto de comprimento l_j , $j = 1, N$. Para a configuração inicial de bits C , calcular o valor da função objetivo V e definir $C_{\text{melhor}} = C$ e $V_{\text{melhor}} = V$.
- II. Para cada bit i da cadeia, em uma dada iteração: a) trocar o bit e calcular o valor da função objetivo V_i da configuração C_i ; b) definir a adaptabilidade do bit como $\Delta V_i = (V_i - V_{\text{melhor}})$, que indica o ganho (ou perda) relativo ao mutar o bit, comparado com o melhor valor da função objetivo encontrado até o momento; e c) retornar o bit ao seu valor original.
- III. Ranquear os bits de acordo com os seus valores de adaptabilidade, de $k = 1$, para o menos adaptado, até $k = L$ para o melhor adaptado. Caso dois ou mais bits tenham a mesma adaptabilidade, os mesmos devem ser ranqueados aleatoriamente com distribuição uniforme.
- IV. Escolher com probabilidade uniforme um bit candidato i para mutar. Gerar um número aleatório r com distribuição uniforme no intervalo $[0,1]$. Se a probabilidade

$$P_i(k) = k^{-\lambda} \quad (3.22)$$

for igual ou maior que r , o bit é mutado definitivamente. Caso contrário, escolher um novo bit candidato e repetir o processo até que um bit seja mutado.

- V. Definir $C = C_i$ e $V = V_i$, onde i é o bit mutado no passo IV.
- VI. Se $V < V_{\text{melhor}}$ ($V > V_{\text{melhor}}$, para um problema de maximização), então definir $V_{\text{melhor}} = V$ e $C_{\text{melhor}} = C$.
- VII. Repetir os passos II a VI até que um dado critério de parada seja alcançado.
- VIII. Retornar V_{melhor} e C_{melhor} .

No passo IV, qualquer bit pode ser escolhido para mutar, mas a probabilidade de que um dado bit seja mutado depende da sua posição no ranque (k). De acordo com a Eq. 3.22, os mais adaptados (com valores de ranque maiores) são menos propensos a serem mutados e apenas o bit menos adaptado ($k = 1$) sempre será mutado, caso seja escolhido. A probabilidade de mutar o bit escolhido é regulada pelo parâmetro ajustável λ . Quanto maior o valor de λ , menor a probabilidade que um bit (com $k > 1$) seja mutado. A possibilidade de realizar mutações que não melhoram o valor da função objetivo é o que permite que o algoritmo escape de mínimos locais.

O número de bits necessários para representar cada variável deve ser definido de acordo com a precisão desejada. Para variáveis contínuas, o número mínimo m de bits necessários para atingir determinada precisão é dado por

$$2^m \geq \left[\frac{(x_j^u - x_j^l)}{p} + 1 \right], \quad (3.23)$$

onde x_j^l e x_j^u são os limites inferior e superior, respectivamente, da variável j , com $j = 1, N$ e p a precisão desejada.

O valor físico de cada variável é obtido por meio da equação

$$x_j = x_j^l + (x_j^u - x_j^l) \left[\frac{I_j}{(2^{I_j} - 1)} \right], \quad (3.24)$$

onde l_j é o número inteiro obtido na transformação da variável j da sua forma binária para uma representação decimal.

O algoritmo GEO será aplicado para o treinamento de redes neurais com o objetivo de simular o comportamento térmico de satélites em tempo real. Embora tenha sido aplicado em diversos problemas, o presente projeto é a primeira aplicação do algoritmo GEO para redes neurais.

No caso das RNAs, as variáveis de projeto são os pesos w da rede. Diferentemente do GEO canônico, no qual as variáveis são descritas em termos de bits, neste caso os pesos podem assumir qualquer valor real. Uma adaptação para valores reais já foi utilizada na literatura e foi chamada de GEO_{real} (LOPES et al., 2008).

Ao invés de simplesmente trocar o valor do bit, no GEO_{real} as variáveis são perturbadas por um valor aleatório definido a partir de uma distribuição normal (LEVA, 1992).

Adaptando o algoritmo para RNAs tem-se os seguintes passos:

- I. Inicializar os valores dos pesos w aleatoriamente. Para essa configuração inicial, calcular o erro quadrático médio E_m e definir $\mathbf{w}_{\text{melhor}} = \mathbf{w}$ e $E_{\text{melhor}} = E_m$.
- II. Para cada peso w_i , em uma dada iteração de padrão de treinamento: a) perturbar o peso com um valor escolhido aleatoriamente com distribuição normal e calcular o erro E_i da configuração de pesos \mathbf{w}_i (neste caso, E_i foi definido com sendo a adaptabilidade; e b) retornar o peso ao seu valor original.
- III. Ranquear os pesos de acordo com os valores do erro E_i , de $k = 1$, para o menos adaptado (menor valor de E_i), até $k = N^w$ (número de pesos) para o melhor adaptado (maior valor de E_i). Caso dois ou mais pesos tenham a mesma adaptabilidade, os mesmos são ranqueados aleatoriamente com distribuição uniforme.
- IV. Escolher, com probabilidade uniforme, um peso candidato w_i para mutar. Gerar um número aleatório r com distribuição uniforme no intervalo $[0,1]$. Se a probabilidade

$$P_i(k) = k^{-\lambda} \quad (3.25)$$

for igual ou maior que r , o peso é mutado definitivamente. Caso contrário, escolher um novo peso candidato e repetir o processo até que um peso seja mutado.

- I. Definir $\mathbf{w} = \mathbf{w}_i$ e $E_m = E_i$, onde i é o peso mutado no passo IV.
- II. Se $E_m < E_{\text{melhor}}$, então definir $E_{\text{melhor}} = E_m$ e $\mathbf{w}_{\text{melhor}} = \mathbf{w}$.
- III. Repetir os passos II a VI até que o erro seja inferior a um valor de tolerância.
- IV. Retornar E_{melhor} e $\mathbf{w}_{\text{melhor}}$.

Por ser um algoritmo de busca global, o GEO é muito eficiente na exploração diversificada. Porém, quando aplicado às redes neurais, o algoritmo necessita de uma grande quantidade de passos para atingir um valor mínimo. Por essa razão, nos treinamentos realizados neste projeto, o GEO foi combinado com o algoritmo do Gradiente Descendente. Em termos práticos, o treinamento é iniciado com o algoritmo GEO até que o erro médio seja menor do que um valor de tolerância especificado ou até que seja atingido um determinado valor limite de épocas. A partir desse ponto o treinamento é passado para o Gradiente Descendente para atingir o mínimo localizado mais rapidamente.

4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos durante o desenvolvimento da pesquisa.

Inicialmente, são retratados os estudos preliminares (Seção 4.1) realizados com o objetivo geral de verificar a capacidade das redes neurais de reproduzir as temperaturas de veículos espaciais a partir dos dados utilizados no treinamento. Esses estudos foram aplicados em três objetos distintos: uma esfera de alumínio (Subseção 4.1.1), um CubeSat (Subseção 4.1.2) e o satélite Amazonia-1 (Subseção 4.1.3), visando avaliar a escalabilidade da solução. Os primeiros resultados buscaram verificar a capacidade de generalização da rede, incluindo uma grande quantidade de cenários com valores intermediários para as variáveis de entrada e, sua capacidade de reproduzir o comportamento de 2 cenários aleatórios não treinados (Subseção 4.1.4).

Na Seção 4.2, é descrita uma arquitetura de rede (Subseção 4.2.1) com o objetivo principal de aprimorar a capacidade de generalização da RNA, mas que também traz benefícios adicionais que serão discutidos mais adiante. Essa arquitetura foi treinada com três algoritmos diferentes: Gradiente Descendente, Levenberg-Marquardt e Otimização Extrema Generalizada, cujos resultados são apresentados nas Subseções 4.2.2, 4.2.3 e 4.2.4, respectivamente. Na sequência, é realizada uma análise e comparação dos resultados para os três algoritmos (Subseção 4.2.5). Adicionalmente, na Subseção 4.2.6 são apresentados os resultados obtidos para o satélite Amazonia-1, utilizando a mesma metodologia aplicada ao CubeSat visando obter uma melhor generalização para cenários não treinados.

Na Subseção 4.3 os resultados obtidos foram comparados com duas metodologias diferentes encontradas na literatura: o Método de Caracterização Térmica de Satélites (MANON et al., 2018) e uma interpolação utilizando o método de Regressão Linear Multivariada (RLM).

Por fim, foi realizada uma análise do tempo necessário para a RNA simular o comportamento térmico do satélite Amazonia-1, em função do requisito de tempo de processamento do Simulador Operacional (Subseção 4.4).

4.1 Estudos Preliminares

O objetivo dos estudos preliminares aqui apresentados foi verificar a capacidade das redes neurais para reproduzirem as temperaturas de diferentes veículos espaciais de complexidade e escala crescentes.

4.1.1 Esfera de alumínio

Com o objetivo de obter uma prova de conceito, o trabalho inicial consistiu em reproduzir o comportamento térmico de uma casca esférica de alumínio em órbita da Terra, com 1 m de diâmetro e 1 cm de espessura. As propriedades termo-físicas utilizadas para modelar o alumínio foram: densidade 2710 kg/m^3 , condutividade térmica $168 \text{ W/m}^\circ\text{C}$ e calor específico $963 \text{ J/kg}^\circ\text{C}$. Com relação às propriedades ópticas, foi considerado que a esfera possuía células solares no seu exterior e que foi pintada com tinta preta no interior, conforme descrito na Tabela 4.1.

Tabela 4.1. Propriedades ópticas da esfera em órbita da Terra.

Material	Absortividade (α)	Emissividade (ϵ)	α/ϵ
Células Solares	0,90	0,80	1,125
Tinta Preta	0,95	0,87	1,092

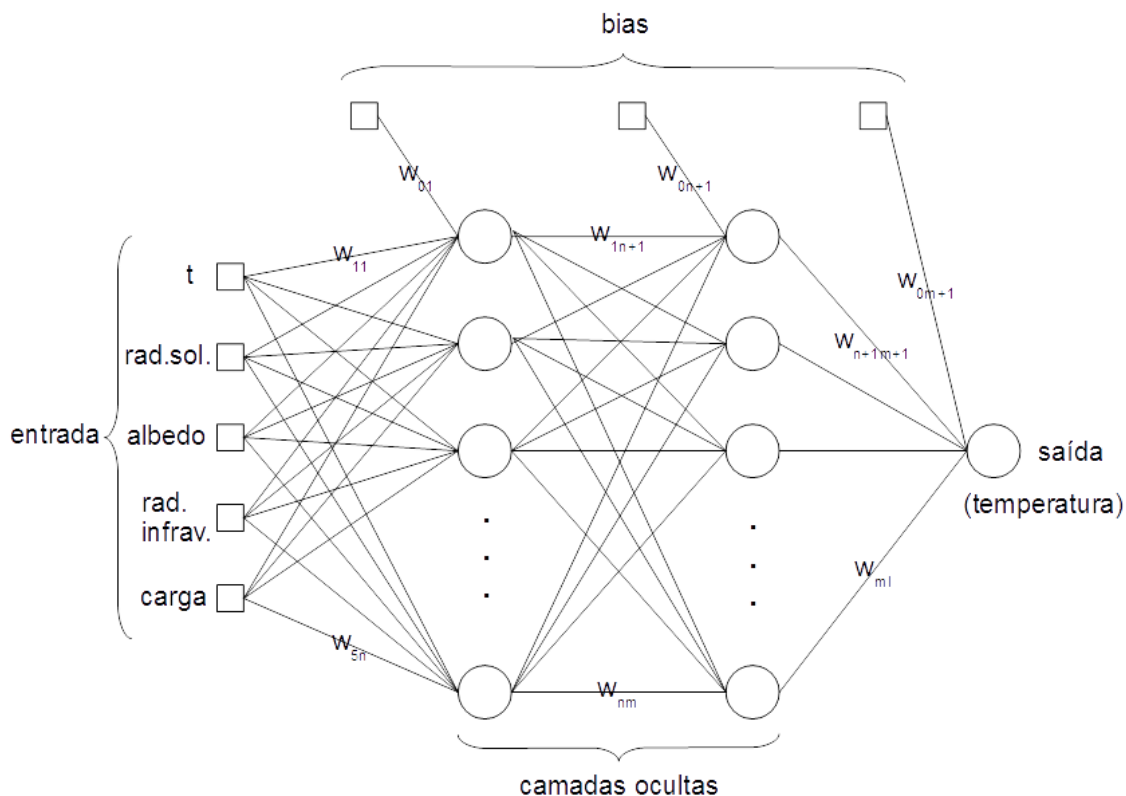
*Valores extraídos da biblioteca interna do Thermal Desktop.

Fonte: C&R Technologies (2013).

Para esse estudo, foram selecionadas como variáveis de entrada: (i) o tempo (t); as variáveis ambientais: (ii) radiação solar, (iii) albedo e (iv) radiação

infravermelha da Terra; e (v) uma carga térmica. Como saída da rede foi escolhida a temperatura na superfície da esfera (que foi considerada como constante ao longo de toda a superfície). Uma ilustração dessa arquitetura pode ser vista na Fig. 4.1.

Figura 4.1. Arquitetura da rede utilizada para a esfera em órbita da Terra.



Fonte: Produção do autor.

Quatro cenários diferentes para treinamento da rede neural foram gerados no *software* SINDA (C&R TECHNOLOGIES, 2013), conforme mostrado na Tabela 4.2. Os cenários Hot (quente) e Cold (frio) contêm os valores máximos e mínimos, respectivamente, para cada uma das variáveis. O cenário A contém os valores mínimos para as variáveis ambientais e o valor máximo para a carga. Já o cenário B contém os valores máximos para as variáveis ambientais e o valor mínimo para a carga.

Tabela 4.2. Cenários simulados para a esfera de alumínio.

Cenário	Radiação Solar (W/m ²)	Albedo*	Radiação Infravermelha (W/m ²)	Carga (W)
Hot	1418	0,42	233	200
Cold	1326	0,34	208	0
A	1326	0,34	208	200
B	1418	0,42	233	0

* O Albedo é descrito como uma fração da radiação solar refletida pela Terra.

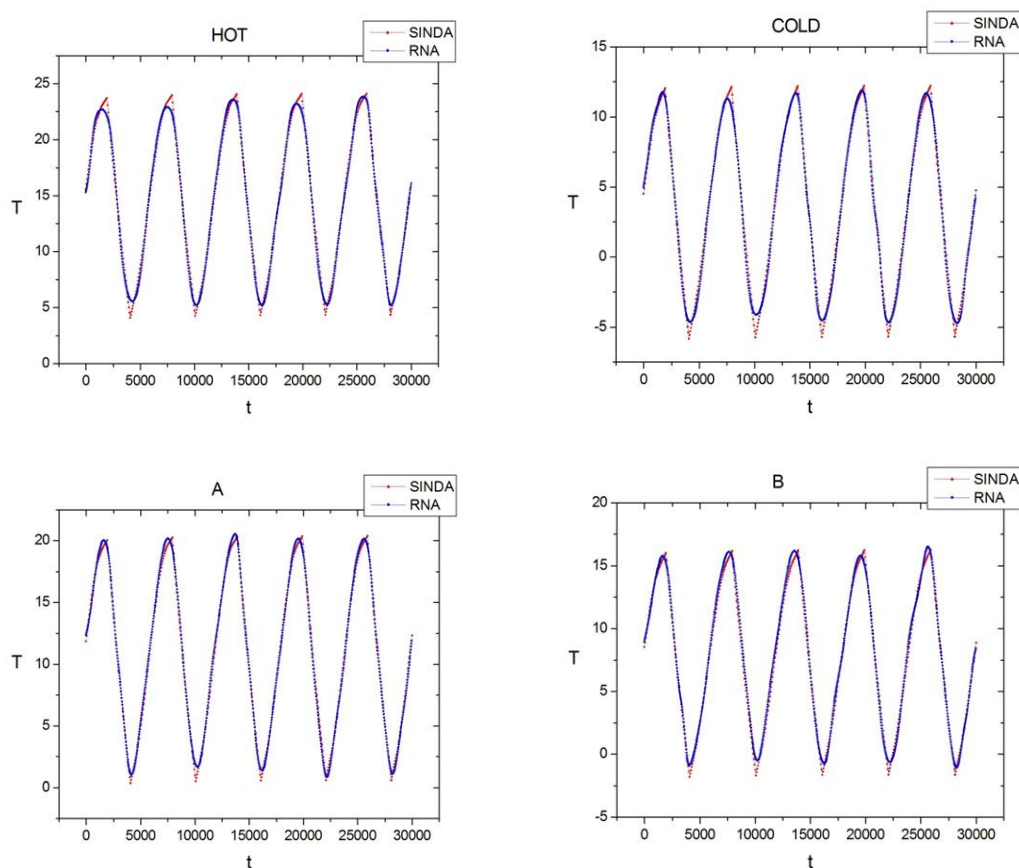
Fonte: Produção do autor.

Os dados dessa tabela, referentes aos cenários Hot, Cold, A e B foram utilizados no software SINDA para gerar os dados de temperatura. Após a simulação atingir o estado de equilíbrio estacionário (*steady state*), foram geradas mais 5 órbitas, com o objetivo de gerar um conjunto de dados contendo algumas órbitas em equilíbrio para serem utilizadas no treinamento da RNA.

A Fig. 4.2 mostra os resultados obtidos pela RNA (curvas azuis) em comparação com os gerados pelo SINDA (curvas vermelhas), para um total de 5 períodos orbitais.

Foram realizados diversos testes variando o número de neurônios da rede, assim como os parâmetros internos. A rede neural selecionada para o treinamento contém duas camadas ocultas com 30 neurônios em cada uma e os parâmetros selecionados para a rede MLP foram: tolerância $\tau = 0,001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Esses parâmetros foram selecionados por apresentarem erros mais baixos e maior estabilidade nos testes realizados.

Figura 4.2. Comparação das curvas de temperatura $T(^{\circ}\text{C})$ em função do tempo $t(\text{s})$ para a esfera de alumínio.



Fonte: Produção do autor.

A Tabela 4.3 contém os erros calculados a partir dos dados mostrados em cada gráfico. O erro máximo observado foi de apenas $1,62^{\circ}\text{C}$, o que indica que a rede neural é capaz de reproduzir os dados apresentados durante o treinamento.

Tabela 4.3. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.

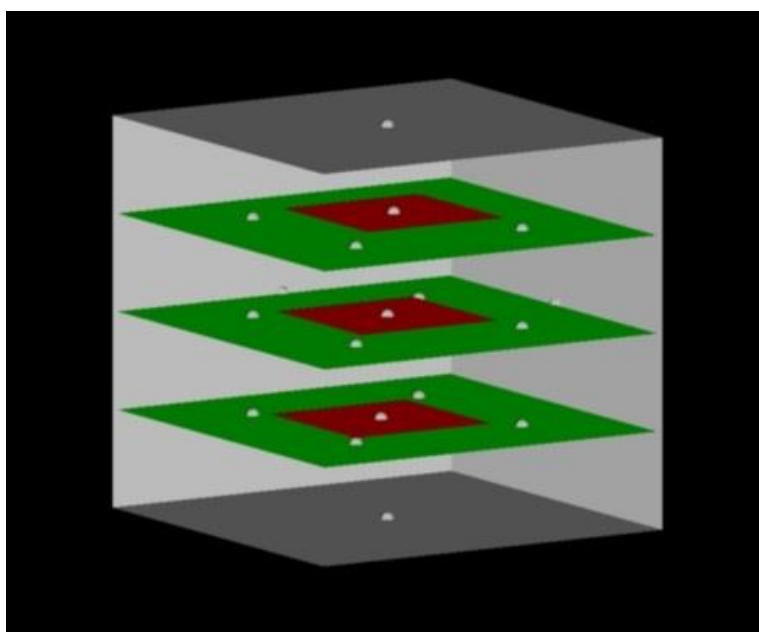
Cenário	Erro médio ($^{\circ}\text{C}$)	Desvio padrão	Erro máximo ($^{\circ}\text{C}$)
Hot	0,38	0,29	1,62
Cold	0,36	0,26	1,62
A	0,31	0,20	1,33
B	0,38	0,25	1,32

Fonte: Produção do autor.

4.1.2 CubeSat

Visando acrescentar a complexidade de um satélite, mas ainda simplificado, foi projetado um modelo de um CubeSat conforme ilustrado na Fig. 4.3. Nessa figura, as paredes frontais foram ocultadas para que o interior possa ser visto.

Figura 4.3. Modelo térmico do nano-satélite.



Fonte: Produção do autor.

O CubeSat 1U é composto por uma caixa de alumínio de 10 x 10 x 10 cm, com três placas de circuito impresso (PCB, do inglês *printed circuit board*; em verde) em seu interior que contêm um componente dissipativo cada (em vermelho). A potência nominal definida para cada um dos equipamentos foi de 0,8, 0,4 e 0,6 W, de cima para baixo, respectivamente. A posição dos componentes superior e inferior foi deslocada em relação eixo central, de modo que o componente superior está mais próximo da aresta ao fundo da figura e o componente inferior está mais próximo da aresta localizada mais à frente. As

pequenas esferas brancas representam os nós térmicos, onde as temperaturas são calculadas.

Foi utilizada uma órbita equatorial (inclinação igual a zero) a 500 km de altitude (LEO). Isso resulta em um período de 5676,98 s ou, aproximadamente, 94,62 min. A atitude é estabilizada em três eixos, com um dos lados sempre voltado para a superfície da Terra. As fontes de radiação externas foram consideradas constantes, com fluxo solar de 1354 W/m^2 , albedo de 35 % e radiação infravermelha referente a um corpo negro de 250 K.

As propriedades termo-físicas utilizadas podem ser vistas na Tabela 4.4 e as propriedades ópticas na Tabela 4.5. Foi considerado que as superfícies externas são cobertas com células solares e as superfícies internas pintadas com tinta preta. Os componentes internos são compostos de silício e cobertos com resina epóxi.

Tabela 4.4. Propriedades termo-físicas.

Material	Densidade (kg/m^3)	Condutividade Térmica ($\text{W/m } ^\circ\text{C}$)	Calor Específico ($\text{J/kg } ^\circ\text{C}$)
Liga de Alumínio	2710	168,0	963,0
Fibra de Vidro (PCB)	2440	1,1	737,0
Silício	2320	148,8	712,0

Fonte: Costa (2010).

Tabela 4.5. Propriedades ópticas.

Material	Absortividade (α)	Emissividade (ϵ)	α/ϵ
PCB	0,75	0,89	0,843
Resina Epóxi	0,93	0,85	1,094
Tinta Preta	0,95	0,87	1,092
Células Solares	0,90	0,80	1,125

Fonte: Costa (2010).

As temperaturas foram calculadas pelo software SINDA. Foram feitas 8 simulações independentes, variando o modo de operação dos 3 equipamentos. Assumindo 0 (zero) para o caso de um equipamento estar desligado e 1 (um) para ligado, o modo de operação dos equipamentos pode ser definido como uma sequência de três números, onde cada número representa o estado de um equipamento (ligado ou desligado). Dessa forma, tem-se que as oito configurações possíveis podem ser representadas como: 000, 001, 010, 011, 100, 101, 110 e 111.

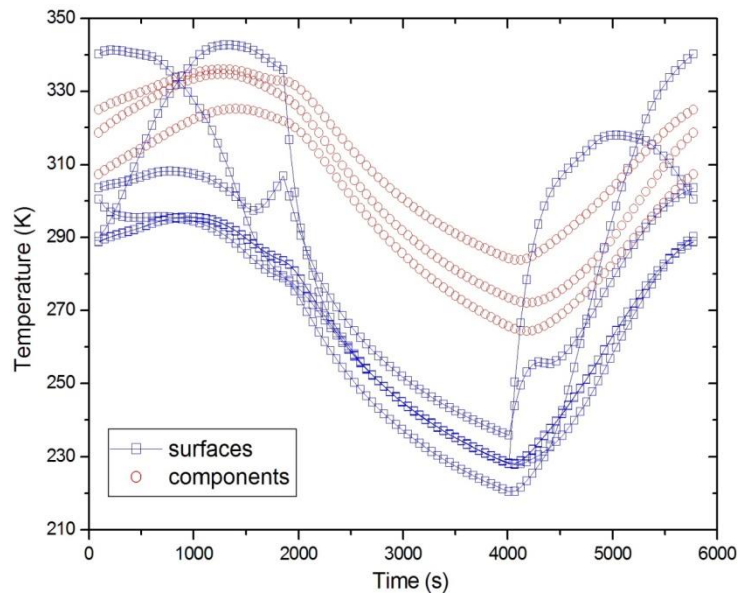
Primeiro, foi calculado no software SINDA o estado de equilíbrio e depois as temperaturas transientes ao longo de 10 órbitas. Em seguida, os dados da última órbita foram extraídos para que pudessem ser utilizados para treinar a RNA.

O modelo térmico utilizado contém 21 nós. Apenas os dados dos nós correspondentes às 6 superfícies externas e aos 3 equipamentos internos (total de 9 nós) foram usados no treinamento. Os nós restantes, referentes a 4 nós em cada PCB, não seriam importantes para um Simulador Operacional, uma vez que a telemetria do satélite normalmente não contém esse tipo de informação.

