



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RENATO ANDRADE FREITAS

Aprimoramento da Modelagem de Módulos Fotovoltaicos em ambiente MATLAB/Simulink

Recife

2024

RENATO ANDRADE FREITAS

Aprimoramento da Modelagem de Módulos Fotovoltaicos em ambiente MATLAB/Simulink

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de mestre em Engenharia Elétrica.

Área de concentração: Processamento de Energia.

Orientador: Prof. Dr. Marcelo Cabral Cavalcanti

Coorientador: Prof. Dr. Fabrício Bradaschia

Recife

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Freitas, Renato Andrade.

Aprimoramento da modelagem de módulos fotovoltaicos em ambiente MATLAB/Simulink / Renato Andrade Freitas. - Recife, 2024.

149f.: il.

Dissertação (Mestrado) - Centro de Tecnologia e Geociências, Programa de Pós-Graduação em Engenharia Elétrica, 2024.

Orientação: Marcelo Cabral Cavalcanti.

Coorientação: Fabrício Bradaschia.

1. Estimação; 2. Modelo global não-linear; 3. Módulo fotovoltaico; 4. Sistema fotovoltaico; 5. Matlab. I. Cavalcanti, Marcelo Cabral. II. Bradaschia, Fabrício. III. Título.

UFPE-Biblioteca Central

RENATO ANDRADE FREITAS

**“APRIMORAMENTO DA MODELAGEM DE MÓDULOS
FOTOVOLTAICOS EM AMBIENTE MATLAB/SIMULINK”**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica, na área de concentração em Processamento de Energia.

Aprovada em: 08/08/2024.

BANCA EXAMINADORA

Profº. Fabricio Bradaschia
(Coorientador e Examinador Interno)
Universidade Federal de Pernambuco

Profº. Dr. Gustavo Medeiros de Souza Azevedo
(Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Leandro Michels
(Examinador Externo)
Universidade Federal de Santa Maria

Dedico esta conquista ao meu filho, João Álvaro.

AGRADECIMENTOS

A minha esposa Fernanda Madeira e meu filho João Álvaro, pelo amor, compreensão e apoio incondicional ao longo desta jornada. Vocês foram minha inspiração para nunca desistir.

A minha família, pai, mãe e irmão Vitor, pelo apoio incondicional, incentivo e compreensão em todos os momentos da minha vida. Vocês são minha base, minha referência e minha força.

Ao meu orientador Marcelo Cabral e coorientador Fabricio Bradaschia, pela orientação precisa, incentivo constante e pela confiança depositada em mim. Sem a ajuda de vocês, este trabalho não teria sido possível.

Aos professores Rafael Cavalcanti, Gustavo Medeiros, Leonardo Limongi, Francisco Neves e Alexander, pelos ensinamentos, orientações e pela contribuição para o meu desenvolvimento acadêmico e profissional.

Aos colegas de GEPAE, Antônio Dutra, Camila Bandeira, Carolina Caldeira, Eduardo Augusto, Eduardo José, Ericles Barbosa, Gustavo Teixeira, Néstor Medina, Ricardo Moura, pela troca de experiências, pelas histórias do boi mamão, pelas conversas minimalistas, colaboração e amizade ao longo deste período. Vocês tornaram essa jornada mais leve e enriquecedora.

Em especial, aos colegas de pesquisa mais diretas, Valdemar Cavalcante, Tiago Fernandes, Diego Hamilton, Nayara Melo, Rafael Farias e Beatriz Almeida pela colaboração, apoio e por compartilharem seus conhecimentos, contribuindo para o crescimento mútuo.

À Polícia Federal, pela colaboração e disponibilidade em auxiliar na realização deste estudo e contribuindo para a qualidade deste trabalho.

“A ignorância gera mais frequentemente confiança do que o conhecimento: são os que sabem pouco, e não aqueles que sabem muito, que afirmam de uma forma tão categórica que este ou aquele problema nunca será resolvido pela ciência”
(Charles Darwin)

RESUMO

Este trabalho investiga as limitações e desafios na modelagem de sistemas fotovoltaicos utilizando o bloco *PV Array* do MATLAB/Simulink, que é uma ferramenta nativa de simulação muito utilizada em diversas aplicações. O bloco *PV Array*, embora amplamente utilizado, apresenta limitações significativas, como a necessidade de parâmetros precisos e dificuldades na representação dinâmica das condições operacionais dos módulos fotovoltaicos, o que compromete a precisão das simulações. Este estudo analisa a variabilidade da irradiância e da temperatura ambiente, propondo metodologias para extração e tratamento das curvas I-V. Além disso, são avaliadas alternativas ao uso do bloco *PV Array*, como os Modelos Globais Não Lineares (MGNL), que ajustam seus parâmetros de acordo com as variações ambientais. Experimentos foram conduzidos utilizando uma bancada experimental para a extração de curvas I-V, com tratamento e análise dos dados obtidos para validar a metodologia proposta. A comparação entre os resultados da ferramenta nativa *PV Array* e da ferramenta desenvolvida baseada no MGNL mostrou que o MGNL oferece maior precisão na representação dos sistemas fotovoltaicos sob diferentes condições operacionais. Um problema do bloco *PV Array* é a confiabilidade dos parâmetros utilizados, assim, os resultados indicam que metodologias alternativas como o uso do MGNL para obter os parâmetros iniciais do bloco podem melhorar significativamente a precisão do modelo. Dessa forma, diversas aplicações podem ser impactadas pelo uso de ferramentas mais precisas de modelagem de módulos FV, tais como previsão de geração de energia FV, técnicas de detecção de degradação em potência, projeto otimizado de conversores e avaliação de técnicas de rastreamento de ponto de máxima potência (MPP).