Os dados gerados pelo SINDA podem ser vistos na Fig. 4.4. As curvas representam a variação da temperatura das 6 superfícies externas (quadrados azuis) e dos 3 componentes internos (círculos vermelhos) ao longo de um período orbital. Neste caso, as linhas são apenas guias para os olhos.

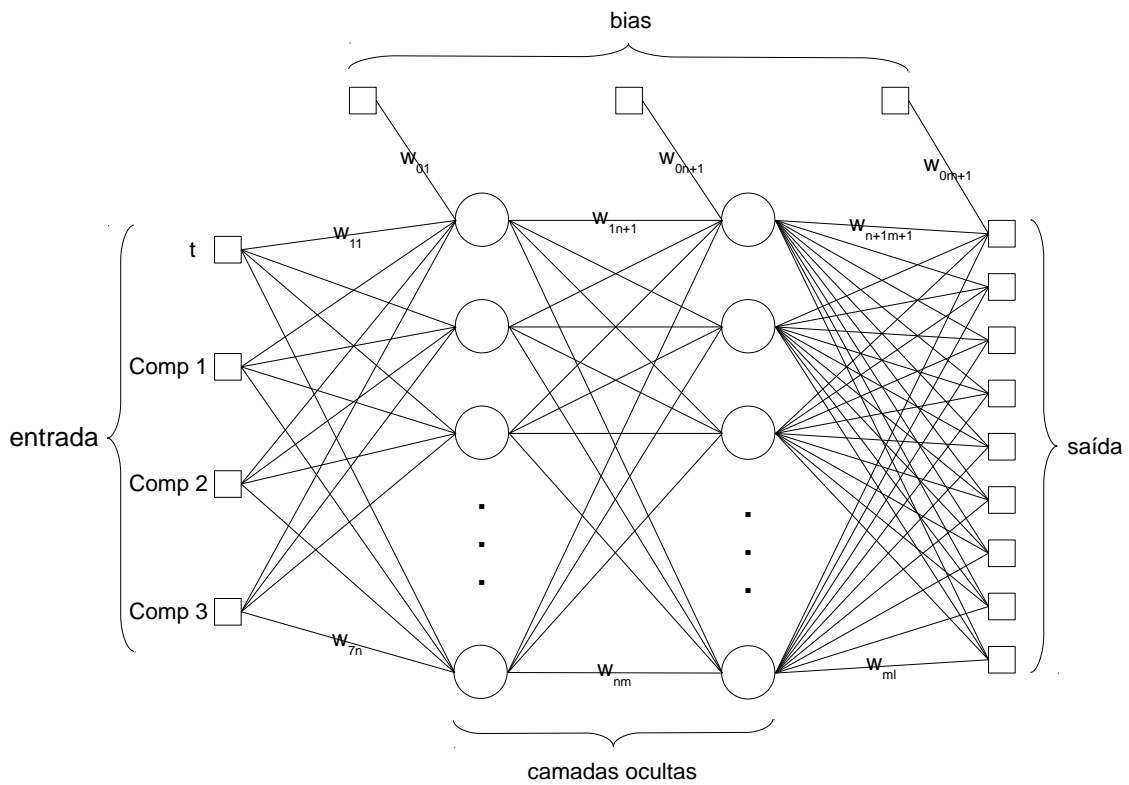
Para realizar o treinamento, foi utilizada uma rede do tipo MLP, ilustrada na Fig. 4.5. As entradas da rede são o tempo de órbita (t) e o estado de funcionamento (0 ou 1) de cada componente (Comp 1, Comp 2 e Comp 3). Nas duas camadas ocultas foram utilizados 40 neurônios (círculos). Durante os testes, o número de neurônios foi variado de 20 a 50 e esse valor de 40 neurônios foi escolhido por apresentar um erro mais baixo. Apesar de não aparecerem na figura, cada conexão possui um peso (w) e cada neurônio possui um *bias* associado. Para este modelo do *CubeSat*, foram utilizados 168 padrões de treinamento.

Figura 4.4. Curvas de temperatura em função do tempo, geradas pelo SINDA.



Fonte: Produção do autor.

Figura 4.5. Arquitetura da RNA para o CubeSat no estudo preliminar 1.



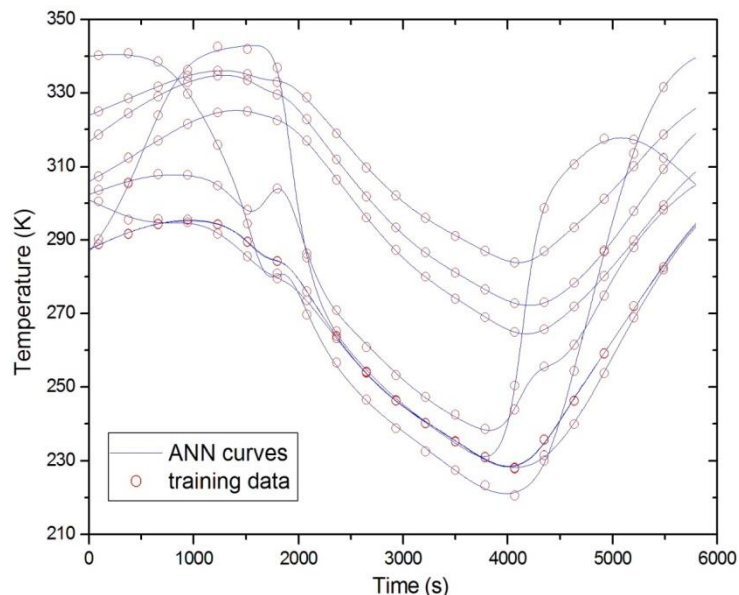
Fonte: Produção do autor.

Após a realização de diversos testes para definir os melhores parâmetros, a RNA foi treinada com o algoritmo GD e os seguintes valores: tolerância $\tau = 0,0001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Em caso da rede não convergir para a tolerância especificada, a execução foi interrompida após 10^6 épocas.

A partir dos dados da Fig. 4.4, 20 pontos de cada curva foram selecionados para treinar a RNA. O processo de treinamento foi realizado com o objetivo de minimizar o erro da rede de acordo com a tolerância especificada, para todas as curvas simultaneamente.

Após a conclusão do treinamento, a RNA foi usada para reconstruir as curvas de temperatura, com base no conhecimento adquirido. Essas curvas são mostradas na Fig. 4.6 (linhas azuis contínuas), juntamente com os pontos utilizados para o treinamento (círculos vermelhos). As curvas produzidas pela RNA mostram um bom acordo com os dados utilizados para treinamento.

Figura 4.6. Curvas de temperatura geradas pela RNA e os pontos usados no treinamento.

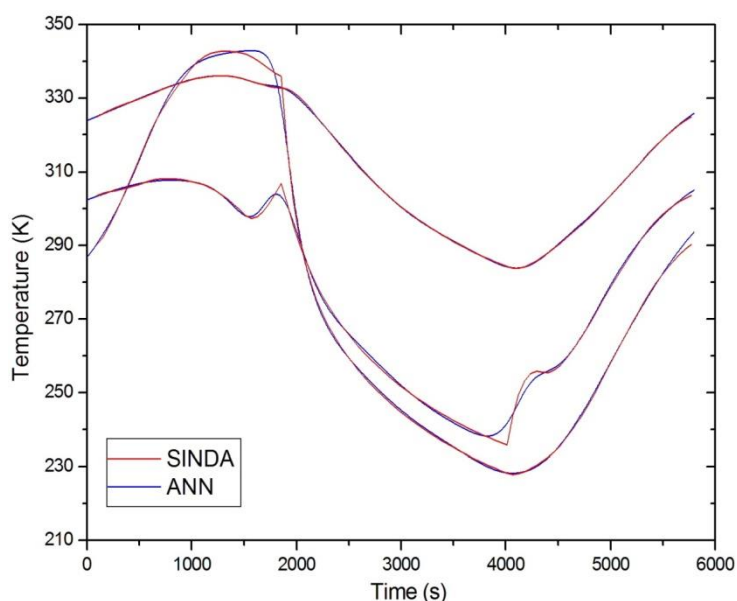


Fonte: Produção do autor.

Como a rede neural foi treinada com apenas 20 pontos por curva para aprender a variação das temperaturas ao longo de toda a órbita, é importante comparar as curvas geradas com todo o conjunto de dados produzido pelo SINDA, incluindo os dados não utilizados no treinamento. A Fig. 4.7 ilustra essa comparação.

Para melhor visualização, foram graficados os dados para apenas 3 nós. As curvas produzidas pela RNA (em azul) apresentam uma grande proximidade com as geradas pelo SINDA (em vermelho). No entanto, em alguns locais a diferença é maior, especialmente nas regiões com variações mais abruptas (picos e vales).

Figura 4.7. Comparação entre as curvas geradas pela RNA e pelo SINDA.



Fonte: Produção do autor.

A Tabela 4.6 contém o erro médio calculado a partir da comparação entre os dados gerados pela RNA com o conjunto de dados completo obtido a partir do SINDA (não apenas os 20 pontos usados no treinamento).

Para todos os casos o erro médio é menor que 1 K. O erro máximo observado foi de 13,3 K, porém, conforme observado na Fig. 4.7, essa diferença ocorre apenas em algumas regiões mais complexas. Uma forma de

evitar esse problema seria utilizar uma quantidade maior de pontos para treinar a rede neural.

Tabela 4.6. Erro médio dos resultados da RNA em comparação com o SINDA.

Nó	Erro Médio (K)
Superfície 1	0,60
Superfície 2	0,61
Superfície 3	0,44
Superfície 4	0,66
Superfície 5	0,42
Superfície 6	0,77
Componente 1	0,14
Componente 2	0,14
Componente 3	0,13

Fonte: Produção do autor.

O resultado demonstrou que a RNA pode aprender e reproduzir o comportamento térmico de um satélite pequeno simples em órbita da Terra com boa precisão. Além disso, a RNA foi capaz de estimar a temperatura para qualquer ponto ao longo da órbita, embora tenha sido treinada com uma quantidade limitada de pontos.

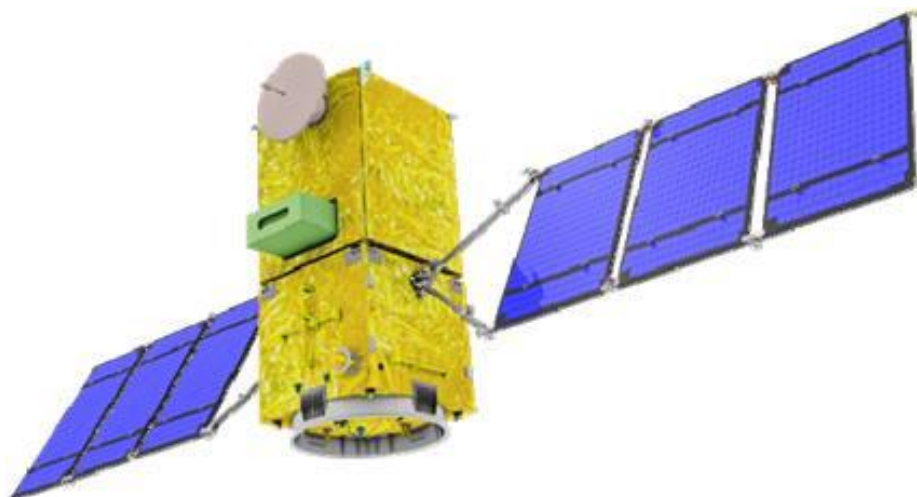
Esses resultados foram apresentados no *23rd ABCM International Congress of Mechanical Engineering* (COBEM 2015), realizado de 6 a 11 de dezembro de 2015, no Rio de Janeiro, RJ (REIS JUNIOR et al., 2015).

4.1.3 Satélite Amazonia-1

Visando avaliar a escalabilidade da solução, a mesma foi aplicada a um satélite real de médio porte com 40 equipamentos, o satélite Amazonia-1 (INPE, 2018). Este é um satélite de sensoriamento remoto que está atualmente em desenvolvimento no INPE. O seu principal objetivo é monitorar os recursos

naturais brasileiros, especialmente a floresta Amazônica e áreas agrícolas. Essa é a primeira missão a utilizar a Plataforma Multimissão (PMM), desenvolvida no INPE com o apoio da Agência Espacial Brasileira (AEB). A Fig. 4.8 ilustra uma concepção artística do satélite Amazonia-1.

Figura 4.8. Concepção artística do satélite Amazonia-1.



Fonte: Produção do autor.

Algumas características do Amazonia-1 são:

- I. Dimensões do corpo principal: 2,20 x 0,95 x 0,95 m;
- II. Massa: cerca de 500 kg;
- III. Órbita: Sol-síncrona;
- IV. Altitude: 752,4 km;
- V. Inclinação: $98,405^\circ$ (polar);
- VI. Estabilização: 3 eixos;
- VII. Tempo de vida da missão: 4 anos.

Para o treinamento da RNA, o software SINDA foi utilizado para simular dois cenários extremos (SILVA et al., 2014). O primeiro, trata-se de um cenário operacional com todos os equipamentos trabalhando normalmente e com os parâmetros do ambiente espacial nos seus valores máximos (caso quente). No segundo cenário, por sua vez, os equipamentos estão em modo de espera (*stand by*) e os parâmetros ambientais foram definidos em seus valores

mínimos (caso frio). Os parâmetros específicos para cada cenário podem ser vistos na Tabela 4.7.

Tabela 4.7. Cenários simulados para o satélite Amazonia-1.

Cenário	Radiação Solar (W/m²)	Albedo (%)	Radiação Infravermelha (W/m²)	Equipamentos
Hot	1418	42	233	Em operação
Cold	1326	34	208	Stand by

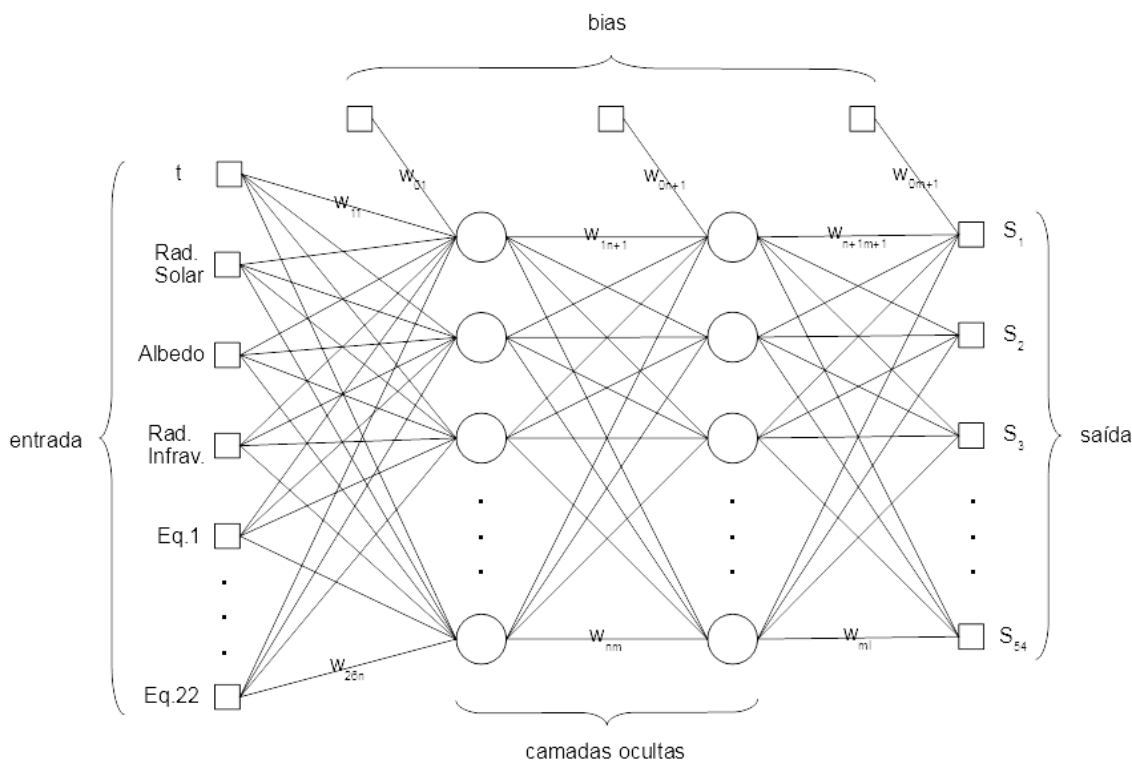
Fonte: Produção do autor.

Após atingir o estado estacionário, foram geradas mais 5 órbitas. Os dados para essas 5 órbitas foram utilizados para treinamento da RNA. Como a potência de alguns equipamentos não varia (ou varia muito pouco) entre os casos frio e quente, foram selecionados 22 equipamentos, dos 40 que compõem o modelo térmico do satélite. Adicionalmente, foram selecionados 54 nós, ou pontos de interesse, para serem reproduzidos pela RNA.

A estrutura da rede neural utilizada pode ser vista na Fig. 4.9. A rede é composta por uma camada de entrada com 26 elementos (tempo, 3 variáveis ambientais e 22 equipamentos), duas camadas ocultas com 50 neurônios cada e 54 neurônios na camada de saída. Foram testados números de neurônios nas camadas ocultas variando de 30 a 60, sendo que o valor de 50 neurônios foi o que apresentou melhores resultados.

Essa rede neural foi treinada com o algoritmo Gradiente Descendente e os parâmetros utilizados foram: tolerância $\tau = 0,00001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Foram testadas diferentes combinações desses parâmetros e os valores escolhidos foram definidos de forma a minimizar os erros de treinamento. Em caso da rede não convergir para a tolerância especificada, o treinamento foi interrompido após 10^6 épocas.

Figura 4.9. Arquitetura da RNA para o CubeSat no estudo preliminar 2.



Fonte: Produção do autor.

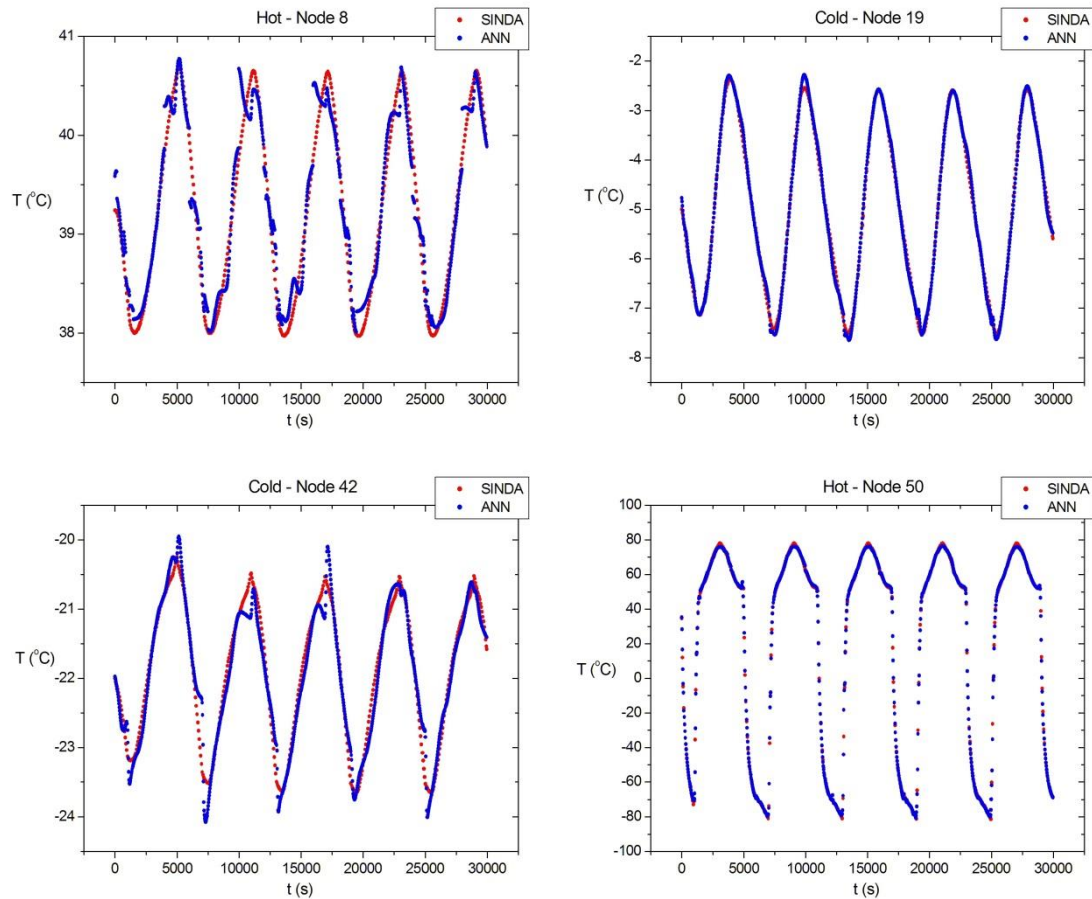
A Fig. 4.10 mostra a comparação entre algumas das curvas de temperatura geradas pela RNA (pontos azuis) e os dados gerados no SINDA (pontos vermelhos) para 5 períodos orbitais.

De forma geral, as curvas produzidas pela RNA apresentam uma extensa sobreposição com as geradas pelo SINDA. No entanto, em alguns locais existem diferenças importantes, com oscilações e variações abruptas de temperatura.

O erro médio para todas as 54 curvas é de $0,2^{\circ}\text{C}$ e o erro máximo observado foi de $18,8^{\circ}\text{C}$, como pode ser visto na Tabela 4.8.

As diferenças observadas para o erro máximo se devem principalmente à grande variação na escala das temperaturas para alguns nós específicos e também ao uso de uma única rede neural para aprender o comportamento de todos os 54 nós de temperatura do satélite. Embora o comportamento geral esteja bem preservado como pode ser observado nos gráficos.

Figura 4.10. Comparação das curvas de temperatura entre SINDA e RNA para quatro nós diferentes.



Fonte: Produção do autor.

Tabela 4.8. Erros da comparação entre as curvas geradas pela RNA e pelo SINDA.

Cenário	Erro médio (°C)	Desvio padrão	Erro máximo (°C)
Hot	0,2	0,3	18,8
Cold	0,2	0,2	13,6

Fonte: Produção do autor.

Esses resultados foram apresentados no 7º Workshop em Engenharia e Tecnologia Espaciais (WETE), realizado de 23 a 24 de agosto de 2016 em São José dos Campos, SP (REIS JUNIOR et al., 2016).

Os resultados apresentados até aqui evidenciam que as redes neurais do tipo MLP podem aprender o comportamento térmico de veículos espaciais em órbita da Terra, inclusive de um satélite real como o Amazonia-1, com elevada precisão, pelo menos para alguns cenários específicos. Além disso, a RNA pode estimar a temperatura para qualquer ponto da órbita, mesmo que tenha sido treinada com um número limitado de pontos.

Contudo, nosso objetivo não é apenas reproduzir as temperaturas de satélites em órbita, mas sim simular em tempo real o comportamento térmico para qualquer cenário ambiental e de operação do satélite.

4.1.4 Inclusão de cenários intermediários

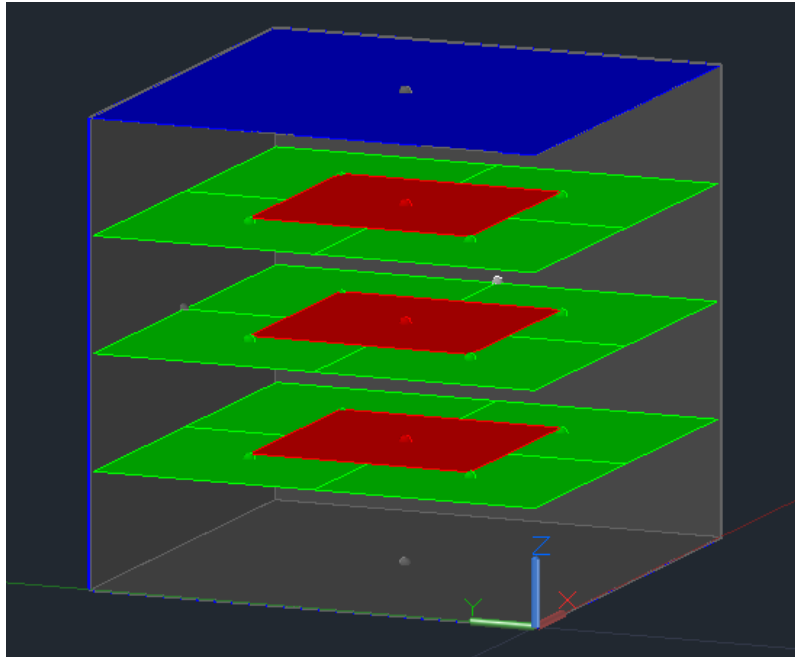
Com o objetivo de obter a generalização da RNA na simulação do comportamento térmico de satélites para cenários não apresentados durante o treinamento, foi realizado um estudo com a inclusão de diversos cenários com valores intermediários para as variáveis de entrada da rede.

A ideia está baseada na premissa de que, ao receber informação sobre qual o comportamento térmico do satélite frente a mudanças nas variáveis de entrada, a RNA será capaz de aprender qual a influência de cada variável no comportamento térmico.

Para tanto, o modelo de CubeSat (Seção 4.1.2) foi utilizado, considerando agora, os componentes centralizados em relação às placas de circuito impresso (PCBs), conforme Fig. 4.11. Além disso, foram adicionados contatos térmicos entre as PCBs e as paredes do CubeSat (ocultados na imagem para melhor visualização), para tornar o modelo um pouco mais realista.

A órbita é polar, com inclinação igual a 90° (condizente com a de um satélite de observação da Terra), a uma altitude de 500 km. O período de uma órbita tem duração de 6000 s ou 100 min e a atitude é estabilizada em 3 eixos, com um lado do satélite sempre apontado para a superfície da Terra. As propriedades termo-físicas e ópticas são as mesmas das Tabelas 4.4 e 4.5.

Figura 4.11. Novo modelo térmico do CubeSat.



Fonte: Produção do autor.

Os cenários simulados podem ser vistos na Tabela 4.9. O primeiro é um cenário operacional com todos os componentes trabalhando com potência máxima e todos os parâmetros do ambiente espacial em seus valores máximos (Hot). No segundo cenário, os componentes estão todos em *stand by* e os parâmetros ambientais estão em seus valores mínimos (Cold). Os outros cenários consistem em variações individuais de cada variável, alternando entre seus valores máximo, médio e mínimo enquanto os demais parâmetros são mantidos nos seus valores máximos ou mínimos. Um cenário com todas as variáveis em seus valores médios (Medium), e, por fim, dois cenários adicionais arbitrários, A e B, para testar a capacidade de generalização da RNA, foram simulados.

No SINDA, inicialmente o estado estacionário e, então, as temperaturas transientes para 10 órbitas foram calculados. Em seguida, a RNA foi treinada com dois conjuntos de treinamento. O primeiro foi formado a partir dos dados

da última órbita dos cenários 1 a 14, e o segundo com os dados das últimas 5 órbitas dos cenários 1 a 27.

Tabela 4.9. Cenários simulados para o CubeSat.

#	Cenário	Comp1 (W)	Comp2 (W)	Comp3 (W)	Radiação Solar (W/m ²)	Albedo* (W/m ²)	Rad. Infrav. (W/m ²)
1	Hot	0,80	0,40	0,60	1418	595,56	233
2	Cold	0,20	0,10	0,15	1326	450,84	208
3	MaxCp1	0,80	0,10	0,15	1326	450,84	208
4	MaxCp2	0,20	0,40	0,15	1326	450,84	208
5	MaxCp3	0,20	0,10	0,60	1326	450,84	208
6	MaxSol	0,20	0,10	0,15	1418	450,84	208
7	MaxAlb	0,20	0,10	0,15	1326	595,56	208
8	MaxIV	0,20	0,10	0,15	1326	450,84	233
9	MinCp1	0,20	0,40	0,60	1418	595,56	233
10	MinCp2	0,80	0,10	0,60	1418	595,56	233
11	MinCp3	0,80	0,40	0,15	1418	595,56	233
12	MinSol	0,80	0,40	0,60	1326	595,56	233
13	MinAlb	0,80	0,40	0,60	1418	450,84	233
14	MinIV	0,80	0,40	0,60	1418	595,56	208
15	Medium	0,50	0,25	0,375	1372	523,20	220,5
16	MedCp1a	0,50	0,40	0,60	1418	595,56	233
17	MedCp1b	0,50	0,10	0,15	1326	450,84	208
18	MedCp2a	0,80	0,25	0,60	1418	595,56	233
19	MedCp2b	0,20	0,25	0,15	1326	450,84	208
20	MedCp3a	0,80	0,40	0,375	1418	595,56	233
21	MedCp3b	0,20	0,10	0,375	1326	450,84	208
22	MedSola	0,80	0,40	0,60	1372	595,56	233
23	MedSolb	0,20	0,10	0,15	1372	450,84	208
24	MedAlba	0,80	0,40	0,60	1418	523,20	233
25	MedAlbb	0,20	0,10	0,15	1326	523,20	208
26	MedIVa	0,80	0,40	0,60	1418	595,56	220,5
27	MedIVb	0,20	0,10	0,15	1326	450,84	220,5
28	A	0,63	0,12	0,48	1345	551,45	229
29	B	0,26	0,31	0,19	1398	503,28	214

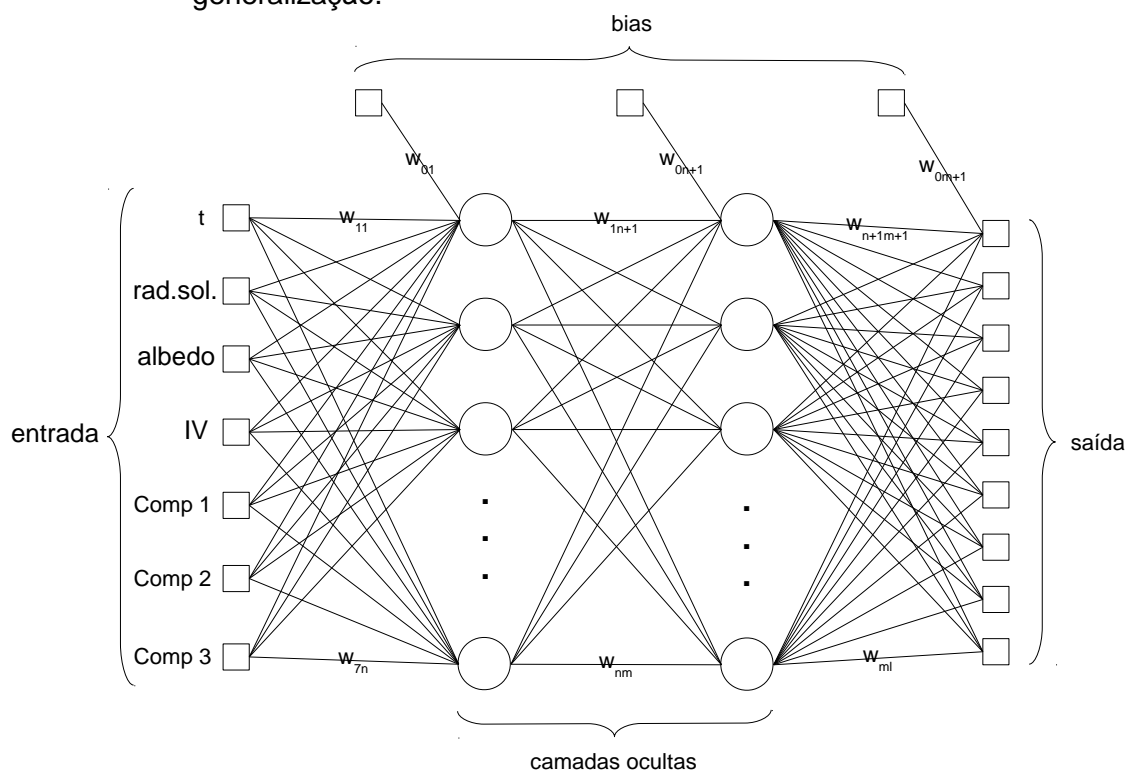
* O Albedo aqui é descrito em W/m², para permitir o cálculo dos valores máximo, médio e mínimo.

Fonte: Produção do autor.

A Fig. 4.12 ilustra a rede neural MLP utilizada. A estrutura é formada por 7 elementos na camada de entrada, duas camadas ocultas, com 30 a 50 neurônios cada e 9 neurônios na camada de saída. As variáveis de entrada referem-se ao tempo (t), a potência dos 3 componentes, a radiação solar, o albedo e a radiação infravermelha da Terra.

Novamente, a RNA foi treinada com o algoritmo GD e os melhores parâmetros definidos durante os testes foram: tolerância $\tau = 0,0001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Em caso da rede não convergir para a tolerância especificada, o treinamento foi interrompido após 10^6 épocas.

Figura 4.12. Arquitetura da rede utilizada para o CubeSat nos estudos de generalização.



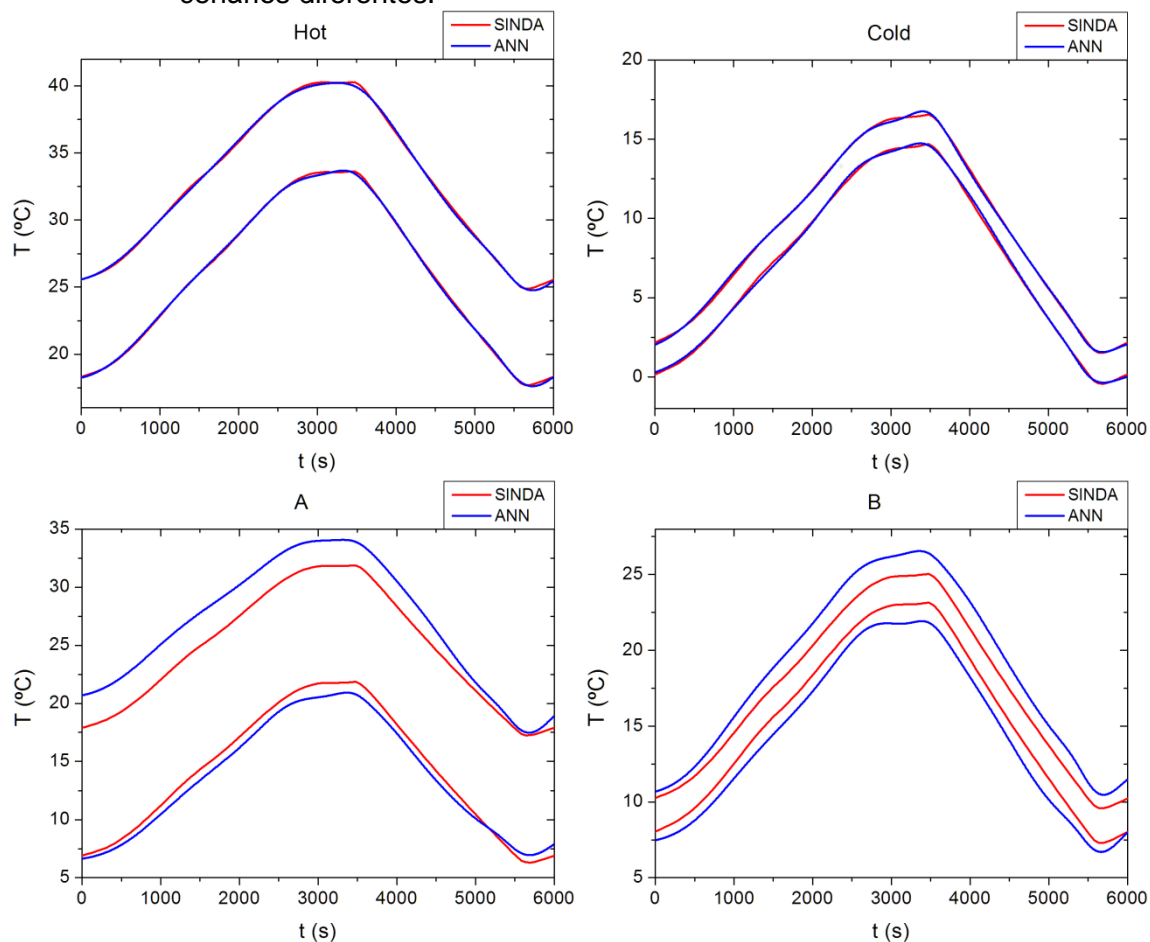
Fonte: Produção do autor.

Os resultados para o primeiro conjunto de treinamento são mostrados na Fig. 4.13. Nessa figura pode ser vista a comparação entre as curvas de temperatura geradas a partir dos dados obtidos no SINDA (em vermelho) e os

gerados pela RNA (em azul), para quatro cenários diferentes (Hot, Cold, A e B). Para facilitar a visualização, as curvas são mostradas como linhas contínuas e foram selecionados apenas dois componentes.

Para os casos Hot e Cold as curvas geradas pela RNA apresentam um bom acordo com as produzidas a partir dos dados gerados pelo SINDA. Já para os cenários A e B (não usados no treinamento) existe uma pequena diferença entre os dados das duas fontes. Como os cenários A e B não foram apresentados à rede durante o treinamento, as saídas geradas para estes casos são exemplos da capacidade de generalização da RNA.

Figura 4.13. Comparação das curvas de temperatura entre SINDA e RNA para quatro cenários diferentes.



Fonte: Produção do autor.

A comparação quantitativa é mostrada na Tabela 4.10. Os erros tanto médios quanto máximos para os casos Hot e Cold são menores do que 1,0°C. Já para os casos A e B os erros médios são 1,1°C e 0,5°C e os erros máximos são 3,0°C e 1,8°C, respectivamente. Conforme descrito na Subseção 3.1, erros menores que 5°C atendem o requisito definido para o Simulador Operacional.

Tabela 4.10. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.

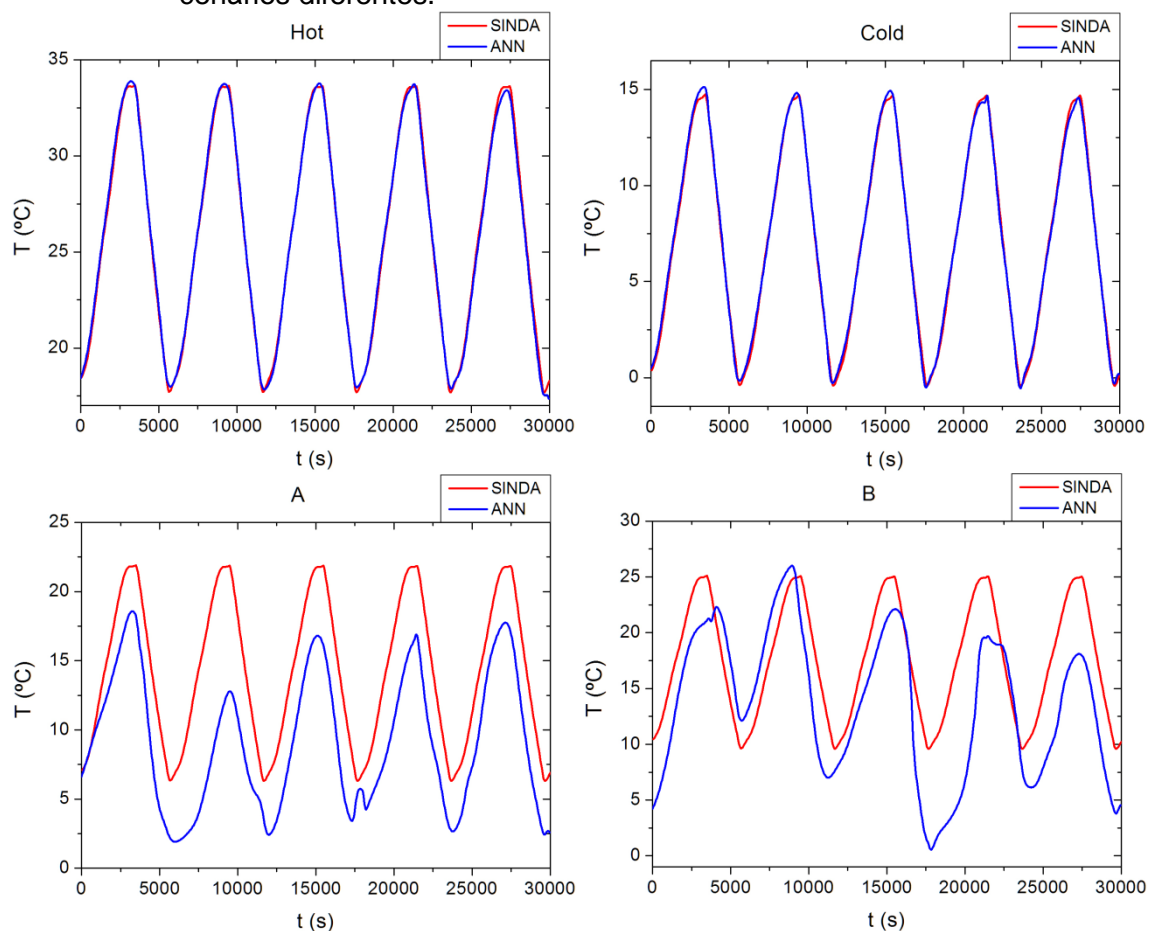
Cenário	Erro médio (°C)	Desvio padrão	Erro máximo (°C)
Hot	0,1	0,08	0,7
Cold	0,1	0,07	0,6
A	1,1	0,29	3,0
B	0,5	0,21	1,8

Fonte: Produção do autor.

As curvas para o segundo conjunto de treinamento podem ser vistas na Fig. 4.14. Desta vez, as figuras foram geradas para 5 órbitas e apenas um componente foi selecionado, para melhor visualização. Novamente, para os cenários Hot e Cold as curvas geradas pela RNA apresentam um bom acordo com as produzidas a partir do SINDA. No entanto, nos casos A e B o resultado da RNA apresenta maior diferença, inclusive divergindo do comportamento oscilatório esperado.

A Tabela 4.11 contém os erros calculados para o segundo conjunto de treinamento. Assim como no primeiro conjunto, os erros para os cenários Hot e Cold são menores do que 1°C. Para os cenários A e B, os erros médios são 3,6°C e 5,6°C, respectivamente. O que, considerados isoladamente, seriam erros até aceitáveis. Contudo, o erro máximo observado é maior do que 20°C para alguns pontos específicos. O que é bem maior do que o limite aceitável citado anteriormente.

Figura 4.14. Comparação das curvas de temperatura entre SINDA e RNA para quatro cenários diferentes.



Fonte: Produção do autor.

Tabela 4.11. Erros calculados da comparação entre as curvas geradas pela RNA e pelo SINDA.

Cenário	Erro médio (°C)	Desvio padrão	Erro máximo (°C)
Hot	0,2	0,12	1,2
Cold	0,1	0,13	1,1
A	3,6	3,14	20,9
B	5,6	4,47	21,8

Fonte: Produção do autor.

Esses resultados foram apresentados na *4th Conference of Computational Interdisciplinary Science (CCIS 2016)*, realizada de 7 a 10 de novembro de 2016, em São José dos Campos, SP (REIS JUNIOR et al., 2016b).

Nos testes realizados, foi observado que ao tentar generalizar para cenários onde apenas uma ou duas variáveis de entrada eram alteradas e as demais eram mantidas em valores utilizados no treinamento, as curvas geradas pela RNA apresentavam o comportamento esperado, com as oscilações referentes às 5 órbitas. No entanto, ao alterar três ou mais variáveis simultaneamente, as curvas divergiram do comportamento esperado, conforme observado para os cenários A e B.

Analisando as curvas de temperatura para várias órbitas, observa-se que o comportamento térmico do satélite após atingir um estado estacionário é cíclico, isto é, para um determinado cenário de ambiente espacial e estado de funcionamento do satélite, a curva de temperatura se repete ao longo de cada nova órbita. Por essa razão, a rede neural pode ser treinada com apenas uma órbita.

Na próxima Seção, é apresentada a metodologia proposta que inclui algumas alterações na arquitetura da rede neural, com o objetivo de obter melhores resultados para a generalização.

4.2 Generalização para Cenários não Treinados

Nesta Seção, são apresentados os resultados obtidos com o aprimoramento da generalização da rede neural para quaisquer cenários de ambiente espacial e estados de funcionamento do satélite.

Com o objetivo de fazer as redes neurais aprenderem o comportamento dinâmico temporal de um sistema físico de forma mais efetiva e realizarem previsões para o seu comportamento futuro, a literatura recomenda a inclusão de informações sobre instantes anteriores ao longo de cada passo de

treinamento da rede, chamada de memória de curto prazo ou de recorrência (HAYKIN, 2001).

A memória de curto prazo na RNA é gerada inserindo como entrada na rede os valores das temperaturas no instante anterior. Dessa forma, as novas temperaturas na saída da rede serão geradas a partir das temperaturas anteriores.

Além disso, os três parâmetros de ambiente espacial: radiação solar, albedo e radiação infravermelha, foram substituídos pelas taxas de aquecimento (*heating rates*) em cada superfície externa do satélite. Em termos físicos, as taxas de aquecimento descrevem de forma mais precisa a quantidade de calor que cada superfície recebe de fontes externas a cada instante.

Nos estudos anteriores, o tempo foi inserido explicitamente nas redes neurais, isto é, havia uma variável de entrada dedicada exclusivamente a inserir os valores de tempo a cada passo do treinamento. Daqui em diante, o tempo não será mais utilizado, pois a informação sobre o tempo agora está inserida na rede de forma implícita por meio das novas variáveis de entrada.

Na Subseção 4.2.1, é apresentada a arquitetura da rede neural e a metodologia de treinamento utilizada para o CubeSat.

Nas Subseções seguintes, são apresentados os resultados para o treinamento dessa rede com os algoritmos Gradiente Descendente (Subseção 4.2.2), Levenberg-Marquardt (Subseção 4.2.3) e Otimização Extrema Generalizada (Subseção 4.2.4), com o objetivo de verificar a influência do algoritmo de treinamento nos resultados. A análise e comparação dos resultados para os três algoritmos é apresentada na Subseção 4.2.5.

A Subseção 4.2.6 traz os resultados de generalização para o satélite Amazonia-1.

4.2.1 CubeSat

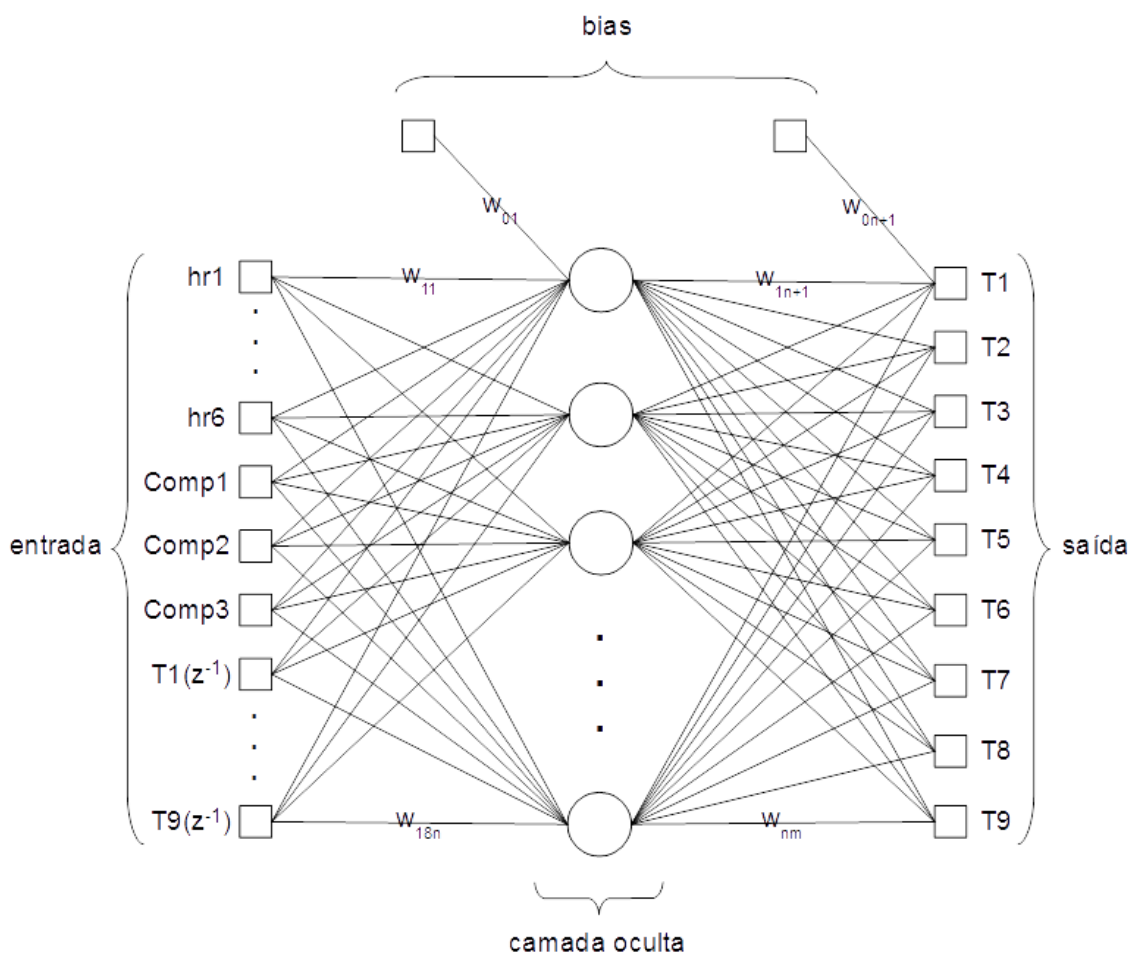
Nesta subseção é apresentada a arquitetura da RNA e a metodologia de treinamento utilizada para o CubeSat. O modelo de CubeSat utilizado como base para esses estudos é o mesmo apresentado na Subseção 4.1.4.