Palavras-chave: Estimação; Modelo global não-linear; Módulo fotovoltaico; Sistema fotovoltaico; Matlab.

ABSTRACT

This work investigates the limitations and challenges in modeling photovoltaic systems using the *PV Array* block from MATLAB/Simulink, which is a widely used native simulation tool in various applications. The *PV Array* block, although extensively utilized, presents significant limitations, such as the need for precise parameters and difficulties in dynamically representing the operating conditions of photovoltaic modules, which compromises the accuracy of simulations. This study analyzes the variability of irradiance and ambient temperature, proposing methodologies for the extraction and treatment of I-V curves. Additionally, alternatives to the *PV Array* block are evaluated, such as Global Nonlinear Models (GNLM), which adjust their parameters according to environmental variations. Experiments were conducted using an experimental setup for the extraction of I-V curves, with processing and analysis of the obtained data to validate the proposed methodology. The comparison between the results of the native *PV Array* tool and the developed tool based on the GNLM showed that the GNLM provides greater accuracy in representing photovoltaic systems under different operating conditions. One issue with the *PV Array* block is the reliability of the parameters used, thus the results indicate that alternative methodologies, such as the use of the GNLM to obtain the block's initial parameters, can significantly improve the model's accuracy. In this way, various applications can be impacted by the use of more accurate tools for modeling PV modules, such as PV energy generation forecasting, fault detection techniques, optimized converter design, and the evaluation of maximum power point tracking (MPPT) techniques.

Keywords: Global nonlinear model; Photovoltaic module; Photovoltaic system; Matlab.

LISTA DE FIGURAS

Figura 1 – Fluxograma da metodologia de detecção de faltas do MBDM.	24
Figura 2 – Processo de conversão de energia em uma célula FV.	28
Figura 3 – Representação do conjunto de células formando um módulo e do conjunto de módulos formando um arranjo.	29
Figura 4 – Configuração de arranjos FV: adição de módulos em série e paralelo.	29
Figura 5 – Curvas I-V apresentadas no <i>datasheet</i> do Módulo Canadian Solar Inc. CS3U-365P.	30
Figura 6 – Exemplo de curva I-V e curva P-V destacando pontos relevantes.	31
Figura 7 – Comportamento da curva I-V e P-V de um módulo FV com a mudança de irradiância incidente no módulo usando o modelo <i>PV Array</i> do MATLAB/Simulink.	32
Figura 8 – Comportamento da curva I-V e P-V de um módulo FV com a mudança da temperatura do módulo usando o modelo <i>PV Array</i> do MATLAB/Simulink.	32
Figura 9 – Representação do circuito elétrico do SDM.	34
Figura 10 – Comportamento da curva I-V e P-V de um módulo FV com a mudança do n	35
Figura 11 – Comportamento da curva I-V e P-V de um módulo FV com a mudança de R_s	36
Figura 12 – Comportamento da corrente e da potência de um módulo FV com a mudança de R_{sh}	37
Figura 13 – Representação do circuito elétrico do DDM.	38
Figura 14 – Classificação de faltas em sistemas FV de acordo com (PILLAI et al., 2018).	40
Figura 15 – Bloco <i>PV Array</i>	47
Figura 16 – Parâmetros Bloco <i>PV Array</i> no MATLAB/Simulink R2023a.	48
Figura 17 – Parâmetros internos do Bloco <i>PV Array</i> no MATLAB/Simulink R2023a.	50
Figura 18 – Parâmetros internos do Bloco <i>PV Array</i> no MATLAB/Simulink R2023a.	51
Figura 19 – Plano de R_s para módulos de silício.	52
Figura 20 – Comportamento da energia de banda proibida de acordo com o Efeito Varshni para materiais semicondutores Si, GaAs, CdTe e CdS em função da variação de temperatura.	54
Figura 21 – Energia de banda proibida comparando Efeito Varshni em (29) e a linearização apresentada por (de SOTO et al., 2006) em (28) para materiais semicondutores (a) Si, (b) GaAs, (c) CdTe e (d) CdS em função da variação de temperatura.	55
Figura 22 – Datasheet do módulo First Solar - FS-4122A-3.	56
Figura 23 – Implementação da energia de banda proibida no bloco <i>PV Array</i> para o módulo FS-4122A-3.	56