A Fig. 4.15 ilustra a arquitetura da RNA utilizada para o CubeSat. As variáveis $hr1$ a $hr6$ representam as taxas de aquecimento nas 6 superfícies externas do satélite e as variáveis $T1(z^{-1})$ a $T9(z^{-1})$ são as temperaturas esperadas para o instante imediatamente anterior ($\Delta t = 1$ min), extraídas do conjunto de treinamento. Dessa forma, tem-se um total de 18 parâmetros de entrada. No Apêndice A consta um exemplo de conjunto de dados utilizado para treinamento da rede neural.

Um dos fatores que influenciam a capacidade de generalização das redes neurais é a própria complexidade da rede, isto é, o número de camadas, o número de neurônios, etc. Em outras palavras, existe uma relação entre a complexidade da RNA e dos dados contidos no conjunto de treinamento. Por exemplo, se o número de neurônios for muito pequeno, a RNA não será capaz de aprender o conhecimento necessário e, por outro lado, se o número de neurônios for muito grande ocorrerá o que chamamos de *overfitting* ou ajuste excessivo. O *overfitting* ocorre quando a rede se torna excessivamente adaptada ao conjunto de treinamento, perdendo sua capacidade de generalização (HAYKIN, 2001; HAGAN, 2014).

Dessa forma, para diminuir a complexidade da rede, o número de camadas ocultas foi reduzido para apenas uma e o número de neurônios nessa camada foi reduzido para valores que resultaram em menores erros de generalização. No caso dos algoritmos Gradiente Descendente e Otimização Extrema Generalizada os melhores resultados foram obtidos para 35 neurônios, enquanto para o algoritmo Levenberg-Marquardt o melhor resultado foi obtido para 40 neurônios. Esses foram os menores valores obtidos em nossos testes sem prejudicar a capacidade de aprendizagem da rede neural.

Figura 4.15. Arquitetura da RNA proposta para generalização das temperaturas do CubeSat.



Fonte: Produção do autor.

Diferentemente da estratégia adotada na Subseção 4.1.4, onde foram adicionados vários cenários com valores intermediários para cada variável de entrada, aqui foram adicionados cenários completamente aleatórios de forma gradativa e em treinamentos independentes. Além disso, para cada conjunto de cenários, foram realizadas 10 rodadas independentes com diferentes inicializações aleatórias dos pesos. Essas 10 rodadas tem o objetivo de reduzir a tendência dos algoritmos de ficarem presos em mínimos locais, permitindo a comparação dos resultados obtidos com cada algoritmo a partir das médias calculadas.

Em outras palavras, primeiro foi realizado um treinamento apenas com os cenários Hot e Cold, com 10 rodadas diferentes de inicialização dos pesos. Em seguida, foi incluído o cenário A, além dos cenários Hot e Cold. Na sequência, foi sendo adicionado um novo cenário aleatório de cada vez, sempre realizando 10 rodadas independentes para cada conjunto. O último conjunto de cenários utilizado no treinamento incluiu os cenários Hot, Cold, A, B, C e D.

O objetivo é que, com a inclusão dos novos cenários aleatórios, a rede neural seja capaz de aprender a relação entre as mudanças nas variáveis de entrada e as variações decorrentes no comportamento térmico do satélite.

Todo esse processo foi repetido para os três algoritmos: Gradiente Descendente (GD), Levenberg-Marquardt (LM) e Otimização Extrema Generalizada (GEO).

No caso específico do algoritmo LM, para o cálculo de inversão da matriz Hessiana aproximada (ver Eq. 3.21), foi utilizada a biblioteca de funções de álgebra linear LAPACK (*Linear Algebra Package*), mais especificamente a sua interface para linguagem C, denominada LAPACKC (LAPACKC, 2018).

Tabela 4.12. Cenários simulados para o CubeSat.

Cenário	Comp1 (W)	Comp2 (W)	Comp3 (W)	Radiação Solar (W/m²)	Albedo* (W/m²)	Rad. Infrav. (W/m²)
Hot	0,80	0,40	0,60	1418,0	595,56	233,0
Cold	0,20	0,10	0,15	1326,0	450,84	208,0
A	0,63	0,12	0,48	1345,0	551,45	229,0
B	0,26	0,31	0,19	1398,0	503,28	214,0
C	0,50	0,21	0,27	1327,0	491,07	215,0
D	0,45	0,32	0,15	1363,0	546,19	231,0
E	0,69	0,37	0,56	1376,9	519,48	221,6
F	0,49	0,26	0,52	1329,0	468,13	222,6
G	0,53	0,16	0,41	1370,2	538,03	213,4
H	0,61	0,14	0,24	1392,7	502,77	223,5
I	0,45	0,34	0,58	1341,6	523,69	220,3

* O Albedo aqui é descrito em termos de W/m².

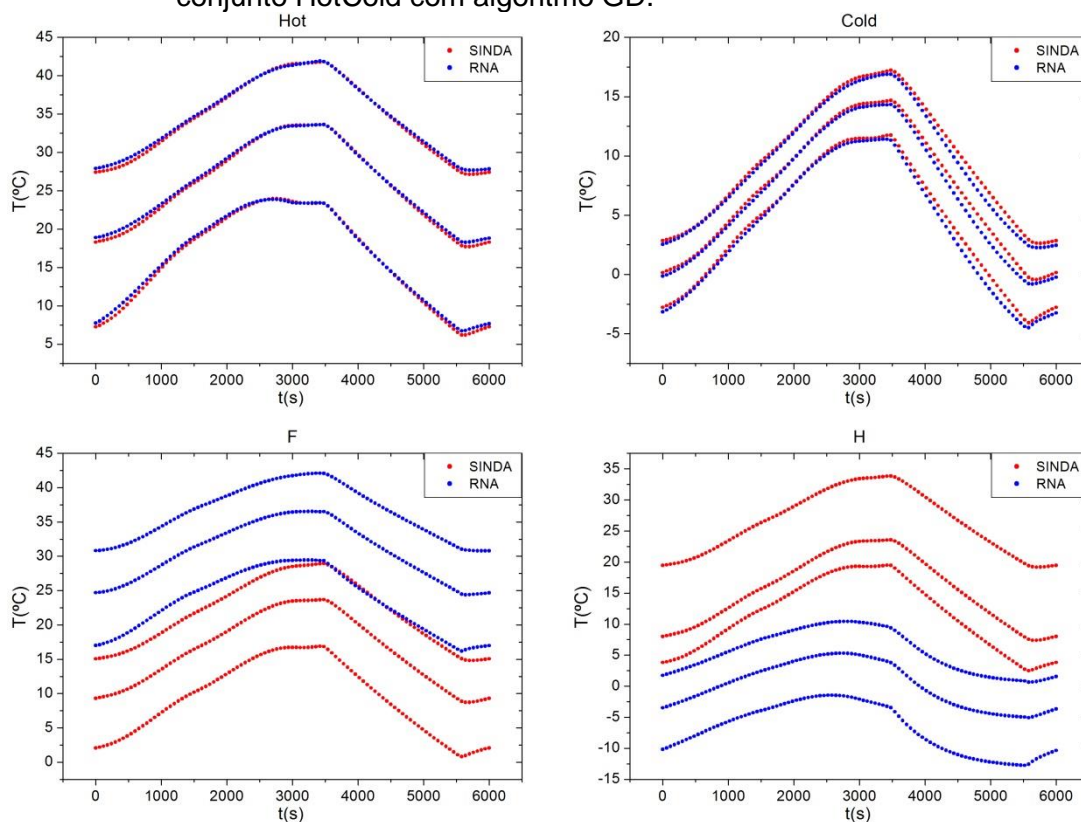
Fonte: Produção do autor.

Para verificar a capacidade de generalização da RNA, foram geradas curvas de temperatura para 5 outros cenários aleatórios (E, F, G, H e I), não apresentados à rede durante o treinamento. Os valores dos parâmetros para cada cenário estão descritos na Tabela 4.12.

4.2.2 Resultados para o algoritmo Gradiente Descendente

A seguir são apresentados os resultados obtidos a partir do treinamento da RNA com o algoritmo Gradiente Descendente. Os parâmetros utilizados para treinamento foram: tolerância $\tau = 0,00001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Em caso de a rede não convergir para a tolerância especificada, o treinamento foi interrompido após 10^6 épocas.

Figura 4.16. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo GD.

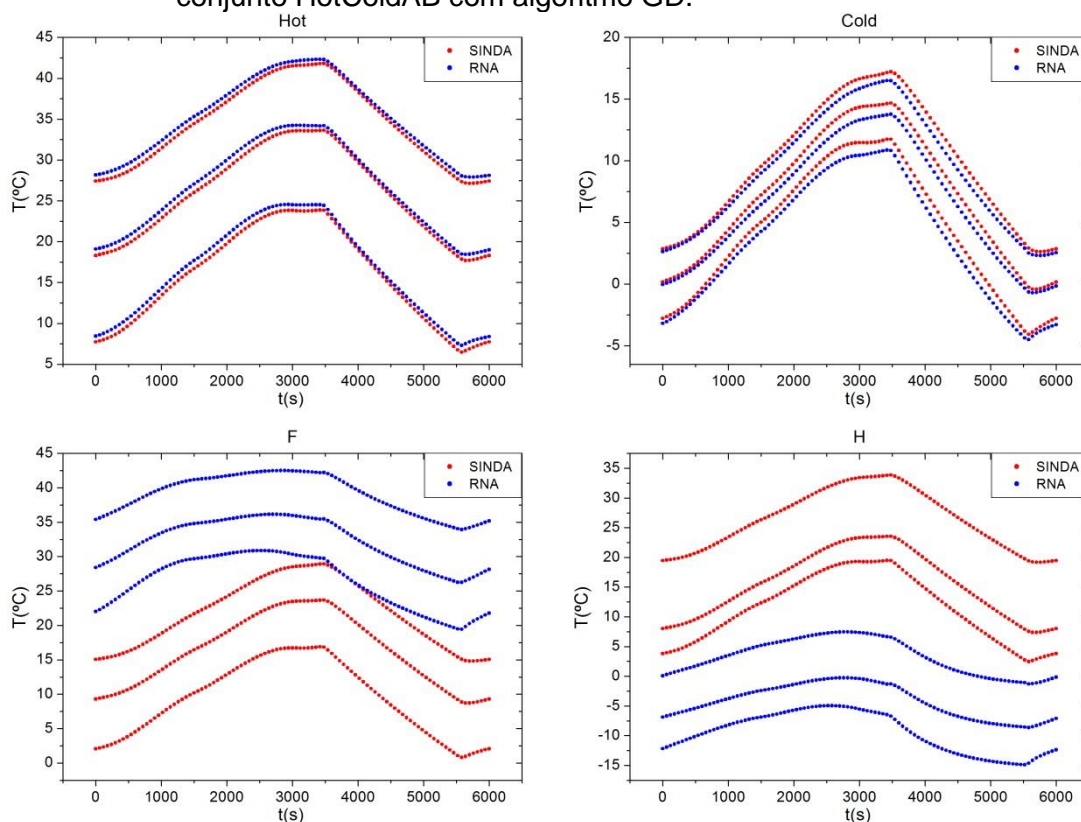


Fonte: Produção do autor.

Na Fig. 4.16, podem ser vistas as curvas de temperatura obtidas para o conjunto HotCold. Para ilustrar a diferença entre os casos treinados e os casos não treinados (generalizados), é mostrada a comparação entre as curvas geradas pela RNA e pelo SINDA para os cenários Hot, Cold, F e H (a diferença para os demais cenários não treinados é semelhante a esses dois casos). Neste caso, fica evidente que a rede neural foi capaz de reproduzir os casos treinados com grande precisão e que o erro de generalização para os casos não treinados é grande.

A seguir, na Fig. 4.17, são mostradas as curvas de temperatura para o conjunto HotColdAB.

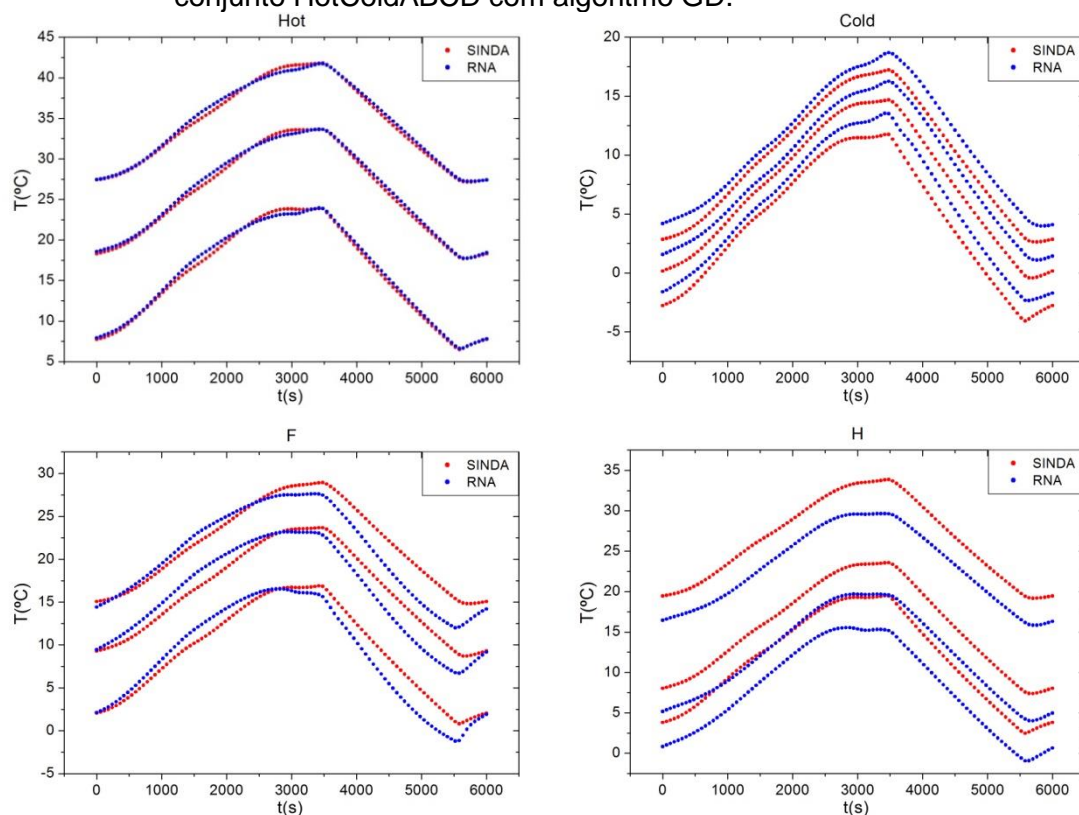
Figura 4.17. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo GD.



Fonte: Produção do autor.

Os resultados para o conjunto HotColdABCD podem ser vistos na Fig. 4.18.

Figura 4.18. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo GD.



Fonte: Produção do autor.

Os resultados para os conjuntos HotColdA e HotColdABC não foram mostrados para evitar repetições excessivas, uma vez que os mesmos conjuntos serão analisados para os algoritmos LM e GEO.

Tabela 4.13. Erros médios de generalização para o algoritmo GD.

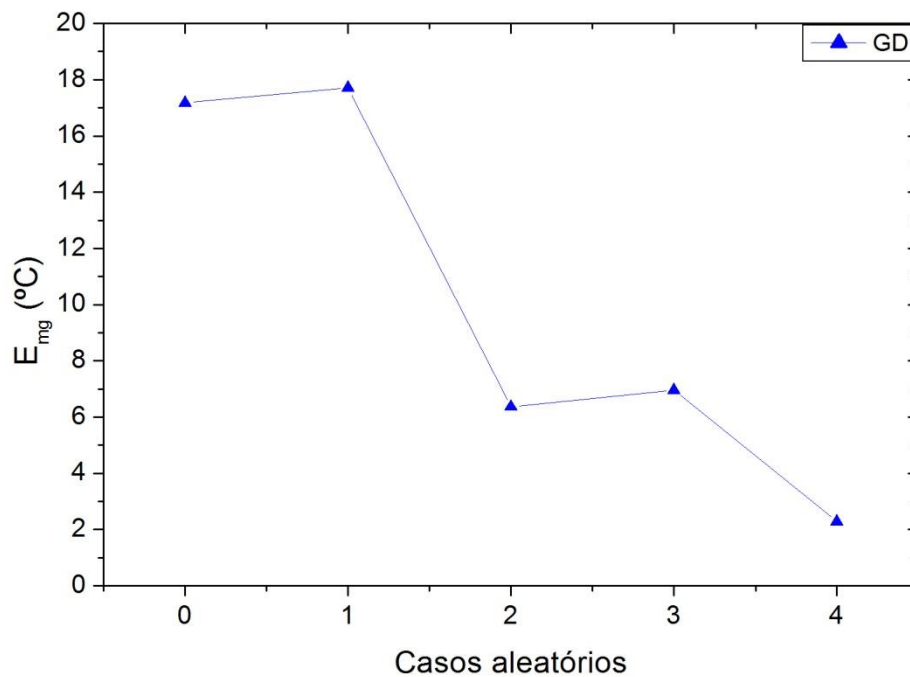
Conjunto	Erro médio (°C)					Média Total
	E	F	G	H	I	
HotCold	17,8	18,4	18,8	16,0	14,9	17,2
HotColdA	20,6	19,7	14,5	13,4	20,4	17,7
HotColdAB	4,1	7,2	1,4	10,0	9,1	6,4
HotColdABC	5,6	9,5	2,5	6,3	10,8	7,0
HotColdABCD	1,2	2,4	1,1	3,9	2,8	2,3

Fonte: Produção do autor.

A Tabela 4.13 contém os valores dos erros médios de generalização para todos os conjuntos de dados e treinamentos realizados com o algoritmo GD.

A evolução do erro médio de generalização para cada conjunto pode ser vista na Fig. 4.19. As linhas são apenas guias para os olhos.

Figura 4.19. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo GD.



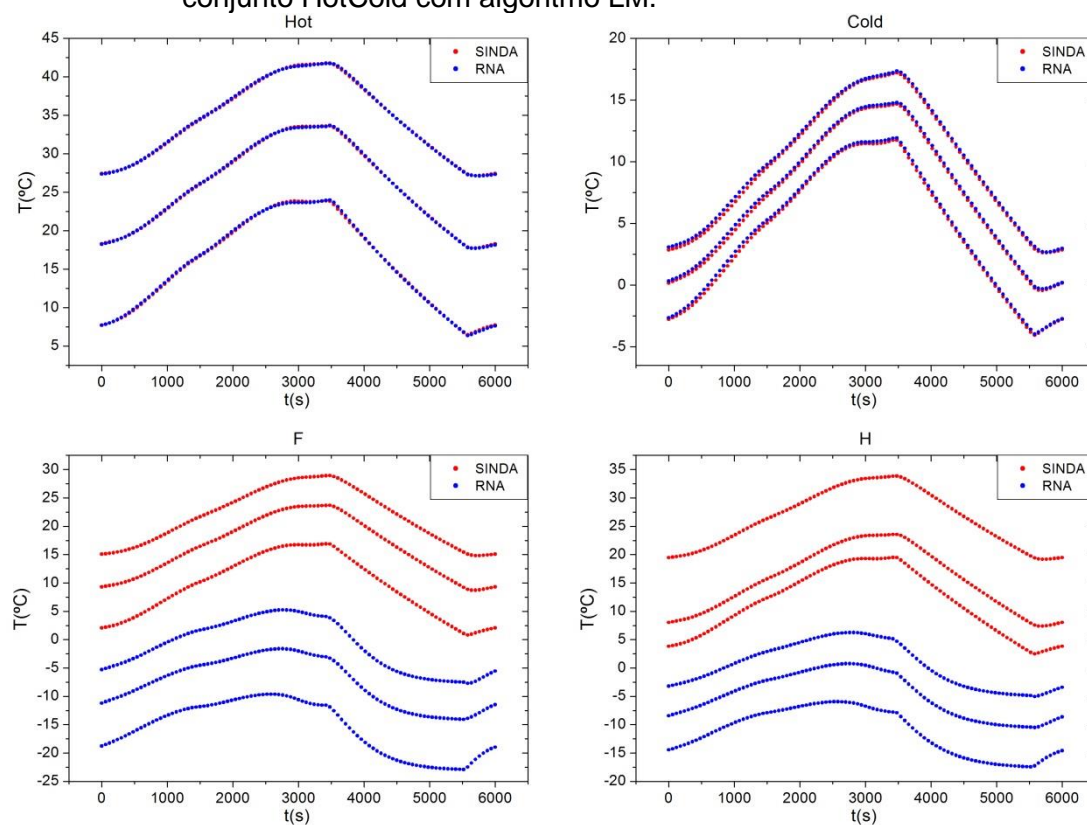
Fonte: Produção do autor.

4.2.3 Resultados para o algoritmo Levenberg-Marquardt

A seguir são apresentados os resultados obtidos a partir do treinamento da RNA com o algoritmo Levenberg-Marquardt. Os melhores parâmetros selecionados para o treinamento com esse algoritmo foram: tolerância $\tau = 0,00001$ e $\mu_i = 0,01$, com $\Delta\mu$ variando de 10,0 até 1,1, sendo multiplicado por 0,99 a cada época. Em caso da rede não convergir para a tolerância especificada, o treinamento foi interrompido após 10^3 épocas.

Na Fig. 4.20, são mostradas as curvas de temperatura para o conjunto HotCold.

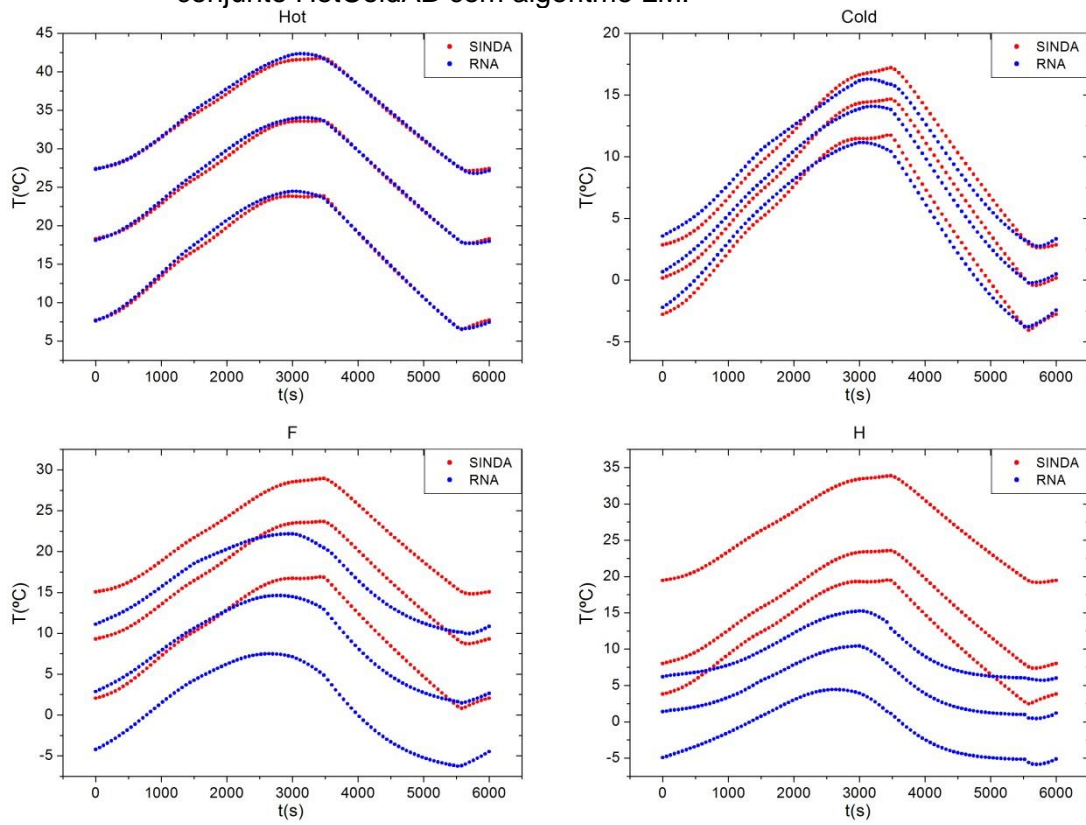
Figura 4.20. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo LM.



Fonte: Produção do autor.

As curvas para o conjunto HotColdAB são mostradas na Fig. 4.21.

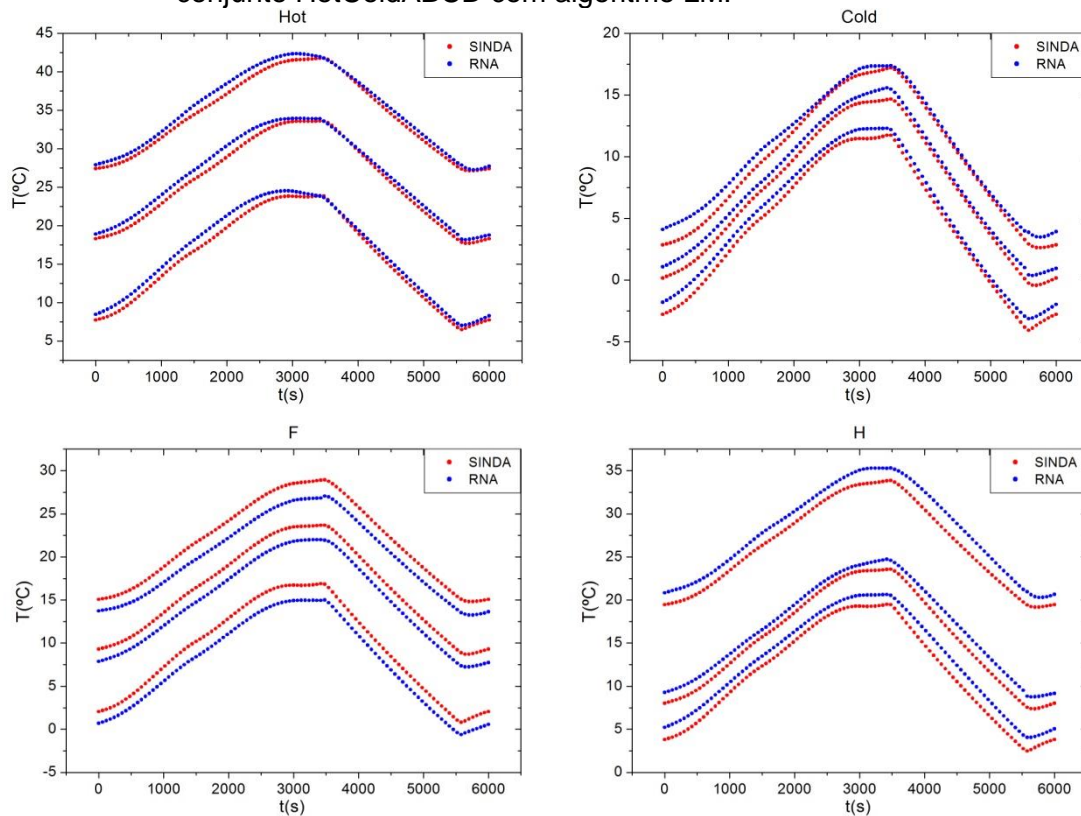
Figura 4.21. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo LM.



Fonte: Produção do autor.

A Fig. 4.22 ilustra as curvas de temperatura para o conjunto HotColdABCD.

Figura 4.22. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo LM.



Fonte: Produção do autor.

Na Tabela 4.14 constam os valores dos erros médios de generalização para todos os conjuntos de dados treinados com o algoritmo LM.

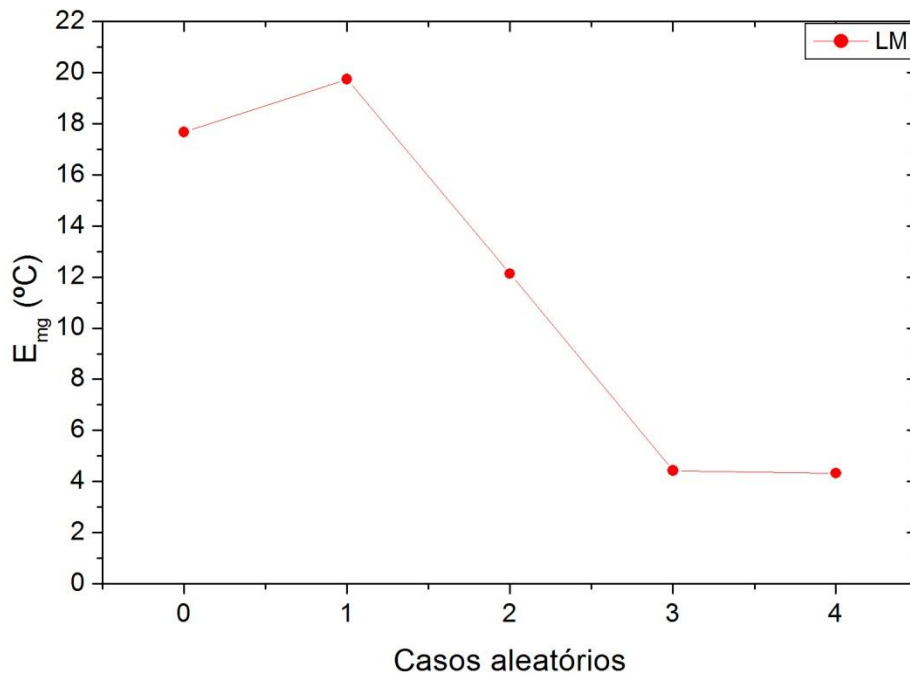
Tabela 4.14. Erros médios de generalização para o algoritmo LM.

Conjunto	Erro Médio ($^{\circ}\text{C}$)					Média Total
	E	F	G	H	I	
HotCold	4,3	19,5	15,6	25,3	23,6	17,7
HotColdA	14,4	16,3	13,0	31,6	23,3	19,7
HotColdAB	5,8	13,6	6,8	17,4	16,9	12,1
HotColdABC	2,3	4,1	1,3	9,1	5,3	4,4
HotColdABCD	1,5	3,4	2,9	9,4	4,5	4,3

Fonte: Produção do autor.

A evolução do erro médio de generalização para cada conjunto pode ser vista na Fig. 4.23. As linhas são apenas guias para os olhos.

Figura 4.23. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo LM.



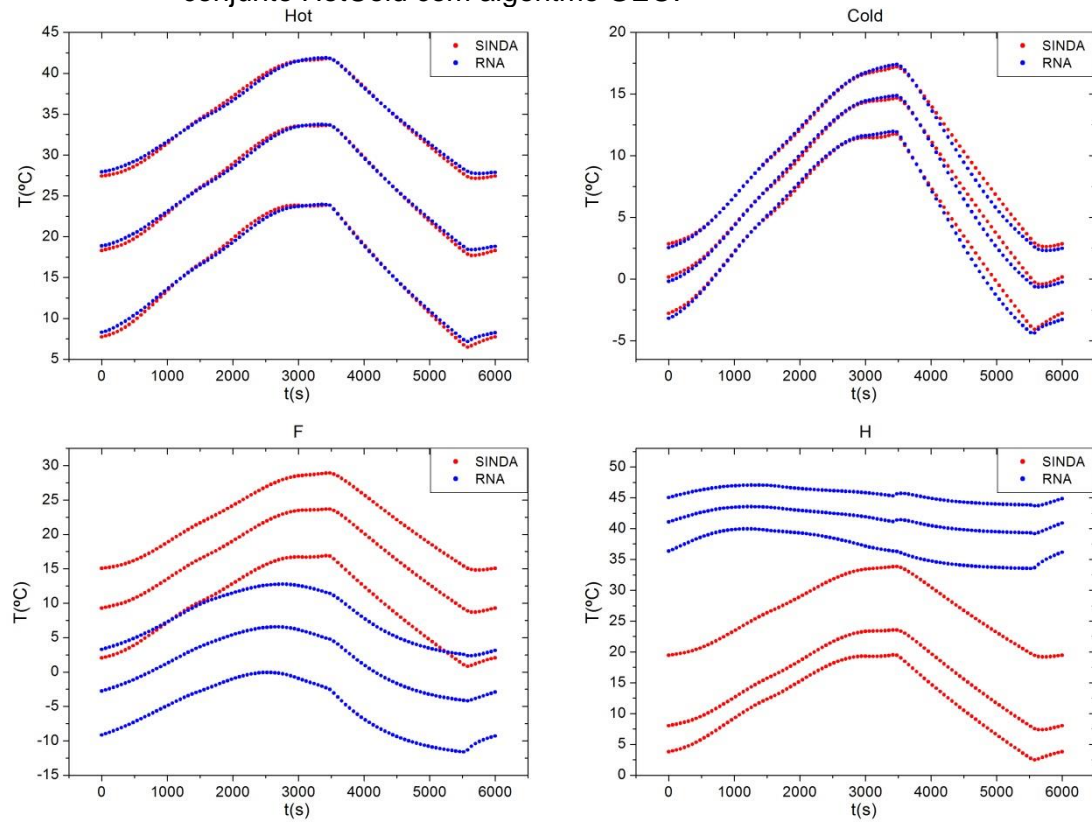
Fonte: Produção do autor.

4.2.4 Resultados para o algoritmo Otimização Extrema Generalizada

Nesta Subseção são apresentados os resultados para o algoritmo GEO. Conforme explicado na Subseção 3.4, o algoritmo GEO foi utilizado em conjunto com o algoritmo GD. Os parâmetros utilizados para o treinamento com o algoritmo GEO foram: tolerância $\tau = 0,00001$ e parâmetro ajustável $\lambda = 0,01$; e para o algoritmo GD foram: tolerância $\tau = 0,00001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Em caso da rede não convergir para a tolerância especificada, o treinamento de cada algoritmo foi interrompido após 10^6 épocas.

A Fig. 4.24 contém os resultados para o conjunto HotCold.

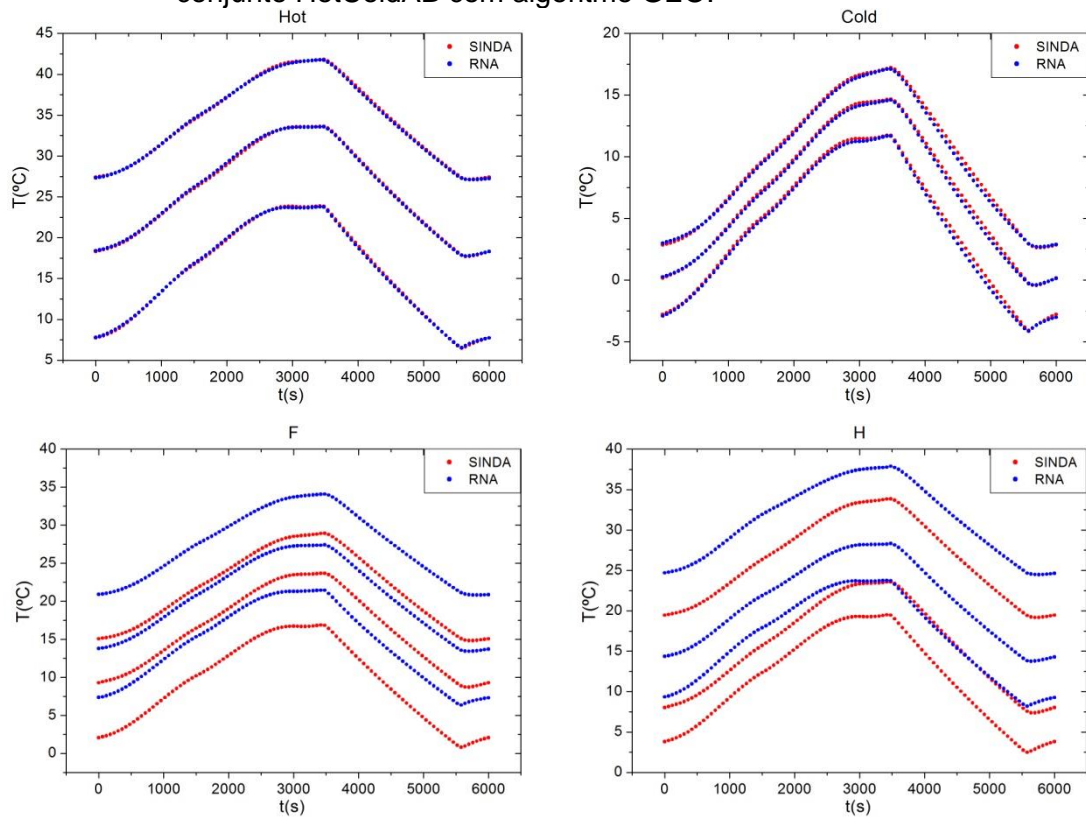
Figura 4.24. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotCold com algoritmo GEO.



Fonte: Produção do autor.

A comparação para o conjunto HotColdAB está ilustrada na Fig. 4.25.

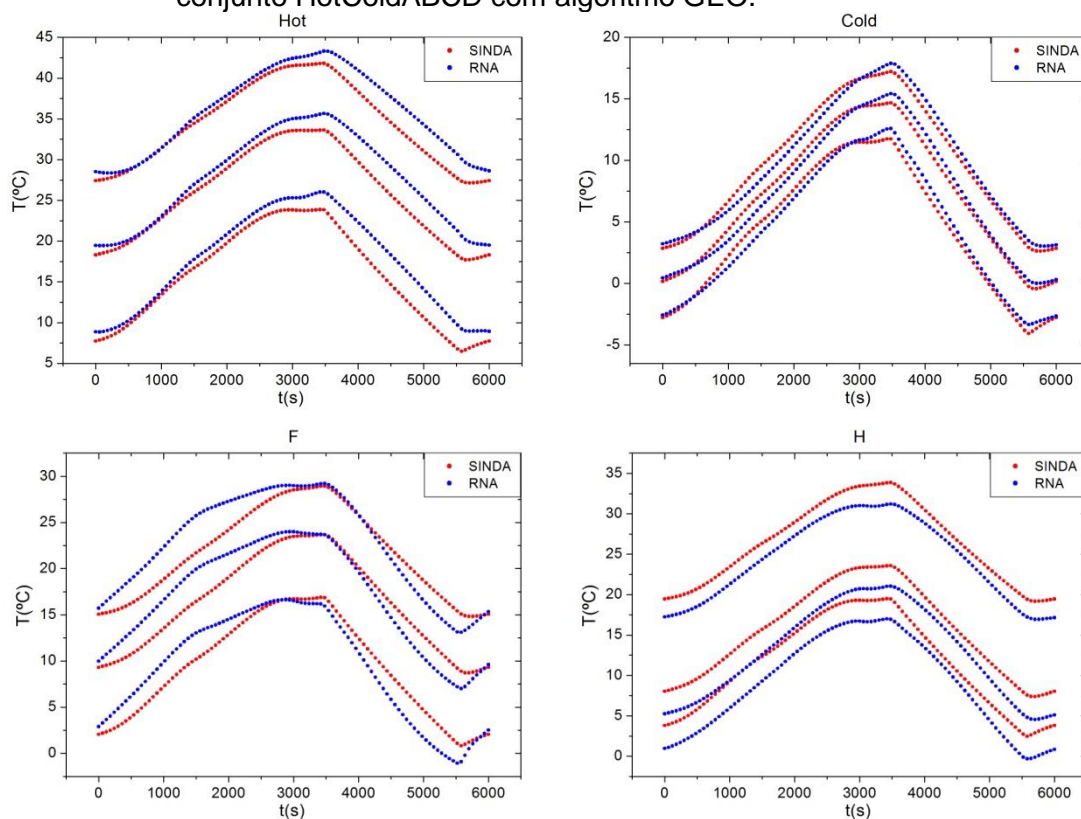
Figura 4.25. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdAB com algoritmo GEO.



Fonte: Produção do autor.

Finalmente, na Fig. 4.26 pode ser vista a comparação entre as curvas de temperatura para o conjunto HotColdABCD.

Figura 4.26. Comparação das curvas de temperatura entre SINDA e RNA para o conjunto HotColdABCD com algoritmo GEO.



Fonte: Produção do autor.

Os valores dos erros médios de generalização para todos os conjuntos treinados com o algoritmo GEO estão organizados na Tabela 4.15.

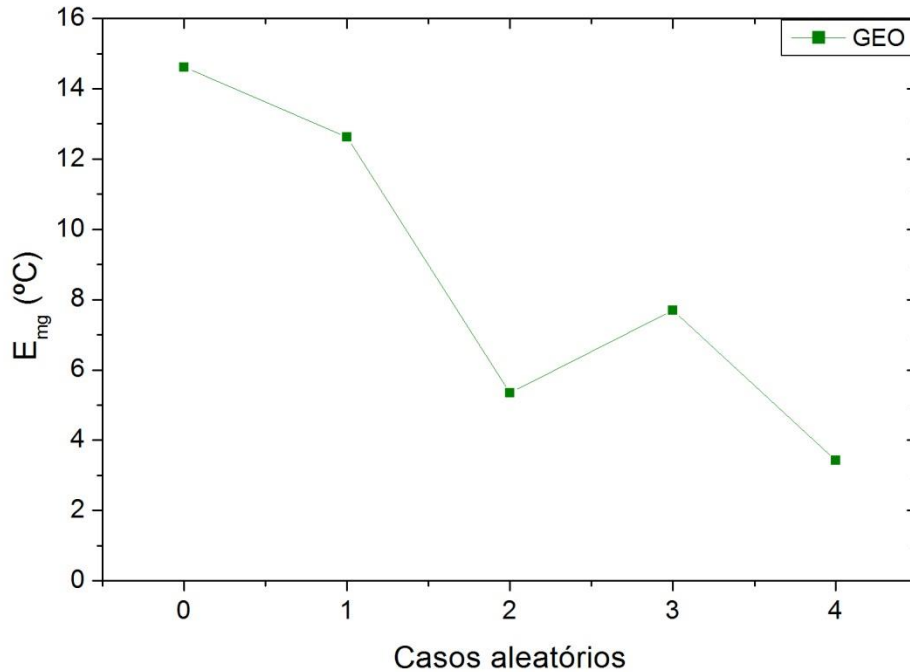
Tabela 4.15. Erros médios de generalização para o algoritmo GEO.

Conjunto	Erro Médio ($^{\circ}\text{C}$)					Média Total
	E	F	G	H	I	
HotCold	14,0	14,2	13,5	15,5	15,9	14,6
HotColdA	14,6	12,2	6,8	14,3	15,2	12,6
HotColdAB	2,7	6,5	1,1	7,4	9,0	5,3
HotColdABC	4,8	10,4	2,5	8,7	11,9	7,7
HotColdABCD	3,1	2,9	1,8	4,6	5,3	3,4

Fonte: Produção do autor.

A evolução do erro médio de generalização para cada conjunto pode ser vista na Fig. 4.27. As linhas são apenas guias para os olhos.

Figura 4.27. Erro médio de generalização em função do número de cenários aleatórios adicionados para o algoritmo GEO.



Fonte: Produção do autor.

4.2.5 Comparação entre os algoritmos

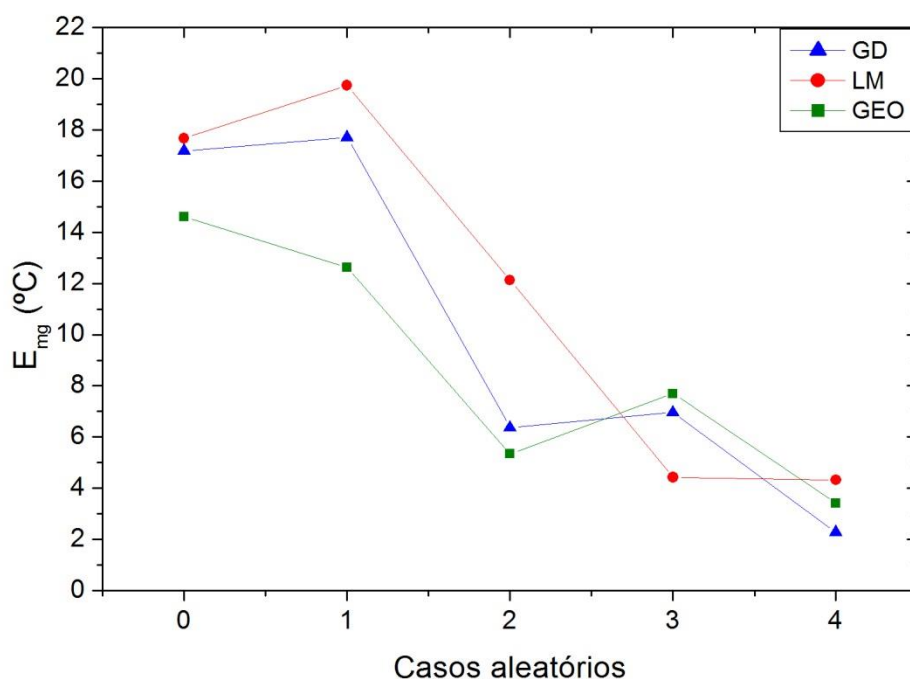
Comparando os resultados para os três algoritmos, observa-se que o comportamento geral é bem semelhante. Conforme são adicionados novos cenários no treinamento, a rede neural vai adquirindo cada vez mais informação sobre a relação entre as entradas e as saídas esperadas e, conseqüentemente, o erro de generalização vai sendo reduzido.

Em alguns casos, a inclusão de um novo cenário aumentou um pouco o erro de generalização ao invés de diminuir. Este fenômeno é atribuído ao fato de os novos cenários incluídos não estarem relacionados diretamente aos cenários

usados para generalizar. Em outras palavras, o novo cenário incluído pode levar a rede em uma direção que não seria a mais adequada para reduzir o erro de generalização. Contudo, observa-se que com a adição de um outro cenário, o erro de generalização volta a cair, o que corrobora a estratégia adotada.

Na Fig. 4.28 pode ser vista a comparação da evolução do erro de generalização para os três algoritmos. As linhas são apenas guias para os olhos.

Figura 4.28. Comparação do erro médio de generalização para os três algoritmos.



Fonte: Produção do autor.

Inicialmente, o GD e o LM apresentam erros mais elevados (acima de 17°C) e o GEO possui erros menores (abaixo de 15°C), até a inclusão do terceiro cenário aleatório. Esta vantagem inicial do GEO está relacionada ao fato deste ser um algoritmo de busca global, que evita os mínimos locais com mais eficiência na presença de poucos cenários aleatórios.

O erro de generalização para o GD apresenta uma grande oscilação com a adição de novos cenários. Isso talvez esteja relacionado com a tendência maior de o GD ficar retido em mínimos locais, mesmo que os valores sejam resultantes da média de 10 rodadas com diferentes inicializações.

A partir da adição do terceiro cenário, a vantagem inicial do GEO deixa de ser relevante, com maior vantagem para o LM. Com a adição do quarto cenário, os algoritmos passam a ter comportamento semelhante, com vantagem de cerca de 2°C para o GD em comparação com o LM e de cerca de 1°C em comparação com o GEO. Isso é um indicativo de que a inclusão de mais informação relevante no treinamento das redes neurais as torna mais resilientes às deficiências intrínsecas de cada algoritmo de treinamento.

Como o algoritmo GEO foi executado em conjunto com o GD, seria esperado que a combinação desses algoritmos sempre apresentasse resultados melhores ou pelo menos iguais ao algoritmo GD quando aplicado isoladamente. Contudo, o fato do treinamento atingir um mínimo mais baixo não significa necessariamente que a rede irá apresentar melhores resultados de generalização para cenários não treinados. Em outras palavras, um menor erro de treinamento nem sempre resulta em um menor erro de generalização.

A Tabela 4.16 contém os valores para os erros de generalização, para permitir a comparação quantitativa entre os algoritmos.

Tabela 4.16. Erros médios de generalização para os três algoritmos.

Conjunto	Erros Médios (°C)		
	GD	LM	GEO
HotCold	17,18	17,67	14,61
HotColdA	17,71	19,74	12,63
HotColdAB	6,37	12,13	5,34
HotColdABC	6,96	4,43	7,70
HotColdABCD	2,28	4,33	3,42

Fonte: Produção do autor.

Após a inclusão dos 4 cenários aleatórios, o GD apresenta o melhor resultado entre os três algoritmos (2,28°C). O GEO é o segundo melhor, com um erro médio de 3,42°C, um pouco mais alto do que o apresentado pelo GD. O LM, por sua vez, apresenta um erro ligeiramente mais alto (4,33°C), mas que também pode ser considerado um erro baixo dentro dos critérios estabelecidos.

Além dos erros médios, podem ser analisados também os melhores casos de cada algoritmo, isto é, os casos nos quais os algoritmos apresentaram os menores erros de generalização. Estes menores erros são mostrados na Tabela 4.17.

Tabela 4.17. Menores erros de generalização para os três algoritmos.

Conjunto	Menores Erros (°C)		
	GD	LM	GEO
HotCold	14,37	5,51	12,41
HotColdA	14,94	4,80	9,20
HotColdAB	1,46	1,93	1,39
HotColdABC	4,98	0,79	5,4
HotColdABCD	0,87	1,08	2,03

Fonte: Produção do autor.

O algoritmo GD continua apresentando o melhor resultado com a inclusão do quarto cenário aleatório, com um erro médio de apenas 0,87°C. Para esse mesmo treinamento, o maior erro de generalização observado foi de 2,8°C. Ambos os valores estão abaixo da meta de 5°C, referente ao requisito definido para o Simulador Operacional. O algoritmo LM, desta vez, possui o segundo melhor resultado ao final e apresenta erros baixos de generalização desde o conjunto HotCold.

A análise desses resultados nos leva à conclusão de que a definição dos dados de entrada utilizados e a estratégia de treinamento das redes neurais é mais importante do que a escolha do algoritmo de treinamento.

A seleção e otimização do algoritmo de treinamento não deixam de ser importantes. Porém, de nada serve um bom algoritmo de treinamento se a informação que está sendo apresentada à RNA não contém o conhecimento relevante para que esta possa aprender e generalizar.