Figura 24 – Energia de banda proibida comparando Efeito Varshni em (29) e a adaptação da linearização apresentada por (de SOTO et al., 2006) em (28) para módulos de CdTe.	57
Figura 25 – Comparativo das curvas I-V e P-V do modulo FS-4122A-3 simulado usando o bloco <i>P-V array</i> sem considerar a correção da variação de E_g e considerando esta correção (E_{g-novo}).	58
Figura 26 – Informações do <i>datasheet</i> do módulo RSM-100P do fabricante Resun. (a) Dados elétricos em STC. (b) Curva I-V para diferentes irradiâncias. (c) Curva I-V para diferentes temperaturas.	59
Figura 27 – <i>Datasheet</i> do Módulo YINGLI - YL265C-30b - Americano.	60
Figura 28 – <i>Datasheet</i> do Módulo YINGLI - YL265C-30b - Alemão.	60
Figura 29 – Parâmetros do MATLAB para o módulo YL265C-30b.	61
Figura 30 – Comparação entre as curvas I-V e P-V do Módulo YL265C-30b para diferentes fontes de dados.	62
Figura 31 – Configuração de Parâmetros do MATLAB/Simulink.	63
Figura 32 – Avaliação do comportamento transitório da potência elétrica para diferentes variações em degrau dos parâmetros: (a) irradiância (b) temperatura (c) tensão.	64
Figura 33 – Extrator de curvas - <i>PV Analyzer I-V Curve Tracer - PVA 1000S</i> e o sensor de irradiância, temperatura e inclinação sem fio <i>SolSensor</i>	67
Figura 34 – Circuito elétrico do método da carga capacitiva.	68
Figura 35 – Bancada experimental para verificação do extrator de curva com o painel fotovoltaico.	69
Figura 36 – Comparativo de curvas I-V obtidas com extrator de curvas e com osciloscópio.	70
Figura 37 – Bancada experimental externa para ensaios de módulos FV.	71
Figura 38 – Fluxograma proposto para tratamento e filtro de curvas I-V a serem utilizadas como banco de dados do MGNL.	72
Figura 39 – Exemplos de falhas de medição do módulo RSM-100P: (a) erro próximo de V_{oc} , (b) erro próximo de I_{sc}	73
Figura 40 – Correção de falhas de medição do módulo RSM-100P: (a) correção próxima de V_{oc} (b) correção próxima de I_{sc}	74
Figura 41 – Exemplos de falhas de medição do módulo RSM-100P: (a) ausência de medição, (b) incompatibilidade entre a curva I-V e o valor de G medido.	75
Figura 42 – Exemplo de relação entre I_{sc} (A) e G (W/m^2) do módulo RSM-100P utilizada como método de filtragem.	76
Figura 43 – Exemplo de relação entre I_{mpp} (A) e G (W/m^2) do módulo RSM-100P utilizada como método de filtragem.	76
Figura 44 – Distribuição de amostras ao longo da curva I-V para o módulo RSM-100P.	77
Figura 45 – Exemplo de problema de curva I-V com múltiplas medições em um pequeno intervalo de tensão de um módulo RSM-100P.	78

Figura 46 – Distribuição de amostras ao longo da curva I-V para o módulo RSM-100P após tratamento.	79
Figura 47 – Exemplo de tratamento da curva I-V com múltiplas medições em um pequeno intervalo de tensão de um módulo RSM-100P.	80
Figura 48 – Comparativo entre o valor da Curva I-V medido em bancada experimental e valores esperados baseado em método de translação de curva.	81
Figura 49 – Fluxograma básico do MGNL de Silva.	82
Figura 50 – Curvas I-V selecionadas como referência e como treinamento para o primeiro módulo ensaiado RSM-100P (P1).	83
Figura 51 – Curvas I-V selecionadas como referência e como treinamento para o segundo módulo ensaiado RSM-100P (P2).	83
Figura 52 – Curvas I-V selecionadas como referência e como treinamento para o terceiro módulo ensaiado RSM-100P (P3).	84
Figura 53 – Detalhe da caixa de junção de um dos módulos ensaiados do tipo RSM-100P.	85
Figura 54 – Trecho do <i>datasheet</i> do RSM-100P que versa sobre as características mecânicas do módulo.	86
Figura 55 – Curvas I-V selecionadas como referência e como treinamento para o módulo ensaiado YL265C-30b.	86
Figura 56 – Elemento principal da implementação do bloco em Simulink do MGNL: (a) apresentação em Simulink e (b) algoritmo da função de cálculo do bloco.	88
Figura 57 – Fluxograma do algoritmo de função de cálculo do bloco MGNL.	89
Figura 58 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco <i>PV Array</i> com valores distintos de irradiância e temperatura para o módulo RSM-100P (P1).	90
Figura 59 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco <i>PV Array</i> com valores distintos de irradiância e temperatura para o módulo YL265C-30b.	91
Figura 60 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco <i>PV Array</i> e <i>PV Array</i> ajustado com valores distintos de irradiância e temperatura para o módulo RSM-100P (P1).	94
Figura 61 – Esquemático da bancada experimental para extração de curvas com arranjo de quatro módulos RSM-100P.	95
Figura 62 – Bancada experimental para extração de curvas com arranjo de quatro módulos RSM-100P.	96
Figura 63 – Curvas I-V selecionadas como referência e como treinamento para configuração com quatro módulos em série (4x1).	97
Figura 64 – Comparativo entre curvas P-V obtidas através do MGNL e utilizando o bloco <i>PV Array</i> com valores distintos de irradiância e temperatura para quatro módulos RSM-100P (4x1).	98

Figura 65 – Associação de módulos de acordo com o SDM.	99
Figura 66 – Relação entre R_s e n	100
Figura 67 – Relação entre R_{sh} e n	101
Figura 68 – Fluxograma proposto para tratamento e filtro de curvas I-V a serem utilizadas como banco de dados do MGNL conforme organização dos algoritmos. . . .	111