4.2.6 Satélite Amazonia-1

Nesta subseção são apresentados os resultados obtidos de generalização para cenários não treinados, para o Satélite Amazonia-1. Em função do aumento da escala e da complexidade desse satélite, foi decidido utilizar apenas o algoritmo Gradiente Descendente para treinamento da rede neural. Essa escolha se deve ao fato deste algoritmo ter apresentado melhores resultados para o CubeSat, além de possuir um menor custo computacional em comparação com os demais algoritmos.

O modelo utilizado para as simulações do Amazonia-1 no software SINDA é o próprio modelo usado atualmente pela equipe de engenharia térmica do INPE para o desenvolvimento do Subsistema de Controle Térmico desse satélite. Diferentemente do modelo do CubeSat que possuía apenas 21 nós térmicos, dos quais foram utilizados 9 nós para o treinamento da RNA, o modelo do Amazonia-1 possui 11736 nós, dos quais foram selecionados 50 para o treinamento.

Esse modelo, inclusive, já foi validado pelo Teste de Balanço Térmico (TBT) do satélite, o que garante a sua fidelidade com o sistema real que será colocado em órbita. Mais detalhes sobre o satélite Amazonia-1 podem ser vistos na Subseção 4.1.3.

A Fig. 4.29 ilustra a arquitetura da RNA utilizada. As variáveis hr_1 a hr_6 representam as taxas de aquecimento nas 6 superfícies externas do satélite, as variáveis Eq_1 a Eq_{39} dizem respeito à potência dos equipamentos internos selecionados para treinamento e as variáveis $T_1(z^{-1})$ a $T_{50}(z^{-1})$ são as temperaturas geradas na saída da rede para o instante imediatamente anterior

($\Delta t = 1$ min). Dessa forma, tem-se um total de 95 parâmetros de entrada na rede. Para fins de comparação, no treinamento com o modelo do CubeSat eram apenas 18 parâmetros na camada de entrada.

A rede é composta ainda por 20 neurônios na camada oculta. Foram realizados testes com 10 a 70 neurônios e o valor de 20 neurônios foi o que apresentou maior estabilidade nos resultados e, ao mesmo tempo, um baixo erro de generalização.

Os parâmetros utilizados para treinamento foram: tolerância $\tau = 0,00001$, taxa de aprendizagem $\eta = 0,01$ e constante de momentum $\alpha = 0,5$. Em caso da rede não convergir para a tolerância especificada, o treinamento foi interrompido após 10^6 épocas.

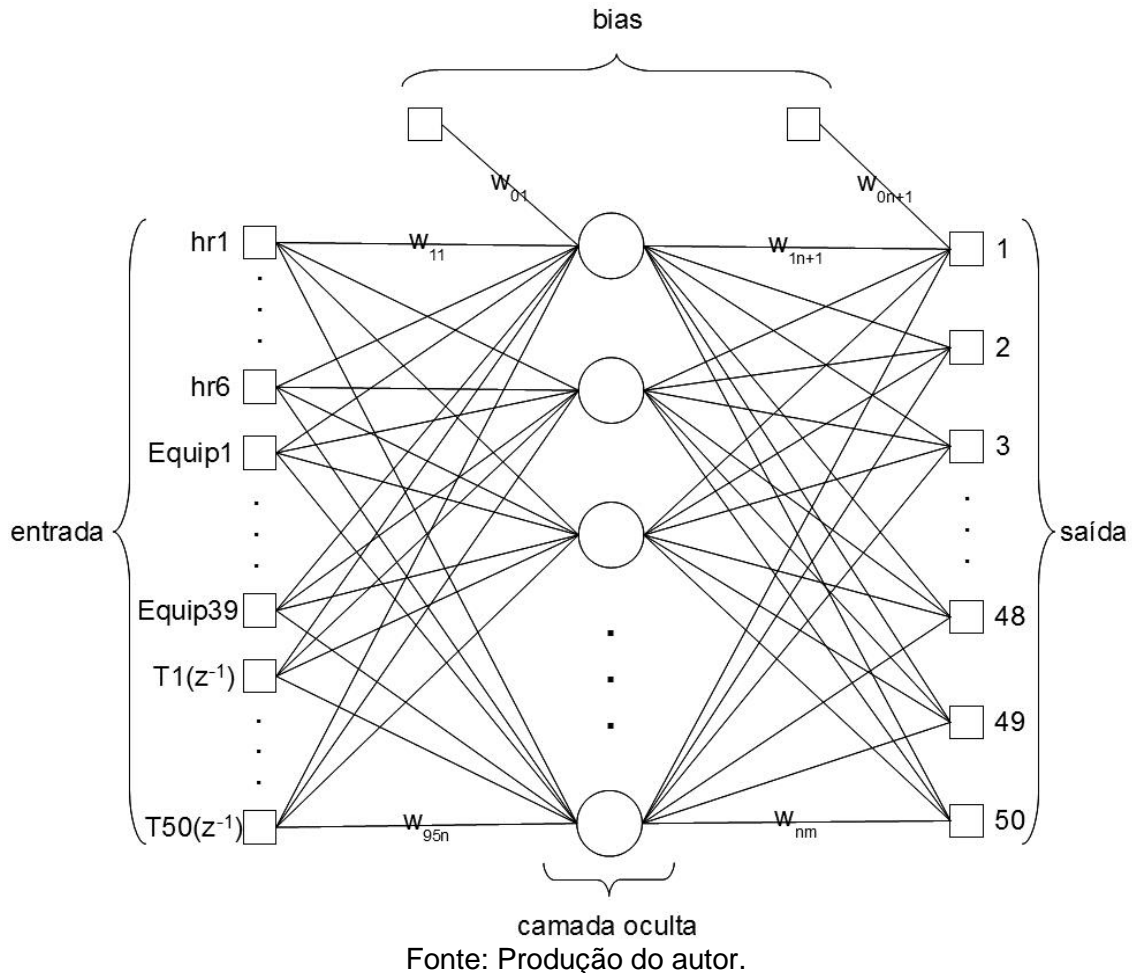
Importante notar que o procedimento de recorrência é diferente do utilizado no caso do CubeSat, onde foram usadas as temperaturas esperadas do instante anterior, extraídas do conjunto de treinamento, e não as temperaturas geradas na saída da rede, como foi feito para o presente caso.

Foram gerados ao todo 12 cenários no SINDA, sendo dois deles os cenários nominais quente (Hot) e frio (Cold), além de dez cenários cujos parâmetros foram definidos de forma aleatória, dentro dos intervalos nominais para cada variável, denominados de A a J. Devido à grande quantidade de parâmetros que definem esses cenários, os mesmos foram dispostos no Apêndice B.

Da mesma forma como no caso do CubeSat, primeiro foi realizado o treinamento com os cenários Hot e Cold, em seguida foram sendo adicionados os cenários aleatórios de forma gradativa e em treinamentos independentes. O conjunto de treinamento para cada cenário possui 101 padrões. Dessa forma, o conjunto de treinamento inicial (HotCold) possui 202 padrões e o conjunto de treinamento final, com a inclusão de 5 cenários aleatórios (HotColdABCDE) possui 707 padrões.

Para cada conjunto de treinamento foram realizadas 10 rodadas independentes com diferentes inicializações aleatórias dos pesos, com o objetivo de reduzir, na média, a tendência do algoritmo GD de ficar preso em mínimos locais.

Figura 4.29. Arquitetura da RNA utilizada para a generalização para o satélite Amazonia-1.

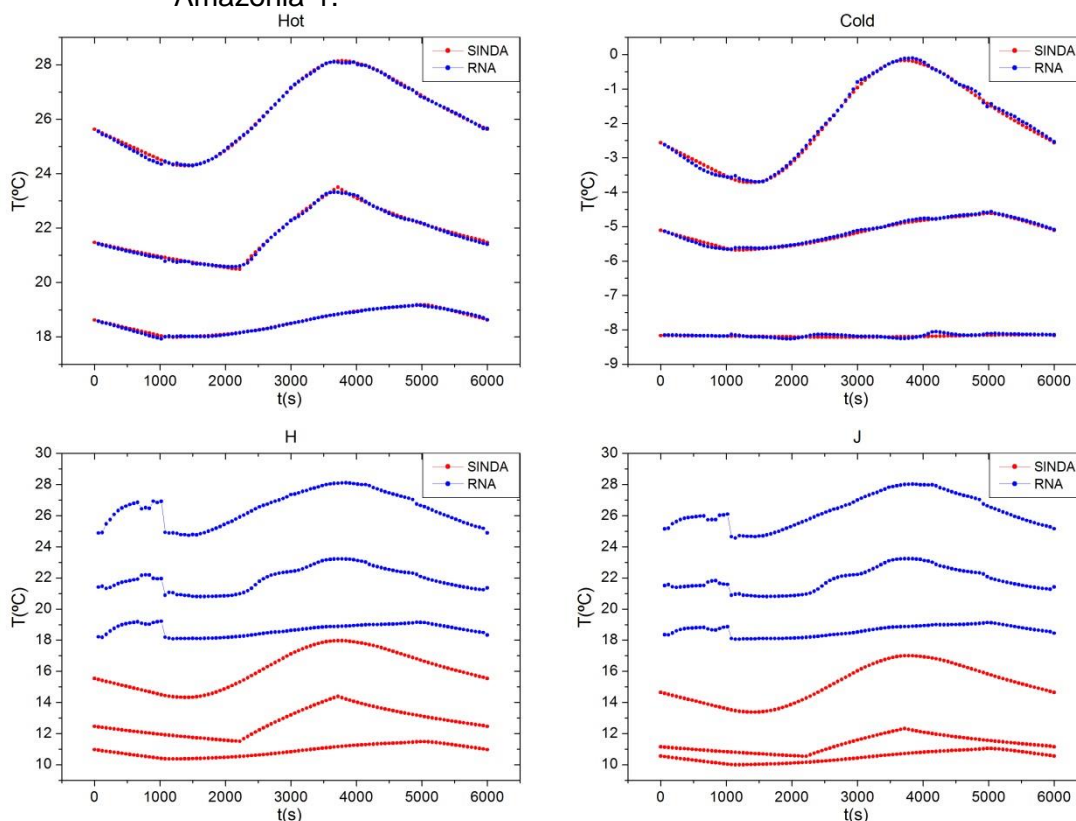


Adicionalmente, foram selecionados dois cenários (F e G) para serem usados durante o treinamento para validação. O procedimento de validação consiste em, a cada intervalo pré-definido de épocas (neste caso foram utilizadas 1000 épocas), verificar qual o erro dos valores gerados pela RNA para um conjunto de dados não utilizados no treinamento. Dessa forma, o conjunto completo de dados é separado em três grupos: treinamento, validação e teste. Ao final do treinamento, são selecionados os pesos da rede que apresentaram o menor erro de validação. Esse procedimento evita o chamado

overfitting, que seria a adaptação excessiva da rede ao conjunto de treinamento, prejudicando a sua capacidade de generalização (HAYKIN, 2001).

Na Fig. 4.30, são mostradas as curvas de temperatura obtidas para o conjunto HotCold.

Figura 4.30. Curvas de temperatura referentes ao o conjunto HotCold para o satélite Amazonia-1.



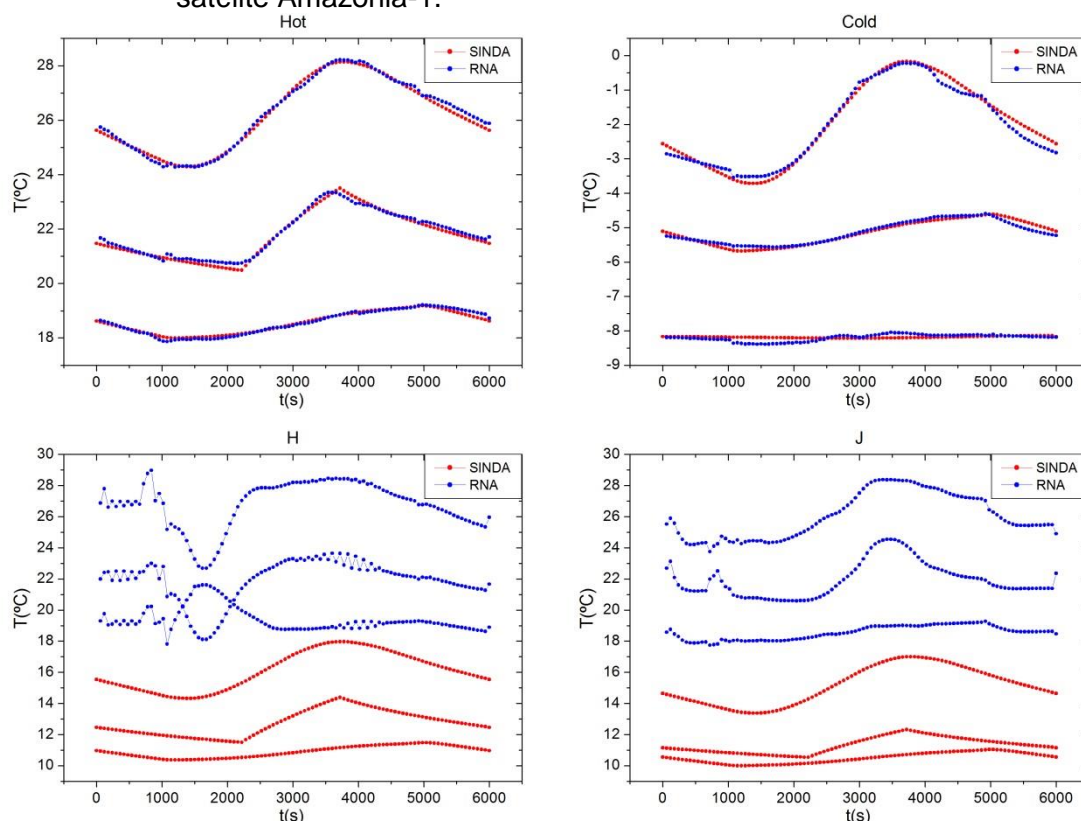
Fonte: Produção do autor.

Essas curvas de temperatura referem-se a 3 equipamentos TCE (*Thruster Control Electronics*; Eletrônica de Controle dos Propulsores), Gyro_ICU (*Gyro Inertial Control Unit*; Unidade de Controle Inercial do Sensor Giroscópico) e SPE_WFI (*WFI Signal Processing Electronics*; Eletrônica de Processamento de Sinais da câmera WFI) e 4 cenários diferentes (Hot, Cold, H e J), para a rede neural treinada apenas com os cenários Hot e Cold. Esses equipamentos foram selecionados por apresentar uma diferença de temperatura suficiente para permitir a sua visualização de forma independente em um mesmo gráfico.

Como esperado, as temperaturas geradas pela RNA para os casos treinados Hot e Cold apresentam boa sobreposição com as simuladas no SINDA. Já para os casos não treinados H e J, a diferença entre as curvas é significativa,

A seguir, na Fig. 4.31, podem ser vistas as curvas de temperatura para o conjunto de cenários HotColdAB.

Figura 4.31. Curvas de temperatura referentes ao o conjunto HotColdAB para o satélite Amazonia-1.

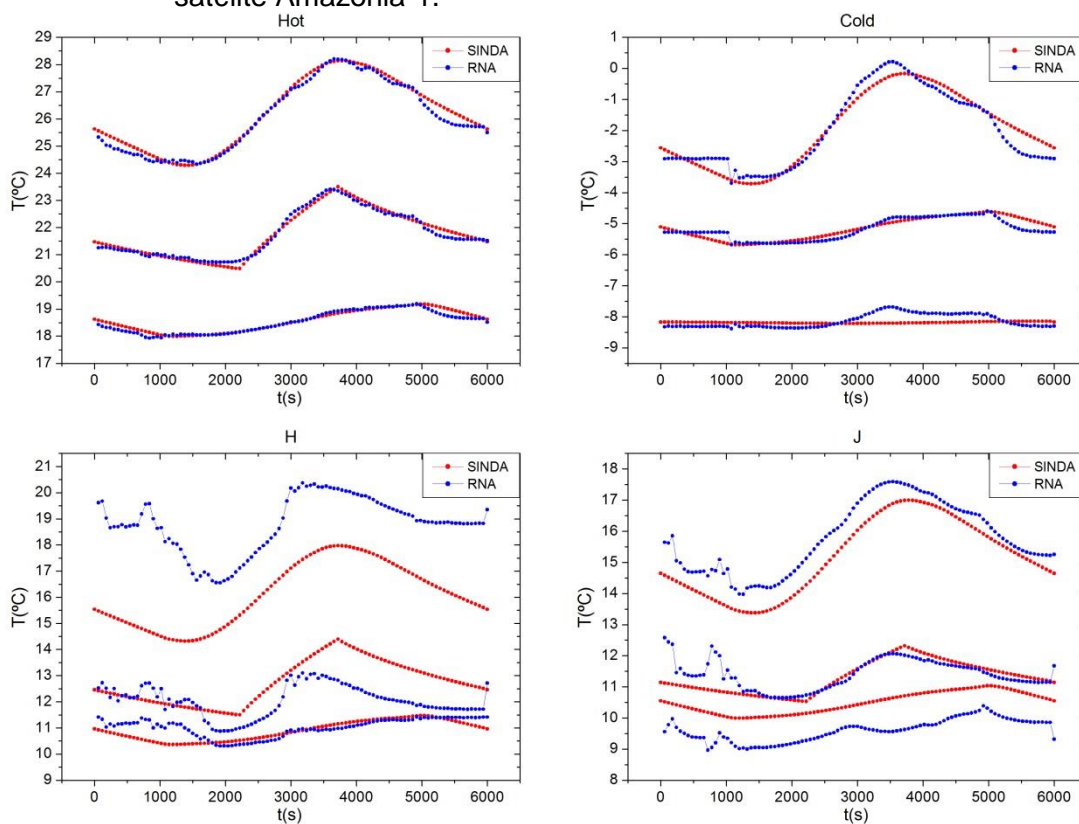


Fonte: Produção do autor.

As curvas produzidas pela RNA para os casos Hot e Cold continuam apresentando um erro pequeno em comparação com as geradas pelo SINDA, apesar de haver diferenças perceptíveis em alguns pontos. As curvas para os cenários H e J, por sua vez, continuam distantes e apresentam algumas oscilações (ou ruído) localizadas.

Os resultados para o conjunto HotColdABCDE são mostrados na Fig. 4.32.

Figura 4.32. Curvas de temperatura referentes ao o conjunto HotColdABCDE para o satélite Amazonia-1.



Fonte: Produção do autor.

Apesar do erro pequeno, as curvas para os cenários treinados apresentam algumas diferenças mais evidentes, especialmente para o cenário Cold. O aumento do erro para os casos treinados, conforme são incluídos novos cenários, se deve ao fato que a rede neural deve se adaptar a todos os dados de treinamento ao mesmo tempo. Em outras palavras, quanto maior o número de cenários apresentados durante o treinamento, maior será a dificuldade da rede se adaptar a cada um individualmente.

Considerando agora os cenários não treinados (H e J), observa-se que a diferença entre as curvas geradas pela RNA e pelo SINDA foi reduzida. As oscilações localizadas também diminuíram sua intensidade, porém ainda permanecem em algumas regiões. Esse ruído se deve, principalmente, ao fato que, na maior parte dos cenários, os equipamentos iniciam a órbita ligados e, em pontos específicos os mesmos são colocados em modo *standby*.

Os resultados para os conjuntos HotColdA, HotColdABC e HotColdABCD não foram mostrados para evitar repetições excessivas.

A Tabela 4.18. contém os valores dos erros médios de generalização para todos os conjuntos de dados, considerando apenas os casos de teste H, I e J, não utilizados no treinamento.

Tabela 4.18. Erros médios de generalização para o satélite Amazonia-1, contendo apenas os cenários usados para teste.

Conjunto	Erro médio (°C)			Média Total
	H	I	J	
HotCold	9.4	7.3	8.2	8.3
HotColdA	5.4	4.8	5.0	5.1
HotColdAB	10.4	6.5	7.8	8.3
HotColdABC	2.2	9.4	2.4	4.7
HotColdABCD	5.4	5.3	4.7	5.1
HotColdABCDE	4.5	2.7	4.8	4.0

Fonte: Produção do autor.

A Tabela 4.19. contém os mesmos valores dos erros médios da tabela anterior, porém agora incluindo também os casos F e G usados para validação.

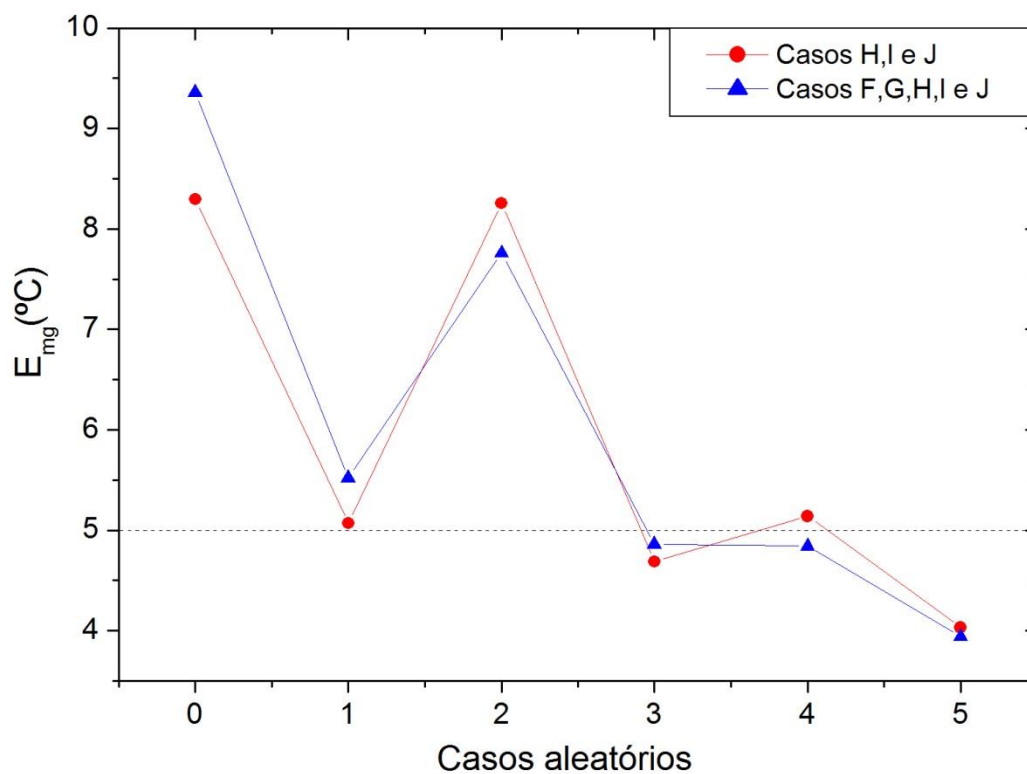
Tabela 4.19. Erros médios de generalização para o satélite Amazonia-1, incluindo os cenários usados para validação.

Conjunto	Erro médio (°C)					Média Total
	F	G	H	I	J	
HotCold	11.8	10.1	9.4	7.3	8.2	9.4
HotColdA	8.3	4.1	5.4	4.8	5.0	5.5
HotColdAB	5.4	8.6	10.4	6.5	7.8	7.8
HotColdABC	5.5	4.7	2.2	9.4	2.4	4.9
HotColdABCD	4.5	4.2	5.4	5.3	4.7	4.8
HotColdABCDE	5.6	2.0	4.5	2.7	4.8	3.9

Fonte: Produção do autor.

A Fig. 4.33 demonstra a evolução do erro médio de generalização (E_{mg}) para cada conjunto. As duas curvas foram construídas a partir dos dados das Tabelas 4.18 e 4.19. As linhas contínuas são apenas guias para os olhos e a linha tracejada representa o limite de 5°C, referente ao requisito de que o erro de generalização deve ser inferior a esse valor de temperatura.

Figura 4.33. Erro médio de generalização em função do número de cenários aleatórios adicionados para o Amazonia-1.



Fonte: Produção do autor.

Ambas as curvas apresentam comportamento bastante semelhante. Apesar da queda abrupta com a inclusão do primeiro caso aleatório, o E_{mg} não foi reduzido abaixo dos 5°C. Isso ocorreu somente após a adição do terceiro cenário, voltando a aumentar um pouco para o quarto (considerando os casos H, I e J apenas) e caindo novamente para valores próximos de 4°C após a inclusão do quinto cenário aleatório.

Conforme citado anteriormente, o resultado descrito acima se refere à média obtida a partir de 10 inicializações independentes dos pesos. Isso significa que em algumas rodadas de treinamento foi possível obter erros ainda mais baixos. Por conseguinte, os menores erros médios de generalização, obtidos para a melhor rodada de treinamento para o conjunto HotColdABCDE, assim como os erros máximos (E_{max}) observados para essa rodada, são mostrados na Tabela 4.20.

Tabela 4.20. Menores erros médios e erros máximos de generalização para o melhor caso de treinamento para o satélite Amazonia-1.