LISTA DE TABELAS

Tabela 1 – Custos nivelados de eletricidade (LCOE) para diferentes tecnologias de geração de energia em 2022.	22
Tabela 2 – Parâmetros da energia de banda proibida para materiais semicondutores Si, GaAs, CdTe e CdS.	53
Tabela 3 – Comparação entre dados do módulo YL265C-30b de diferentes Fontes. . . .	61
Tabela 4 – Propriedades do extrator de curva <i>PVA-1000S</i>	67
Tabela 5 – Soluções comerciais para a aquisição de curvas I-V.	68
Tabela 6 – Características dos módulos ensaiados	71
Tabela 7 – Limites inferiores e superiores utilizados na busca de parâmetros para o modelo de (SILVA et al., 2017) modificado.	81
Tabela 8 – Curvas de referência e treinamento utilizadas no modelo de Silva para cada módulo do <i>RSM-100P</i>	84
Tabela 9 – Parâmetros obtidos a partir dos módulos do modelo RSM-100P ensaiados em ambiente externo.	85
Tabela 10 – Curvas de referência e treinamento utilizadas no modelo de (SILVA, 2019) para o módulo <i>YL265b-30C</i>	87
Tabela 11 – Parâmetros obtidos a partir do módulo do modelo YL265b-30C ensaiado. .	87
Tabela 12 – Avaliação dos parâmetros utilizados e resultados obtidos.	93
Tabela 13 – Curvas de referência e treinamento utilizadas no modelo de (SILVA, 2019) para o módulo RSM-100P (4x1).	96
Tabela 14 – Parâmetros obtidos a partir da configuração 4x1.	97
Tabela 15 – Avaliação de R_s e R_{sh} dos módulos isolados e em série.	99

LISTA DE ABREVIATURAS E SIGLAS

a-Si	Silício Amorfo
CdTe	Telureto de Cádmio
CIGS	Disseleneto de Cobre, Gálio e Índio
DDM	<i>Double Diode Model</i> - Modelo de Célula Fotovoltaica de Dois Diodos
EMAP	Erro Médio Absoluto em Potência
EMAPN	Erro Médio Absoluto em Potência Normalizado
EPE	Empresa de Pesquisa Energética
EPRI	<i>Electric Power Research Institute</i> - Instituto de Pesquisa de Potência Elétrica
EUA	Estados Unidos da América
FV	Fotovoltaico
LCOE	<i>Levelized Cost of Electricity</i> - Custo Nivelado de Eletricidade
HIT	Heterojunção com Película Fina
IEC	<i>International Electrotechnical Commission</i> - Comissão Internacional de Eletrotécnica
IEA	<i>International Energy Agency</i> - Agência Internacional de Energia
MAPS	Modelo Adaptável com Povoamento Superficial
MGNL	Modelo Global Não-Linear
MPP	<i>Maximum Power Point</i> - Ponto de Máxima Potência
m-Si	Silício Policristalino
NREL	<i>National Renewable Energy Laboratory</i> - Laboratório Nacional de Energia Renovável
NRMSD	<i>Normalized Root Mean Square Deviation</i> - Desvio Médio Quadrático Normalizado
PS	<i>Pattern Search</i> - Busca de Padrões
RMSD	<i>Root Mean Square Deviation</i> - Desvio Médio Quadrático

SDM	<i>Single Diode Model</i> Modelo de Célula Fotovoltaica de Um Diodo
STC	<i>Standard Test Conditions</i> - Condições Padrões de Teste
TDM	<i>Three Diode Model</i> - Modelo de Célula Fotovoltaica de Três Diodos
UE	União Europeia
x-Si	Silício Policristalino

LISTA DE SÍMBOLOS

E_g	Energia do <i>gap</i> - Energia de banda proibida
$E_{g,ref}$	Energia do <i>gap</i> de referência
G	Irradiância
I	Corrente elétrica
I_d	Corrente de junção
I_{est}	Corrente estimada
I_{med}	Corrente medida
I_{mpp}	Corrente no ponto de máxima potência
I_{ph}	Corrente fotogerada
$I_{ph,ref}$	Corrente fotogerada de referência
I_{sat}	Corrente de saturação
I_{sat1}	Corrente de saturação referente ao primeiro diodo
I_{sat2}	Corrente de saturação referente ao segundo diodo
I_{sat3}	Corrente de saturação referente ao terceiro diodo
$I_{sat,ref}$	Corrente de saturação de referência
I_{sc}	Corrente de curto-circuito
$I_{sc,ref}$	Corrente de curto-circuito de referência
K	Constante de Boltzmann
K_{R_s}	Coefficiente linear de temperatura para resistência série
$K_{R_{sh}}$	Coefficiente linear de temperatura para resistência <i>shunt</i>
n	Fator de idealidade
N_{pontos}	Número de pontos extraídos da curva I-V real do módulo fotovoltaico
N_s	Número de células em série no módulo fotovoltaico
P_{est}	Potência estimada

P_{med}	Potência medida
P_{mp}	Potência no ponto de máxima potência
q	Carga do elétron
R_s	Resistência série
$R_{s,ref}$	Resistência série de referência
$R_{s,ref1}$	Resistência série referente à irradiância
$R_{s,ref2}$	Resistência série referente à temperatura
R_{sh}	Resistência <i>shunt</i> - Resistência paralelo
$R_{sh,ref}$	Resistência <i>shunt</i> de referência
σ_n	Coefficiente linear de irradiância para fator de idealidade
σ_{R_s}	Coefficiente linear de irradiância para resistência série
S_{ref}	Irradiância de referência
T	Temperatura
T_{ref}	Temperatura de referência
V	Tensão elétrica
V_{mp}	Tensão no ponto de máxima potência
V_{oc}	Tensão de curto aberto
$V_{oc,ref}$	Tensão de circuito aberto de referência
V_t	Tensão térmica
ε_{R_s}	Coefficiente linear de temperatura para resistência série
$\alpha_{I_{mp}}$	Coefficiente térmico de corrente de máxima potência
$\alpha_{I_{sc}}$	Coefficiente térmico de corrente de curto
α_k	Fator adaptativo
α_{E_g}	Coefficiente de variação da energia de gap 1
β_S	Coefficiente de irradiância para tensão de circuito aberto
β_T	Coefficiente térmico de tensão de circuito aberto

β_{E_g}	Coeficiente de variação da energia de gap 2
$\delta_{R_{sh}}$	Coeficiente linear de temperatura para resistência <i>shunt</i>
ϵ_n	Coeficiente linear de temperatura para fator de idealidade
γ_T	Coeficiente térmico de máxima potência
γ_{R_s}	Coeficiente exponencial da irradiância para resistência série
$\gamma_{R_{sh}}$	Coeficiente exponencial da irradiância para resistência <i>shunt</i>
μ_k	Fator de amortecimento

SUMÁRIO

1	INTRODUÇÃO	21
1.1	PROBLEMA E MOTIVAÇÃO	23
1.2	OBJETIVOS	25
1.2.1	Objetivo geral	25
1.2.2	Objetivos específicos	25
1.3	CONTRIBUIÇÕES	25
1.4	ESTRUTURA DO TEXTO	26
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	CURVA I-V	27
2.2	MODELOS BASEADOS EM DIODO	33
2.2.1	Modelo de um diodo (SDM)	33
2.2.1.1	Fator de idealidade do diodo (n)	35
2.2.1.2	Resistência em série (R_s)	36
2.2.1.3	Resistência em paralelo (R_{sh})	36
2.2.2	Modelo de dois diodos (DDM)	37
2.3	DEGRADAÇÃO EM POTÊNCIA EM MÓDULOS FV	38
2.4	MODELOS GLOBAIS NÃO LINEARES	42
2.4.1	MGNL proposto por (SILVA, 2019)	43
2.5	CRITÉRIOS DE AVALIAÇÃO UTILIZADOS	44
2.6	CONCLUSÃO	45
3	DESAFIOS NA MODELAGEM DE SISTEMAS FOTOVOLTAICOS COM BLOCO PV ARRAY DO MATLAB	46
3.1	BLOCO <i>PV ARRAY</i> DO MATLAB	46
3.2	MÉTODO DE TRANSLAÇÃO DE CURVAS	49
3.3	ENERGIA DE BANDA PROIBIDA (E_g)	53
3.4	A ESCOLHA DOS PARÂMETROS DE REFERÊNCIA	58
3.5	OPERAÇÃO DO <i>PV ARRAY</i> DURANTE TRANSITÓRIOS	62
3.6	CONCLUSÃO	65
4	APLICAÇÃO DO MGNL COM EXTRAÇÃO DE DADOS DE MÓDULOS FV	66
4.1	BANCADA EXPERIMENTAL PARA EXTRAÇÃO DE CURVAS I-V	66
4.1.1	Extrator de curvas I-V	67
4.1.2	Montagem da bancada experimental	71
4.2	TRATAMENTO DOS DADOS OBTIDOS	72
4.2.1	Tratamento inicial - Problemas na região de I_{sc} e V_{oc}	73

4.2.2	Triagem de curvas	74
4.2.3	Uniformização das amostras	77
4.3	OBTENÇÃO DOS PARÂMETROS E APLICAÇÃO DO MGNL	78
4.3.1	Análise dos dados	78
4.3.2	MGNL utilizado	79
4.3.3	Parâmetros extraídos	81
4.3.3.1	RSM-100P	82
4.3.3.2	YL265C-30b	86
4.4	CONCEPÇÃO DO MGNL NO MATLAB/SIMULINK	87
4.5	COMPARAÇÃO DOS RESULTADOS DO <i>PV ARRAY</i> E DO MGNL	88
4.6	<i>PV ARRAY</i> AJUSTADO	92
4.7	AVALIAÇÃO COM ARRANJO DE QUATRO MÓDULOS	95
4.8	CONCLUSÃO	101
5	CONCLUSÃO	102
5.1	SUGESTÃO DE TRABALHOS FUTUROS	103
5.2	ARTIGOS PUBLICADOS NO TEMA DO MESTRADO	104
	 REFERÊNCIAS	 105
	 APÊNDICE	 111

1 INTRODUÇÃO

A geração de energia solar fotovoltaica (FV) tem se tornado uma solução cada vez mais relevante e viável no cenário energético global, oferecendo inúmeros benefícios econômicos, ambientais e sociais. Por um lado, é uma fonte considerada limpa e renovável, contrastando significativamente com fontes de energia tradicionais, como carvão e petróleo, que contribuem para a poluição do ar e as mudanças climáticas, e sua adoção pode reduzir significativamente as emissões de dióxido de carbono, ajudando a mitigar os efeitos do aquecimento global (JAN et al., 2023). Por outro lado, a geração de energia solar FV tem se mostrado cada vez mais competitiva economicamente, devido ao avanço das tecnologias e à redução dos custos de produção e instalação dos módulos, o que tem tornado essa energia uma alternativa atraente em comparação com fontes de energia convencionais; segundo a Agência Internacional de Energia, em inglês *International Energy Agency* (IEA), os custos da energia solar caíram nos últimos anos, com a perspectiva de que caiam ainda mais até 2050 (International Energy Agency, 2023).