Cenário	Menor E_{mg} (°C)	E_{max} (°C)
F	3.5	15.0
G	1.8	10.2
H	1.7	19.2
I	2.4	12.3
J	2.0	18.3
Média geral	2.3	

Fonte: Produção do autor.

Como pode ser visto nessa tabela, o menor E_{mg} obtido para os cenários não treinados foi de 2,3°C. Este valor está bem abaixo do requisito de 5°C, o que significa que as temperaturas geradas pela RNA ficaram bem próximas dos valores esperados, apesar do ruído apresentado nas curvas. Por outro lado, em alguns pontos das curvas foram observados erros maiores, sendo que o maior erro observado foi de 19,2°C para um dos nós e em um ponto específico da órbita para o cenário H.

4.3 Comparação com Outras Metodologias

Nesta seção a abordagem proposta e os resultados obtidos para o CubeSat e o satélite Amazonia-1 na Seção 4.2 são comparados com duas metodologias diferentes encontradas na literatura. No caso do CubeSat é realizada a comparação com a metodologia proposta por Manon et al. (2018), denominada Método de Caracterização Térmica de Satélites. Em seguida, os resultados para o Amazonia-1 são comparados com uma interpolação realizada por meio do método de Regressão Linear Multivariada.

4.3.1 Comparação com o Método de Caracterização Térmica de Satélites

Esta subseção apresenta a comparação da abordagem proposta nesta Tese para simulação térmica em simuladores operacionais de satélites com o Método de Caracterização Térmica de Satélites, proposto por membros da agência espacial francesa (CNES) em Manon et al. (2018) (ver Subseção 2.2.1).

Apesar de o Cubesat utilizado no trabalho de Manon et al. (2018) e o Cubesat modelado nesta Tese serem diferentes, ambos os trabalhos propõem uma abordagem para o cálculo da variação de temperatura de satélites a ser aplicada a um Simulador Operacional. Considerando que os Cubesats são satélites construídos com padrões de hardware bem definidos, a comparação aqui é feita apenas para os resultados obtidos para o Cubesat (ver Subseção 4.1.2).

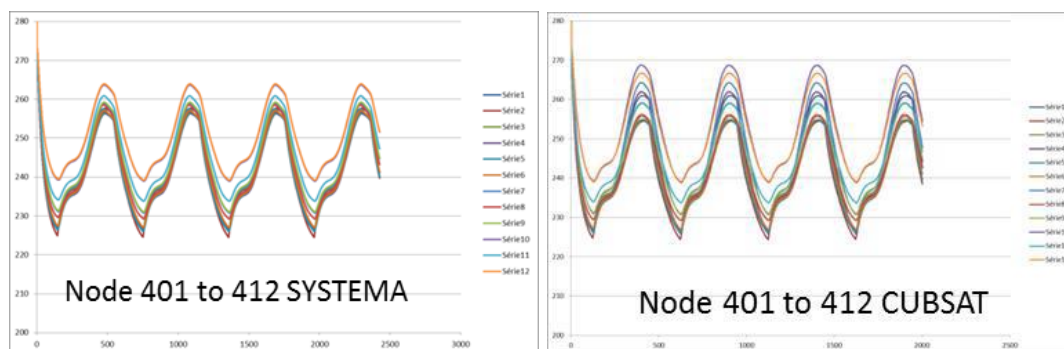
Na abordagem citada, para cada nó do modelo e cada direção de uma fonte, o fluxo total de calor recebido pelo nó é calculado, que inclui o fluxo direto e o fluxo refletido (incluindo sombras e reflexões múltiplas). O fluxo total é normalizado pelo fluxo direto, tornando-o independente da potência da fonte. Esta operação é realizada para todas as direções discretizadas de uma esfera centralizada no nó.

Para esse cálculo é utilizado um pacote de análise térmica chamado THERMICA-SYSTEMA (AIRBUS, 2018), semelhante ao SINDA, e os coeficientes resultantes das complexas trocas radiativas de calor entre os nós do modelo, fontes de calor e dissipadores térmicos, são armazenados em tabelas. Além disso, é realizada uma migração do modelo térmico nodal do satélite para modelos de nós integráveis no simulador.

No Simulador, o valor da atitude e da posição orbital do satélite permitem determinar a direção das fontes de calor e, conseqüentemente, interpolar, nas tabelas de coeficientes, o valor do fluxo total recebido.

Esse método foi implementado e validado em um modelo de um CubeSat com cerca de 30 nós térmicos. De acordo com os autores, os resultados obtidos mostram que a modelagem térmica no Simulador do CubeSat segue perfeitamente as mudanças na temperatura calculadas pelo software THERMICA, como pode ser visto na Fig. 4.34.

Figura 4.34. Comparação entre as curvas de temperatura obtidas pelo THERMICA-SYSTEMA e o CubeSat.



Fonte: Manon et al. (2018).

A diferença máxima de temperatura observada pelos autores foi de 5°C. Nos resultados obtidos nesta Tese, a partir da RNA para o CubeSat, particularmente com o algoritmo GD, foi possível atingir um erro médio de generalização de 0,87°C (vide Tabela 4.17) e um erro máximo de 2,8°C. Apesar do modelo do CubeSat utilizado no trabalho de Manon et al. (2018) ser diferente do modelo utilizado no presente trabalho, a comparação dos

resultados obtidos indica que a abordagem proposta nesta Tese é uma alternativa competitiva para o desenvolvimento do modelo térmico em Simuladores Operacionais.

4.3.2 Comparação com Regressão Linear Multivariada

Nesta subseção, os resultados obtidos a partir da rede neural, para as temperaturas do satélite Amazonia-1, são comparados com uma interpolação matemática realizada utilizando o método de Regressão Linear Multivariada (RLM), semelhante à aplicada tradicionalmente na literatura (PERPIÑÁN, 1994).

O método RLM é um método estatístico utilizado para modelar a relação entre um conjunto de variáveis independentes e outro de variáveis dependentes relacionadas. Supondo que se dispõe de p variáveis resposta:

$$Y_1, \dots, Y_p. \quad (4.1)$$

estando associado a cada qual um grupo de k variáveis explicativas $\{X_{1j}, \dots, X_{kj}\}$, onde $j = 1, \dots, p$, pode ser definido para cada uma das p regressões um modelo linear geral (NOGUEIRA, 2007; MAIA, 2017):

$$Y_j = \beta_0 + X_{1j}\beta_{1j} + \dots + X_{kj}\beta_{kj} + \varepsilon_j, \quad j = 1, \dots, p. \quad (4.2)$$

onde $\beta_j = [\beta_{0j}, \dots, \beta_{kj}]^T$ é um vetor de parâmetros e ε_j é o vetor de erros associados.

De forma geral, o modelo pode ser escrito na forma matricial:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (4.3)$$

A matriz $\boldsymbol{\beta}$ pode ser calculada por meio da seguinte equação:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y}). \quad (4.4)$$

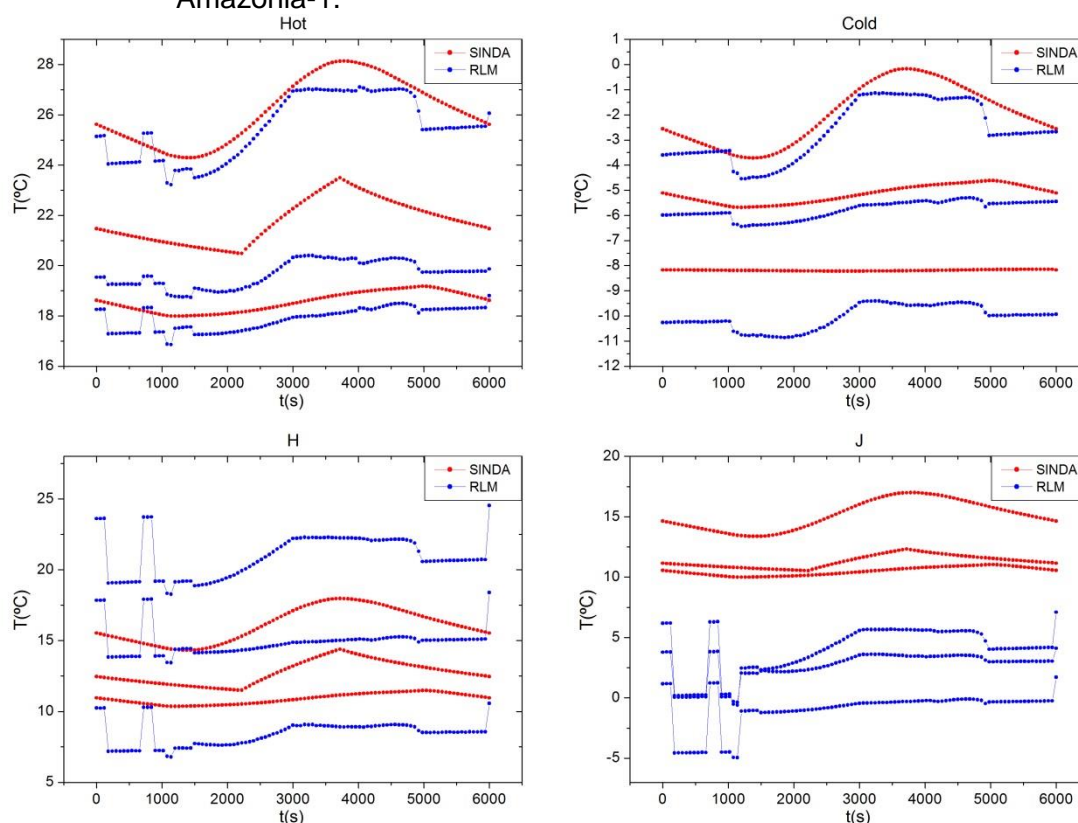
Esse modelo foi resolvido para o mesmo conjunto de cenários utilizados para o treinamento da RNA (Hot, Cold, A, B, C, D e E) e foi utilizado para calcular as temperaturas para os cenários F, G, H, I e J. Para o cálculo de inversão da matriz, foi utilizada a biblioteca de funções de álgebra linear

LAPACK (Linear Algebra Package), mais especificamente a sua interface para linguagem C, denominada LAPACKE (LAPACKE, 2018).

A Fig. 4.35 mostra as curvas de temperatura calculadas a partir do método RLM.

As curvas geradas para os cenários Hot e Cold, que foram utilizados para o cálculo da regressão apresentaram um erro médio de 1,3°C e 1,2°C, respectivamente. Porém, visualmente, as curvas apresentaram uma maior divergência em comparação com as obtidas com a RNA (vide Fig. 4.33). Já as curvas para os cenários que não foram incluídos no cálculo da regressão (interpolados) apresentaram uma maior discrepância do que quando comparados com a RNA.

Figura 4.35. Curvas de temperatura calculadas pelo método RLM para o satélite Amazonia-1.



Fonte: Produção do autor.

O ruído também fica evidente nas curvas geradas pelo método RLM, especialmente nos instantes iniciais da órbita. Assim como no caso da rede neural, isso se deve ao fato que na maior parte dos cenários os equipamentos iniciam a órbita ligados e, em pontos específicos os mesmos são colocados em modo *standby*.

Os erros médios e máximos de interpolação podem ser vistos na Tabela 4.21.

Tabela 4.21. Erros médios e máximos de interpolação obtidos com o método RLM para o satélite Amazonia-1.

Cenário	Erro Médio (°C)	E_{max} (°C)
F	6.4	30.9
G	1.9	14.4
H	4.3	22.5
I	8.1	34.9
J	10.4	37.4
Média geral	6.2	

Fonte: Produção do autor.

Como pode ser visto nessa tabela, com o uso do método RLM o erro médio de interpolação obtido foi de 6,2°C, o que é superior ao limite de 5°C estabelecido como requisito para o Simulador Operacional, apesar de ter ficado bem próximo desse valor. Os erros máximos de interpolação observados, por outro lado, chegaram a atingir o valor de 37,4°C para o cenário J.

Comparando esses valores com o resultado obtido com a RNA mostrados na Tabela 4.20, tem-se que o erro médio geral obtido com a RNA foi de 2,3°C, o que representa uma diferença de quase 3,9°C para o valor obtido com o RLM. Adicionalmente, o maior erro observado no caso da RNA foi de 19,2°C, resultando em uma diferença de 18,2°C para o RLM.

Esses resultados evidenciam, portanto, a superioridade da metodologia proposta com a rede neural artificial em comparação com a regressão linear multivariada.

4.4 Análise do Tempo de Simulação

Conforme ressaltado na Seção 2.2, o modelo de comportamento térmico do satélite no Simulador Operacional deve fornecer informações em um intervalo de tempo curto o suficiente para que o simulador como um todo seja executado em tempo real. Por essa razão, foram realizados testes de tempo de execução em diferentes computadores para verificar o tempo de processamento necessário para a RNA gerar 5 órbitas para o satélite Amazonia-1. Os resultados desses testes podem ser vistos na Tabela 4.22.

Tabela 4.22. Tempo médio para gerar 5 órbitas.

#	Processador	Tempo médio (s)
1	Intel Core i5 M 460	0,063
2	Intel Core i5 750	0,115
3	Intel Core i5-4590S	0,084
Média geral		0,088

Fonte: Produção do autor.

Os valores mostrados na tabela referem-se ao tempo médio para a RNA gerar 5 órbitas, calculado a partir de diversas rodadas independentes. De acordo com os responsáveis pelo desenvolvimento dos Simuladores Operacionais no INPE, o intervalo de tempo disponível para o modelo térmico gerar a distribuição de temperaturas em um ciclo de atualização do simulador está entre 0,1 a 0,2s.

Dessa forma, considerando que no Simulador Operacional a rede neural só precisa gerar um ponto na órbita a cada ciclo de atualização, uma média geral

de 0,088s, para gerar 5 órbitas, representa um intervalo de tempo suficientemente curto para atender os requisitos temporais de execução do modelo térmico no Simulador Operacional.

Para fins de comparação, o tempo necessário para o SINDA gerar as mesmas 5 órbitas para um cenário específico do satélite Amazonia-1 é de cerca de 10,6h.

Conclui-se, portanto, que a rede neural pode ser utilizada em um Simulador Operacional para simular o comportamento térmico de um satélite em tempo real de forma eficiente.

Uma desvantagem da rede neural proposta, é que como o tempo foi inserido na rede de forma implícita, esta não pode fornecer as temperaturas para qualquer intervalo de tempo, mas apenas no intervalo (ou múltiplos deste) para o qual ela foi treinada. No caso deste projeto, a RNA foi treinada para intervalos de 1 minuto, porém este intervalo pode ser reduzido ou aumentado de acordo com a conveniência, desde que os dados de treinamento conttenham a informação no intervalo desejado.

No Apêndice C é apresentada uma discussão mais aprofundada sobre a integração do Modelo Térmico baseado em Rede Neural com os futuros Simuladores Operacionais do INPE.

5 CONCLUSÕES E TRABALHOS FUTUROS

Ao longo deste projeto de Tese de Doutorado, foram realizadas extensas simulações com o objetivo de verificar a aplicabilidade de Redes Neurais Artificiais do tipo MLP, explorando todos os ajustes necessários, para determinar os valores de temperatura de um Satélite Artificial em órbita da Terra, ao longo do seu tempo de vida em operação, atendendo aos requisitos de um Simulador Operacional de Satélites.

Os estudos preliminares, realizados com aumento gradativo de complexidade do satélite, indicaram que as RNAs do tipo MLP são capazes de reproduzir as temperaturas de veículos espaciais apresentadas durante o treinamento, inclusive para um satélite de porte médio como o Amazonia-1 (ver Seção 4.1).

Contudo, apenas reproduzir os dados apresentados no treinamento da rede neural não é suficiente para atender os requisitos de um Modelo Térmico de um Simulador Operacional de Satélite. Por essa razão, o foco da pesquisa foi direcionado para a capacidade de generalização das redes neurais artificiais, isto é, habilidade destas responderem de forma adequada a dados nunca antes apresentados, desde que estejam dentro do intervalo de valores das variáveis de entrada.

Nesta Tese, foi avaliada a capacidade da RNA de fornecer dados de temperatura em diversos pontos pré-definidos do satélite para quaisquer cenários intermediários entre os casos extremos de máxima (quente) e mínima (frio) temperaturas, não apresentados durante o treinamento.

Além disso, a RNA foi avaliada quanto a sua resiliência, ou seja, independente dos valores de entrada da rede (desde que sejam coerentes com a realidade do satélite), esta deve ser capaz de gerar o comportamento térmico correspondente o mais próximo possível do satélite real em órbita da terra.

Essas características das redes neurais foram verificadas a partir dos resultados obtidos com a arquitetura proposta na Seção 4.2. Nessa arquitetura,

foram introduzidas na entrada da RNA informações específicas sobre as taxas de aquecimento às quais o satélite está sujeito ao longo da sua órbita em torno da Terra e também os valores de temperatura para o instante imediatamente anterior. Além disso, o tempo, que na Seção 4.1 era uma variável de entrada da rede, passou a ser tratado de forma implícita, pela própria variação das taxas de aquecimento ao longo da órbita.

Essas características da arquitetura, aliadas à estratégia de adicionar cenários aleatórios ao treinamento, se mostraram eficazes em potencializar a capacidade de generalização da RNA.

Três algoritmos foram explorados: Gradiente Descendente, Levenberg-Marquardt e Otimização Extrema Generalizada. Para o modelo do CubeSat (Subseções 4.2.1 a 4.2.5) os resultados dos três algoritmos foram eficazes no treinamento da rede neural, apresentando resultados equivalentes e atingindo erros de generalização bem abaixo do limite de 5°C (valor definido como requisito para o Simulador Operacional).

Considerando os resultados para o satélite Amazonia-1 (Subseção 4.2.6), no melhor caso foi obtido um erro médio de generalização de 2,3°C, o que pode ser considerado um excelente resultado se for levado em conta o requisito de 5°C e a complexidade do satélite. Por outro lado, o erro máximo de generalização observado foi de 19,2°C. Apesar de ser expressivo, esse erro foi observado para um dos nós (dentre os 50 considerados), em um ponto específico da órbita e para um cenário não treinado específico. Esse erro mais elevado se deve à presença do ruído presente nas curvas geradas pela RNA, que está relacionado com a complexidade da informação que a rede neural precisa adquirir, considerando o comportamento térmico de 50 nós para diversos cenários diferentes.

Os resultados com uso da arquitetura da RNA proposta foram comparados com os seguintes métodos da literatura (Seção 4.3): (i) Método de Caracterização Térmica de Satélites (MANON et al., 2018); e (ii) Método de Regressão Linear Multivariada.

A comparação dos resultados obtidos com a RNA proposta aplicada ao CubeSat e com o Método de Caracterização Térmica de Satélites, também aplicada a um CubeSat, demonstrou que ambas as soluções são capazes de gerar temperaturas próximas ao satélite real, com um erro máximo de 5°C para o Método de Caracterização Térmica contra um erro máximo de 2,8°C para a RNA. Apesar dos modelos utilizados não serem os mesmos, esse resultado indica que a abordagem proposta nesta Tese é competitiva em comparação com os resultados mais recentes encontrados na literatura (ver Subseção 4.3.1).

Os resultados obtidos com a RNA proposta aplicada a um satélite real de médio porte, o satélite Amazonia-1, foram comparados com o Método de Regressão Linear Multivariada (RLM) aplicado ao mesmo conjunto de cenários utilizado para treinamento da RNA, com o objetivo de construir um modelo matemático. Esse modelo matemático foi, em seguida, utilizado para calcular as temperaturas dos mesmos cenários não aplicados no treinamento da rede neural (ver Seção 4.3.2). Com o método RLM o erro médio observado foi de 6,2°C que, apesar de ser relativamente baixo, está acima do limite de 5°C. Além disso, o maior erro observado foi de 37,4°C, o que é bastante elevado. Por outro lado, com a RNA o erro médio foi de 2,3°C e o máximo de 19,2°C.

Finalmente, foi realizada uma comparação do tempo de processamento necessário para que a rede neural gere as temperaturas para 5 órbitas para um cenário específico em comparação com o tempo necessário para simular essas mesmas 5 órbitas utilizando o software SINDA. No caso da RNA o tempo médio foi de 0,088s, enquanto no SINDA foram necessárias cerca de 10h, mostrando o potencial da solução proposta nesta Tese, utilizando redes neurais artificiais para realizar simulações em tempo real (vide Seção 4.4).

Os resultados obtidos nesta tese indicam que é possível definir uma arquitetura e um conjunto de parâmetros para que redes neurais possam ser usadas como base para a construção do Modelo Térmico de Satélites para Simuladores Operacionais. Esta solução se mostrou viável, tendo apresentado

resultados comparáveis (nos casos estudados) às principais metodologias encontradas na literatura.

Os resultados obtidos nesta Tese representam uma contribuição para o estado da arte da Engenharia de Sistemas Espaciais, mais especificamente para o desenvolvimento do Modelo Térmico de Simuladores Operacionais de Satélites. No Apêndice C consta uma discussão de como a solução proposta poderá ser utilizada nos Simuladores Operacionais do INPE.

Trabalhos Futuros

Como trabalhos futuros, pretende-se treinar a RNA para simular a transição para o estado de emergência do satélite Amazonia-1. O estado de emergência é acionado quando há a possibilidade de dano ou perda de controle do satélite. Nesse estado todos os sistemas não essenciais são desligados e o satélite assume uma atitude diferente da normal mantendo a iluminação nos painéis solares e mantendo o controle térmico.

Visando melhorar a precisão dos resultados, reduzindo principalmente os erros máximos observados, poderão ser exploradas outras estratégias, tais como: (i) aumentar ainda mais o número de cenários apresentados no treinamento; (ii) reduzir o intervalo de tempo de 1 min para 1s, aumentando, conseqüentemente, o número de padrões de treinamento para cada cenário; (iii) utilizar uma RNA com mais de uma entrada atrasada, isto é, utilizando dois ou três instantes anteriores; e (iv) utilizar outras metodologias para o treinamento, como a metodologia das Derivadas Instantâneas (WANG e LIN, 1998) e/ou a metodologia das Derivadas Médias (TASINAFFO; RIOS NETO, 2003).

Outro aspecto importante a ressaltar é que a metodologia utilizada nesta Tese pode ser aplicada também para outros problemas que envolvam a simulação em tempo real de sistemas físicos e, de forma mais geral, para casos nos quais haja interesse em simular um sistema físico a partir de um conjunto de dados que seja representativo do seu comportamento dinâmico,

especialmente quando não houver um modelo matemático que descreva o sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- AIRBUS DEFENSE AND SPACE. **Thermica**. Disponível em: <http://www.systema.airbusdefenceandspace.com/products/thermica.html>. Acesso em: 18 out. 2018.
- AMBROSIO, A. M.; CARDOSO, P. E.; ORLANDO, V.; BIANCHI-NETO, J. Brazilian satellite simulators: previous solutions trade-off and new perspectives for the CBERS program. In: CONFERENCE ON SPACE OPERATIONS (SPACEOPS), 8., 2006, Rome, Italy. **Proceedings...** Rome: AIAA, 2006.
- AZEVEDO, D. R.; HOFFMANN, L. T.; AMBROSIO, A. M.; PERONDI, L. F. Analysis of the simulation model platform adoption in the context of INPE simulators. In: WORKSHOP ON SIMULATION FOR EUROPEAN SPACE PROGRAMMES (SESP), 2012, Noordwijk, Netherlands. **Proceedings...** Noordwijk: ESTEC/ESA, 2012.
- BAK, P.; SNEPPEN, K. Punctuated equilibrium and criticality in a simple model of evolution. **Physical Review Letters**, p. 4083-4086, 1993.
- BOETTCHER, S.; PERCUS, A. Optimization with extremal dynamics. **Physical Review Letters**, v.86, p. 353-373, 2001.
- C&R TECHNOLOGIES. **Thermal desktop**: CAD-based thermal engineering tool suite, Version 5.6, 2013. Disponível em: <https://crtech.com/>.
- CARRARA, V. Redes neurais aplicadas ao controle de atitude de satélites com geometria variável. 1997. 195p. Tese (Doutorado em Engenharia e Tecnologia Espaciais) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1997.
- CHEN, S.; BILLINGS, S.A.; GRANT, P.M. Non-linear system identification using neural networks. **International Journal of Control**, v.51, n.6, p.1191-1214, 1990.
- CHOJACZYK, A.A.; TEIXEIRA, A.P.; NEVES, L.C.; CARDOSO, J.B.; GUEDES SOARES, C. Review and application of artificial neural networks models in reliability analysis of steel structures. **Structural Safety**, v.52, p. 78-89, 2015.
- COSTA, L. L. **Projeto do subsistema de controle térmico para Cubesats**. Trabalho (Conclusão de Curso) – Universidade Federal de Santa Maria, Santa Maria, 2010.
- COTTER, N.E. The Stone-Weierstrass and its application to neural networks. **IEEE Transactions on Neural Networks**, v.1, n.4, p.290-295, 1990.

CYBENKO, G. Continuous valued networks with two hidden layers are sufficient. **Technical Report**, Department of Computer Science, Tufts University, 1988.

CYBENKO, G. Approximation by superposition of a sigmoidal function. **Mathematics Control Signals Systems**, v. 2, p. 303-314, 1989.