Neste contexto, pode-se apresentar dados na Tabela 1, em que se extraiu dados do *World Energy Outlook* de 2023 (International Energy Agency, 2023). Cabe ressaltar que o termo Custo Nivelado de Eletricidade (LCOE, do inglês *Levelized Cost of Electricity*) é uma métrica utilizada para comparar o custo total de construção e operação de diferentes tecnologias de geração de energia ao longo da vida útil de um projeto. O LCOE é normalmente expresso em dólares por megawatt-hora (USD/MWh) e considera todos os custos envolvidos, incluindo capital inicial, manutenção, operação e combustível, divididos pela quantidade total de energia gerada durante a vida útil da instalação. Essa métrica permite uma comparação direta entre tecnologias de energia com perfis distintos de custo e produção, ajudando a identificar quais opções são mais econômicas em termos de custo por unidade de eletricidade gerada.

Na Tabela 1 destacam-se as variações nos custos associados a cada tecnologia nas regiões dos Estados Unidos da América (EUA) e União Europeia (UE). Observa-se que a geração FV nos EUA apresenta um LCOE de 50 USD/MWh, enquanto na UE resulta em um LCOE de 65 USD/MWh. Desprende-se desses dados que a energia solar FV é uma opção competitiva e economicamente viável em ambas as regiões, refletindo avanços tecnológicos e economias de escala que têm reduzido os custos de implementação e operação ao longo dos anos.

Além disso, a energia solar FV promove a independência e segurança energética. Ou seja, países com alto índice de radiação solar podem diminuir sua dependência de consumo ou até mesmo importação de combustíveis fósseis, promovendo uma maior autossuficiência energética.

Em outras palavras, a expansão da energia solar FV pode contribuir para a diversificação das matrizes energéticas e reduzir a vulnerabilidade a crises energéticas (LIU et al., 2010). Por outro lado, avaliando aspectos sociais, a energia solar FV pode contribuir para o desenvolvimento econômico local, uma vez que esta indústria tem sido uma grande criadora de empregos, desde a

Tabela 1 – Custos nivelados de eletricidade (LCOE) para diferentes tecnologias de geração de energia em 2022.

Região	Tecnologia	Custo de Capital (USD/kW)	Custo de Combustível, CO2 O&M (USD/MWh)	LCOE (USD/MWh)
EUA	Nuclear	5000	30	105
	Carvão	2100	30	100
	Gás	1000	45	65
	Solar FV	1120	10	50
	Eólica <i>onshore</i>	1220	10	30
	Eólica <i>offshore</i>	4060	35	120
UE	Nuclear	6600	35	160
	Carvão	2000	125	205
	Gás	1000	170	230
	Solar FV	990	10	65
	Eólica <i>onshore</i>	1750	10	45
	Eólica <i>offshore</i>	5000	40	150

Fonte: Adaptado de (International Energy Agency, 2023).

fabricação dos módulos até a instalação e manutenção dos sistemas. Assim, a expansão desta indústria pode contribuir significativamente para o desenvolvimento econômico de regiões com alto potencial de geração de energia solar (International Renewable Energy Agency, 2020).

Neste cenário, deve-se trazer à pauta a importância da modelagem adequada do funcionamento de módulos FV, pois é fundamental para otimizar a eficiência e a confiabilidade dos sistemas de energia solar. Um modelo preciso permite a previsão do desempenho dos módulos em diversas condições climáticas e operacionais, o que é essencial para maximizar a geração de energia e a vida útil dos sistemas FV. Além disso, modelos precisos ajudam na identificação e mitigação de problemas, como sombreamentos, defeitos nos módulos ou até mesmo envelhecimento, melhorando assim a viabilidade econômica dos sistemas. Não obstante, o modelo dos módulos FV também facilita o desenvolvimento de estratégias de controle do ponto de máxima potência (MPP), que podem aumentar significativamente a eficiência operacional da geração de energia (TAHIRI et al., 2017).

Deve-se assim mencionar a importância do MATLAB/Simulink, que é uma ferramenta amplamente utilizada na modelagem e simulação de sistemas FV devido à sua flexibilidade e capacidade de integração com diversas ferramentas de simulação. O bloco *PV Array*, em particular, facilita a representação detalhada das características elétricas, físicas e construtivas dos módulos FV, permitindo uma análise satisfatória do desempenho do sistema sob diversas condições operacionais. Destaca-se que o bloco *PV Array* pode ser utilizado na modelagem de sistemas FV, pois representa módulos FV e oferece uma interface gráfica para a inserção de parâmetros definidos pelo usuário. Difundido a partir da versão do MATLAB R2015b, este bloco tem sido amplamente utilizado para representar a operação de sistemas FV em diversas condições operacionais.

Em contrapartida, existem softwares específicos para FV, como PVSyst e PVSol, que são voltados principalmente para a análise de desempenho e dimensionamento de sistemas FV. Esses softwares fornecem interfaces simples e bancos de dados meteorológicos prontos, além de permitir a análise de produção de energia com base em parâmetros de configuração padronizados.

No entanto, essas plataformas podem ser limitadas quando há a necessidade de explorar em detalhes o comportamento dinâmico de um módulo FV, como a resposta a variações de irradiância e temperatura em tempo real ou a implementação de algoritmos de controle, assim como técnicas de MPPT.

O MATLAB/Simulink, por sua vez, destaca-se por ser uma ferramenta de propósito geral, amplamente utilizada em pesquisas e desenvolvimento de novas tecnologias. Ele permite a implementação de modelos físicos, incluindo o desenvolvimento de novos algoritmos de controle e a simulação de comportamentos não lineares e transientes. Essa flexibilidade é importante para a integração de sistemas FV com outros componentes, como baterias e inversores, além de permitir a validação experimental de novos conceitos.

Assim, em síntese, para análises que precisam de uma visão mais detalhada dos parâmetros do módulo FV, como o desenvolvimento de novos modelos ou a simulação de diferentes cenários, o MATLAB/Simulink é indicado. Já PVSyst e PVSol são ferramentas indicadas para dimensionamento e análise de viabilidade técnica e econômica de sistemas FV, sendo ideais para fases de projeto ou para aplicações comerciais.

1.1 PROBLEMA E MOTIVAÇÃO

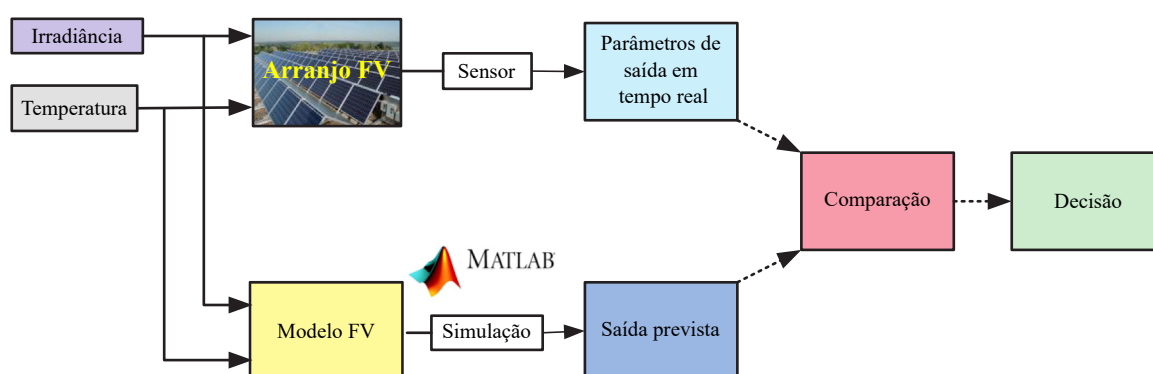
A crescente demanda por energia limpa e sustentável tem impulsionado o desenvolvimento de tecnologias renováveis, como a energia solar FV. Com a adoção crescente de sistemas FV, há uma necessidade por ferramentas que representem com precisão o comportamento dos módulos FV em diversas condições operacionais. Dessa forma, um modelo preciso dos módulos é útil para aplicações de avaliação de produção energética sistemas FV, técnicas de detecção de degradação em potência em módulos e comparação de técnicas de detecção de MPP. Neste contexto, o MATLAB/Simulink é amplamente usado para esse fim, com o bloco *PV Array* do Simulink oferecendo uma interface gráfica para a parametrização de variáveis e modelagem de sistemas FV.

Porém, a modelagem de sistemas FV com o bloco *PV Array* apresenta diversos desafios inerentes aos aspectos construtivos dessa ferramenta. Assim, pode-se destacar a complexidade dos modelos físicos e matemáticos necessários para representar com precisão o comportamento dos módulos FV. Essas dificuldades podem resultar em simulações imprecisas, afetando a capacidade de otimizar e prever o desempenho dos sistemas FV de forma confiável. Uma limitação que pode ser exemplificada é a imprecisão nas translações de curvas para diferentes condições de irradiância e temperatura, além das limitações na representação de módulos de diferentes tecnologias.

As aplicações deste bloco são diversas, podendo ser utilizado para avaliação de plantas solares FV (WADIE, 2023), desenvolvimento e aperfeiçoamento de métodos de rastreamento do MPP (HASSAN et al., 2023) ou até mesmo análise de técnicas de detecção de degradação em potência em módulos FV (HAJJI et al., 2023).

Assim sendo, conforme apresentado por (PILLAI et al., 2018), uma das metodologias utilizadas para detecção de degradação em potência em módulos FV é a medição de diferença baseada em modelo (MBDM, do inglês *Model Based Difference Measurement*). Essencialmente, esta metodologia propõe a comparação entre a curva I-V analisada com uma curva I-V de algum modelo de referência, conforme ilustra a Figura 1.

Figura 1 – Fluxograma da metodologia de detecção de faltas do MBDM.



Fonte: Adaptado de (PILLAI et al., 2018).

Diante desses desafios, é essencial entender o funcionamento e o comportamento do bloco *PV Array* para determinar sua adequação como ferramenta de modelagem de sistemas FV, identificar suas restrições e buscar possíveis soluções alternativas para mitigar essas limitações.

Essas limitações motivam a busca por soluções que possam melhorar a precisão e a confiabilidade dos modelos de sistemas FV. A possibilidade de desenvolver metodologias mais robustas e modelos alternativos que superem as restrições do bloco *PV Array* é uma área de grande interesse tanto para pesquisadores quanto para profissionais da indústria. Melhorar a modelagem desses sistemas contribui para a eficiência energética. Neste contexto, Modelos Globais Não Lineares (MGNL) estão sendo pesquisados, desenvolvidos e publicados com o intuito de aumentar a precisão na representação de módulos, séries e arranjos FV, assim, um novo bloco para o MATLAB/Simulink poderia ser proposto para garantir uma maior precisão nas simulações.