DE SOUSA, F.L.; RAMOS, F.M. Function optimization using extremal dynamics, In: INTERNATIONAL CONFERENCE IN INVERSE PROBLEMS IN ENGINEERING, 4., 2002, Angra dos Reis, Brasil. **Proceedings...** Angra dos Reis, 2002.

DE SOUSA, F. L. **Otimização extrema generalizada: um novo algoritmo estocástico para o projeto ótimo**. 2002. 142p. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2003.

EICKHOFF, J. **Simulating spacecraft systems**. Berlin: Springer-Verlag, 2009.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION – ECSS. **ECSS-E-ST-10-04C**: space engineering: space environment. Noordwijk, The Netherlands, 2008.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION - ECSS. **ECSS-E-TM-10-21A**: system modeling and simulation. Noordwijk, The Netherlands, 2010.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION – ECSS. **ECSS-ETM-40-07**: space engineering: simulation modeling platform. Noordwijk, The Netherlands, 2011.

EUROPEAN COOPERATION FOR SPACE STANDARDIZATION - ECSS. **ECSS-S-ST-00-01C**: ECSS system: glossary of terms. Noordwijk, The Netherlands, out. 2012. 63p.

FANG, K.; DAOBIN, M.; SHI, C.; BORONG, W.; FENG, W. A prediction model based on artificial neural network for surface temperature simulation of nickel–metal hydride battery during charging. **Journal of Power Sources**, v.208, p. 378–382, 2012.

FORTESCUE, P.; STARK, J.; SWINERD, G. **Spacecraft systems engineering**. 3.ed. [S.l.]: Wiley-Blackwell, 2003. 704 p.

FREITAG, S.; CAO, B.T.; NINIC, J.; MESCHKE, G. Recurrent neural networks and proper orthogonal decomposition with interval data for real-time predictions of mechanised tunneling processes. **Computers and Structures**, v.207, p. 258-273, 2018.

GILMORE, D. G. **Spacecraft thermal control handbook**. 2.ed. El Segundo, United States of America: The Aerospace Press, 2002. v.1, 836 p. ISBN 1-884989-11-X.

HAGAN, M.T.; MENHAJ, M. Training feedforward networks with the Marquardt algorithm. **IEEE Transactions on Neural Networks**, v. 5, n. 6, 1994.

HAGAN, M.T.; DEMUTH, H.B.; BEALE, M.H.; DE JESÚS, O. **Neural network design**. 2.ed. [S.l.: s.n.], 2014.

HAYKIN, S. **Redes neurais: princípios e prática**. 2.ed. [S.l.]: Bookman, 2001.

HENDRICKS, R.; EICKHOFF, J. The significant role of simulation in satellite development and verification. **Aerospace Science and Technology**, v. 9, n.3, p. 273-283, 2005.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, p. 359-366, 1989.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS – INPE. **CBERS 3&4 satellite simulator system requirements specification: RT-SRS-1021/01**. São José dos Campos, Brasil: INPE, 2016a.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS – INPE. **Satellite model requirement specification for CBERS-4 satellite simulator: RT-SRS-1025/00**. São José dos Campos, Brasil: INPE, 2016b.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS – INPE. **Satélite Amazonia-1 e Plataforma Multimissão (PMM)**. Disponível em: <http://www3.inpe.br/amazonia-1>. Acesso em: 02 jul. 2018.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS - INPE. **Vídeo INPE – CBERS 2011**. Disponível em: <http://www.inpe.br/noticias/videos.php>. Acesso em: 02 jul. 2018.

IRVINE, M.; FRITZEN, P.; ELLSIEPEN, P. Trends in European Space Simulation: standards, architectures and tools across the Mission Lifecycle. In: INTERNATIONAL CONFERENCE ON RECENT ADVANCES IN SPACE TECHNOLOGIES (RAST), 6., 2013, **Proceedings...** 2013, p. 1163-1167.

KANG, J.Y.; KIM, J.M.; CHUNG, S.J. Design and development of an advanced real-time satellite simulator. **ETRI Journal**, v.17, n.3, 1995.

LINEAR ALGEBRA PACKAGE C INTERFACE – LAPACKE. Disponível em: <http://www.netlib.org/lapack/lapacke.html>. Acesso em: 10 nov. 2018.

LEVA, J.L. A fast normal number generator. **ACM Transactions on Mathematical Software**, v.18, n.4, p.449-453, 1992.

LEVENBERG, K. A Method for the solution of certain non-linear problems in least squares. **The Quarterly of Applied Mathematics**, v. 2, p.164-168, 1944.

LO, Z.-P.; YU, Y.; BAVARIAN, B. Analysis of the convergence properties of topology preserving neural networks. **IEEE Transactions on Neural Networks**, v.4, p.207-220, 1993.

LOPES, I.M.; DE SOUSA, F.L.; DESOUSA, L.C.G. The generalized extremal optimization with real codification, In: INTERNATIONAL CONFERENCE ON ENGINEERING OPTIMIZATION (EngOpt), 2008. **Proceedings...** 2008.

MAIA, A.G. Regressão linear múltipla. Disponível em: http://www4.eco.unicamp.br/docentes/gori/images/arquivos/Econometria/Econometria_RegressaoMultipla.pdf. Acesso em: 10 nov. 2018.

MANON, F.; BAROUKH, J.; MAS, G.; PASQUIER, H.M.; FRANÇOIS, T.; DESCHAMPS, S.; CORTES, H. A novel concept for thermal simulation in spacecraft simulators. In: CONFERENCE ON SPACE OPERATIONS (SPACEOPS), 2018. **Proceedings...** AIAA, 2018.

MARQUARDT, D.W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the Society for Industrial and Applied Mathematics**, v.11, n.2, p.431-441, 1963.

MARTÍNEZ-HERAS, J.A; DONATI, A. Artificial neural networks in support of spacecraft thermal behaviour modeling. In: IEEE AEROSPACE CONFERENCE, 2004. **Proceedings...** 2004.

MCCULLOC, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v.5, p. 115–133, 1943.

MINSKY, M.L.; PAPERT, S.A. **Perceptrons**. Cambridge, MA: MIT Press, 1969.

MOHANRAJ, M.; JAYARAJ, S.; MURALEEDHARAN, C. Applications of artificial neural networks for thermal analysis of heat exchangers: a review. **International Journal of Thermal Sciences**, v.90, p. 150-172, 2015.

NOGUEIRA, F.E. **Modelos de regressão multivariada**. 2007. 97p. Dissertação (Mestrado em Estatística) - Universidade de São Paulo, São Paulo, 2007.

OJHA, V.K.; AJITH, A.; VCLAV, S. Metaheuristic design of feedforward neural networks. **Engineering Applications of Artificial Intelligence**, v.60 n.C, p.97-116, 2017.

PERPIÑÁN, M.A.C. The Modelling of the thermal subsystem in spacecraft real time simulators. In: WORKSHOP ON SIMULATORS FOR EUROPEAN SPACE

- PROGRAMMES (SESP), 3., 1994, Noordwijk, The Netherlands. **Proceedings...** Noordwijk:ESA/ESTEC, 1994. p. 69-78.
- PIDGEON, A.; BÉGIN, M.E. System simulation: risk reduction for the new millennium. In: IEEE AEROSPACE CONFERENCE, 2., 2000. **Proceedings...** 2000, p. 415-425.
- PIOTROWSKI, A.P.; OSUCH, M.; NAPIORKOWSKI, M.J.; ROWINSKI, P.M.; NAPIORKOWSKI, J.J. Comparing large number of metaheuristics for artificial neural networks training to predict water temperature in a natural river. **Computers & Geosciences**, v.64, p. 136-151, 2014.
- RAIF, M.; WALTER, U.; BOUWMEESTER, J. Dynamic system simulation of small satellite projects. In: INTERNATIONAL ASTRONAUTICAL CONGRESS, 2009. **Proceedings...** 2009.
- REGGESTAD, V.; PECCHIOLI, M.; MERRI, M. Virtual reality for real operations: developing and using operational simulators. **European Space Agency Bulletin**, n. 148, p. 42-51, 2011.
- REGGESTAD, V. Increasing performance of ESA operational spacecraft simulators . In: WORKSHOP ON SIMULATION FOR EUROPEAN SPACE PROGRAMMES (SESP), 2012, Noordwijk, Netherlands. **Proceedings...** Noordwijk, 2012.
- REIS JUNIOR, J.D.; AMBROSIO, A.M.; DE SOUSA, F.L. Towards spacecraft real-time thermal simulation with artificial neural networks. In: ABCM INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING (COBEM), 23., 2015. **Proceedings...** 2015.
- REIS JUNIOR, J.D.; AMBROSIO, A.M.; DE SOUSA, F.L. Reproducing Amazonia-1 Satellite thermal behavior with artificial neural networks. In: WORKSHOP EM ENGENHARIA E TECNOLOGIA ESPACIAIS (WETE), 7., 2016, São José dos Campos, Brazil. **Proceedings...** São José dos Campos: INPE, 2016a.
- REIS JUNIOR, J.D.; AMBROSIO, A.M.; DE SOUSA, F.L. Real-time Cubesat thermal simulation using artificial neural networks. In: CONFERENCE OF COMPUTATIONAL INTERDISCIPLINARY SCIENCE (CCIS), 4., 2016, São José dos Campos, Brazil. **Proceedings...** São José dos Campos, 2016b.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v.65, n. 6, p. 386–408, 1958.
- ROSENBLATT, F. **Principles of neurodynamics**. Washington, DC: Spartan Books, 1962.

RUMELHART, D.E.; HINTON, G.E.; WILLIAMS, R.J. Learning representations by back-propagating errors. **Nature**, v.323, p. 533–536, 1986.

SILVA, D.F.; MURAOKA, I.; GARCIA, E.C. Thermal control design conception of the Amazonia-1 satellite. **Journal of Aerospace Technology Management**, v.6, n.2, p.169-176, 2014.

SILVA NETO, A.J.; BECCENERI, J.C.; VELHO, H.F.C. **Inteligência computacional aplicada a problemas inversos em transferência radiativa**, Rio de Janeiro: Ed. UERJ, 2016.

SOGAB NÚCLEO DE REABILITAÇÃO TRAUMATO ORTOPÉDICA. **Neurônio**. Disponível em: <http://www.sogab.com.br/anatomia/neuronio.jpg>. Acesso em: 26 out. 2015.

TASINAFFO, P.M.; RIOS NETO, A. Mean derivatives based neural euler integrator for nonlinear dynamic systems modeling. **Learning and Nonlinear Models**, v.3, n.2, p.98-109, 2005.

TOMINAGA, J.; CERQUEIRA, C.; KONO, J.; AMBROSIO, A. M. Specifying satellite behavior for an operational simulator. In: WORKSHOP ON SIMULATION FOR EUROPEAN SPACE PROGRAMMES (SESP), 2012, Noordwijk, The Netherlands. **Proceedings...** Noordwijk, 2012.

WANG, Y.-J.; LIN, C.-T. Runge-kutta neural network for identification of dynamical systems in high accuracy. **IEEE Transactions on Neural Networks**, v.9, n.2, p.294-307, 1998.

WOELLERT, K.; EHRENFREUND, P.; RICCO, A.J.; HERTZFELD, H. Cubesats: cost-effective science and technology platforms for emerging and developing nations. **Advances in Space Research**, v. 47, n.4, p.663-684, 2011.

YADAV, A.K.; CHANDEL, S.S. Solar radiation prediction using artificial neural network techniques: a review. **Renewable and Sustainable Energy Reviews**, v.33, p. 772-781, 2014.

YANG, K. Artificial Neural Networks (ANNs): a new paradigm for thermal science and engineering. **Journal of Heat Transfer**, v.130, n.9, 2008.

YONDO, R.; ANDRÉS, E.; VALERO, E. A review on design of experiments and surrogate models in aircraft real-time and many query aerodynamic analyses. **Progress in Aerospace Sciences**, v.96, p. 23-61, 2018.

ZEMERICK, S.A.; MORRIS, J.R.; BAILEY, B.T. NASA operational simulator (NOS) for V&V of complex systems. Proceedings of SPIE, v.8752, 2013. Disponível em: <https://www.spiedigitallibrary.org/conference-proceedings-of->

spie/8752/875205/NASA-Operational-Simulator-NOS-for-VampV-of-complex-systems/10.1117/12.2015246.full?SSO=1.

APÊNDICE A – DADOS DE TREINAMENTO PARA O CUBESAT

Neste apêndice é mostrado um exemplo do conjunto de treinamento (cenário Hot) utilizado para o modelo do CubeSat.

Tempo	Hr1	...	Hr6	Eq1	Eq2	Eq3	T1	...	T9
0.000000	0.000000	...	134.489752	0.800000	0.400000	0.600000	18.299652	...	8.036587
60.000000	39.252747	...	136.780421	0.800000	0.400000	0.600000	18.419012	...	8.157823
120.000000	81.595494	...	140.504336	0.800000	0.400000	0.600000	18.535354	...	8.300945
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
5880.000000	0.000000	...	134.550526	0.800000	0.400000	0.600000	18.035639	...	7.757589
5940.000000	0.000000	...	133.637384	0.800000	0.400000	0.600000	18.170273	...	7.907127
6000.000000	0.000000	...	134.489752	0.800000	0.400000	0.600000	18.298991	...	8.035725

APÊNDICE B – CENÁRIOS SIMULADOS PARA O AMAZONIA-1

Neste Apêndice, são mostrados os parâmetros que caracterizam os diferentes cenários simulados para o satélite Amazonia-1, cujos resultados constam da Subseção 4.2.6.

	hot	cold	A	B	C	D	E	F	G	H	I	J
Albedo	0.420	0.340	0.383	0.377	0.397	0.362	0.355	0.406	0.346	0.390	0.374	0.341
Solar [W/m ²]	1418.000	1326.000	1357.000	1367.000	1355.000	1379.000	1407.000	1365.000	1380.000	1375.000	1344.000	1338.000
Terrestre [W/m ²]	233.000	208.000	209.200	221.000	226.000	221.900	209.200	215.600	221.800	221.900	223.400	231.900
ACE	22.000	12.000	13.800	14.400	21.700	16.100	13.900	18.100	16.300	18.400	19.100	16.200
Battery 1	2.790	2.300	2.610	2.800	2.540	2.570	2.330	2.590	2.500	2.760	2.540	2.750
Battery 2	2.790	2.300	2.590	2.570	2.750	2.610	2.540	2.800	2.540	2.500	2.760	2.330
Battery 3	2.790	2.300	2.750	2.590	2.610	2.540	2.570	2.500	2.760	2.330	2.540	2.800
Battery 4	2.790	2.300	2.570	2.540	2.330	2.590	2.760	2.610	2.750	2.800	2.500	2.540
GPS Receiver	11.700	9.300	11.440	10.790	9.500	11.350	11.460	10.580	10.890	10.240	9.320	11.190
GYRO EM	30.000	21.600	26.200	22.300	29.800	24.700	27.500	29.700	28.000	29.600	23.500	25.900
GYRO ICU	0.150	0.080	0.086	0.080	0.119	0.105	0.146	0.132	0.081	0.148	0.082	0.094
LNA	0.650	0.650	0.650	0.650	0.650	0.650	0.650	0.650	0.650	0.650	0.650	0.650
Magnetotorquer 1	2.700	0.000	2.080	1.860	0.060	0.800	0.040	0.770	0.670	0.590	1.750	0.780
Magnetotorquer 2	2.700	0.000	0.060	0.670	0.770	0.780	2.080	0.590	0.800	1.750	0.040	1.860
Magnetotorquer 3	2.700	0.000	0.670	2.080	0.800	0.040	0.590	0.060	1.750	0.770	0.780	1.860
OBDR	32.500	32.500	32.500	32.500	32.500	32.500	32.500	32.500	32.500	32.500	32.500	32.500
PDCU	*	6.300	19.600	56.000	53.300	55.800	22.900	45.800	33.600	21.800	35.100	26.000
Reaction Wheel 1	**	6.700	119.700	69.600	48.300	150.400	132.500	101.800	155.400	55.500	109.200	141.800
Reaction Wheel 2	**	6.700	55.500	150.400	101.800	69.600	141.800	48.300	109.200	132.500	119.700	155.400
Reaction Wheel 3	**	6.700	101.800	141.800	132.500	109.200	119.700	155.400	55.500	69.600	150.400	48.300
Reaction Wheel 4	**	6.700	132.500	119.700	155.400	141.800	55.500	69.600	48.300	150.400	101.800	109.200
SADA 1	6.500	5.100	5.810	5.730	6.240	6.270	6.000	5.960	5.880	6.500	5.960	5.210
SADA 2	6.500	5.100	6.240	6.500	5.810	5.960	5.730	5.880	5.210	5.960	6.270	6.000
SADE	17.500	10.950	16.870	15.440	13.910	13.170	12.000	15.500	15.370	13.610	12.450	15.960
Star Sensor 1	13.500	0.225	6.790	7.950	11.220	7.090	9.600	0.290	7.440	9.190	6.020	0.620
Star Sensor 2	13.500	0.225	11.220	9.600	6.020	0.290	9.190	0.620	6.790	7.440	7.950	7.090
TCE	0.800	0.000	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800	0.800
TT&C 1	6.500	6.500	6.500	12.400	6.500	19.300	6.500	6.500	6.500	10.600	6.500	6.500
TT&C 2	***	6.500	19.100	0.000	8.700	0.000	6.800	20.900	9.000	0.000	8.800	18.900
BPF	***	0.000	1.300	1.800	3.000	6.600	6.800	7.900	6.100	2.700	7.900	9.500
DC/DC	10.400	6.420	8.050	6.490	8.110	10.090	8.790	6.800	8.980	6.610	10.280	8.880
EPC 1	***	0.100	8.100	0.000	8.200	3.000	0.000	5.200	3.400	2.000	1.500	0.000
EPC 2	0.100	0.100	0.100	4.900	0.100	0.100	6.900	0.100	0.100	0.100	0.100	2.700
HPS	***	0.000	0.980	0.300	0.900	0.730	0.220	0.890	0.140	0.170	0.760	0.400
QPSK-TX	***	0.000	11.300	7.000	4.100	9.800	1.300	11.400	7.100	3.200	1.800	7.800
RTU 1	5.700	5.300	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500
RTU 2	5.700	5.300	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500	5.500
SDC	***	0.000	4.500	2.400	2.800	0.400	0.900	0.100	4.800	2.900	2.800	4.800
SPE	****	6.700	48.100	30.000	34.700	27.700	38.300	12.500	8.200	55.000	51.900	36.500
SSR	10.800	10.080	10.800	10.800	10.800	10.800	10.800	10.800	10.800	10.800	10.800	10.800
TWT 1	***	0.000	0.000	18.700	0.000	34.400	2.700	7.200	32.000	37.200	0.000	33.600
TWT 2	0.000	0.000	16.100	0.000	23.900	0.000	0.000	0.000	0.000	0.000	41.100	0.000

* inicia em 57.0W e decai para 15.8W em t=3980s.

** inicia em 161.6W; decai para 26.8W em t=180s; retorna para 161.6W em t=720s; e decai novamente para 26.8W em 900s.

*** iniciam no seu valor nominal (vide abaixo) e decaem para *standby* em t=1200s.

Valores nominais: TT&C2=21.0W; BPF=10.3W; EPC1=9.0W; HPS=1.0W; QPSK-TX=12.0W; SDC=5.3W; TWT1=45.0W.

Valores de *standby*: TT&C2=6.5W; BPF=EPC1=HPS=QPSK-TX=SDC=TWT1=0.0W.

APÊNDICE C – INTEGRAÇÃO NO SIMULADOR DO INPE

Neste Apêndice é discutido como a solução proposta nesta Tese, poderá ser integrada como um Modelo Térmico nos futuros Simuladores Operacionais desenvolvidos no INPE.

De acordo com o documento de Especificação de Requisitos para o Modelo de Satélite do Simulador Operacional dos Satélites CBERS-3&4 (INPE, 2016b), este simulador deve representar o comportamento do satélite por meio da simulação dos seus subsistemas. Os subsistemas representados são: Suprimento de Energia, Supervisão de Bordo, Telemetria e Telecomando, Controle de Órbita e Atitude e Controle Térmico, além dos subsistemas da carga útil.

O Modelo Térmico utilizado no Simulador dos Satélites CBERS-3&4 consiste em um modelo simples baseado em regras. De acordo com o documento de requisitos esse modelo deve simular a variação da temperatura nos termistores (sensores de temperatura) dentro do intervalo de valores nominais. Os valores são incrementados ou decrementados de acordo com o estado de operação de cada equipamento e se o satélite está sendo iluminado ou se está em região de sombra (eclipse). Em outras palavras, a operação desse modelo obedece às seguintes regras:

- I. Os valores de temperatura são mantidos em um intervalo de valores entre um máximo e um mínimo pré-definidos;
- II. O valor a ser acrescido ou decrescido da temperatura é iniciado em um valor máximo e é reduzido conforme o termistor se aproxima dos seus limites;
- III. Os valores dos termistores são incrementados quando:
 - a. O satélite é iluminado; e
 - b. O equipamento está ligado ou carregando (no caso das baterias).
- IV. Os valores dos termistores são decrescidos quando:
 - a. O satélite está em eclipse; e

- b. O equipamento está desligado ou em *standby*, ou descarregando (no caso das baterias).

As entradas desse modelo são o estado de iluminação do satélite, os parâmetros do banco de dados representando os termistores e o estado de operação dos equipamentos simulados.

No caso do Modelo Térmico baseado em Rede Neural Artificial, conforme proposto nesta Tese, as entradas são dadas pelas taxas de aquecimento nas superfícies do satélite e pelo estado de operação dos equipamentos.

A informação sobre o estado de operação dos equipamentos já está disponível nos modelos de cada subsistema, podendo ser obtida a partir destes.

Por outro lado, as taxas de aquecimento nas superfícies, devem ser calculadas previamente em um software de análise térmica, como o SINDA (C&R TECHNOLOGIES, 2013). Como essas taxas também são necessárias para o treinamento da RNA, não há necessidade de nenhum esforço adicional para obtê-las.

No caso de satélites de órbita Sol-síncrona, como é o caso dos satélites da série CBERS e Amazônia desenvolvidos no INPE, o ângulo entre o plano da órbita e um vetor com origem no centro do Sol apontando para o centro da Terra é sempre fixo. Isso facilita o cálculo, pois para uma determinada distância fixa da Terra ao Sol as taxas de aquecimento são sempre as mesmas ao longo da órbita do satélite. Neste caso, a única variável a ser considerada é a variação da distância da Terra ao Sol, ao longo da sua órbita. Dessa forma, as taxas de aquecimento podem ser parametrizadas em função da distância da Terra ao Sol.

Outra questão importante é a sincronização do modelo com o ciclo de atualização do simulador. O tempo de execução do simulador e também dos diferentes modelos é gerenciada por um componente denominado *Scheduler* (agendador). O *Scheduler* agenda a execução dos modelos no simulador em uma ordem pré-definida. Os modelos podem ser agendados de forma cíclica

(em múltiplos de uma frequência base) ou em função de eventos assíncronos (p.ex. a chegada de um telecomando).

Atualmente, a frequência base do Simulador Operacional do INPE é de 1s, isto é, a cada 1s é realizado um ciclo de atualização do simulador. Os modelos são executados em múltiplos dessa frequência base. Em outras palavras, os modelos cuja execução é mais rápida são processados a cada ciclo de 1s, enquanto os modelos com execução mais demorada são processados a cada 2s, 3s, ou mais, conforme seja necessário para o modelo concluir sua execução e fornecer as suas saídas.

Conforme foi analisado na Subseção 4.4, com a rede neural é possível gerar 5 órbitas divididas em 100 minutos em um tempo médio de processamento de 0,088s (em processadores utilizados em computadores *desktop* e *laptop* comuns). Ou seja, o modelo proposto é capaz de gerar a distribuição de temperaturas no satélite a cada ciclo de atualização de 1s.

Nas simulações realizadas nesta Tese a RNA foi treinada com dados no intervalo de 1min, porém esse intervalo pode ser reduzido para atender os requisitos do Simulador. Em última instância, caberá aos especialistas em controle térmico definir qual o intervalo de atualização ideal para o Modelo Térmico.

APÊNDICE D – LISTA DE ARTIGOS PRODUZIDOS

Neste Apêndice são listados os artigos produzidos ao longo do projeto de Doutorado. Os artigos são mostrados de acordo com a data em que foram elaborados:

- I. Título: “Towards Spacecraft Real-Time Thermal Simulation with Artificial Neural Networks”, apresentado no *23rd ABCM International Congress of Mechanical Engineering* – COBEM 2015;
- II. Título: “Reproducing Amazonia-1 Satellite Thermal Behavior with Artificial Neural Networks”, apresentado no *7^o Workshop em Engenharia e Tecnologia Espaciais do INPE – WETE* 2016;
- III. Título: “Real-Time Cubesat Thermal Simulation using Artificial Neural Networks”, apresentado na *4th Conference of Computational Interdisciplinary Science* – CCIS 2016.