Através deste trabalho, busca-se proporcionar uma base sólida para futuros estudos e aplicações, possibilitando que a modelagem de sistemas FV seja utilizada de maneira mais eficiente. Não obstante, é relevante tornar a utilização do MGNL proposto por (SILVA, 2019) mais prática em cenários concretos, vislumbrando, assim, sair do escopo de simulações.

1.2 OBJETIVOS

Para abordar de forma detalhada os desafios e limitações do trabalho, este trabalho foi estruturado com um objetivo geral, que guia o estudo de maneira ampla, e objetivos específicos, que delimitam as etapas e ações necessárias para alcançar o objetivo principal. A seguir, são apresentados o objetivo geral e os objetivos específicos que orientam esta pesquisa.

1.2.1 Objetivo geral

O objetivo geral deste trabalho é investigar as limitações e desafios na modelagem de sistemas FV utilizando o bloco *PV Array* do MATLAB/Simulink, propondo soluções para melhorar a precisão e a eficácia do modelo, trazendo benefícios diretos nas simulações em áreas que dependem do uso deste bloco.

1.2.2 Objetivos específicos

- Desenvolver uma metodologia de extração e tratamento de curvas I-V para utilização em MGNL, assim como analisar a variabilidade da irradiância e da temperatura ambiental e seu impacto nos sistemas FV modelados com o bloco *PV Array*;
- Identificar as principais limitações do bloco *PV Array* na representação precisa dos módulos FV e apresentar possíveis adaptações para este bloco, vislumbrando melhorar sua acurácia;
- Apresentar uma alternativa à utilização do bloco *PV Array* em simulações, avaliando de maneira comparativa os resultados.

1.3 CONTRIBUIÇÕES

No transcorrer deste trabalho é possível identificar algumas contribuições para o estudo de modelagem de sistemas FV, dentre as quais pode-se destacar:

- Desenvolvimento de uma metodologia de tratamento e triagem de curvas I-V obtidas através de um extrator de curva, em ambiente não controlado;
- Uma análise crítica acerca do uso do Bloco *PV Array* do MATLAB/Simulink com sugestão de correções e alterações para aprimorar a modelagem e acurácia do bloco;
- Implementação de bloco próprio em ambiente MATLAB/Simulink que permite a utilização do MGNL como ferramenta de ambiente de simulação.

1.4 ESTRUTURA DO TEXTO

Os capítulos subsequentes deste trabalho serão divididos em quatro partes. No Capítulo 2, trata-se de uma breve fundamentação teórica de conteúdos pertinentes ao entendimento do trabalho. No Capítulo 3, é apresentado de maneira ampla o funcionamento e as características do bloco *PV Array* do MATLAB/Simulink. Posteriormente, no Capítulo 4, é feita uma análise mais ampla e são apresentadas alternativas para o uso do bloco *PV Array*, podendo assim compreender as aplicações e limitações do MGNL para esse escopo. Em seguida, é feita uma avaliação comparativa de aplicações práticas para o uso do bloco *PV Array* e do MGNL. No último capítulo, são apresentadas as conclusões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo é estruturado em cinco seções, cada uma delas abordando aspectos fundamentais para a compreensão e modelagem de sistemas FV. A primeira seção proporciona uma visão geral sobre o comportamento típico da curva I-V, essencial para avaliar o desempenho de módulos FV sob diversas condições de operação.

Na segunda seção, são explorados modelos baseados em diodos, começando com o modelo de um diodo (SDM, do inglês *Single Diode Model*), que inclui parâmetros como o fator de idealidade do diodo, a resistência em série e a resistência em paralelo. Esta seção detalha como esses elementos influenciam a performance dos modelos dos módulos FV, proporcionando uma base sólida para a análise e otimização de sistemas FV. O modelo de dois diodos (DDM, do inglês *Double Diode Model*) também é apresentado, oferecendo uma abordagem mais complexa e precisa para simular o comportamento dos modelos de módulos FV.

As seções subsequentes abordam a degradação em potência em módulos FV, apresentando uma classificação das faltas comuns e seus impactos no desempenho dos sistemas. Além disso, são introduzidos os MGNL, que ajustam seus parâmetros em resposta às variações ambientais. Por derradeiro, é feita uma breve apresentação acerca dos critérios utilizados neste trabalho para a avaliação de erro.

2.1 CURVA I-V

Primeiramente, a curva I-V, ou curva de corrente *versus* tensão, é uma representação gráfica que descreve o comportamento da relação entre a corrente elétrica (I) e a tensão elétrica (V) geradas por um módulo FV sob diferentes condições de operação. Esta curva é fundamental para a avaliação da corrente de curto-circuito (I_{sc}), da tensão de circuito aberto (V_{oc}) e do MPP (LUQUE; HEGEDUS, 2011).

Desprende-se também a importância da curva I-V na análise de módulos FV e na sua capacidade de fornecer informações detalhadas sobre a eficiência do módulo, assim como acerca de possíveis faltas e degradação ao longo do tempo (GREEN, 1982). Não obstante, a curva I-V é amplamente utilizada na pesquisa e desenvolvimento de novas tecnologias FV, permitindo a comparação de diferentes materiais e projetos de células FV (de Blas et al., 2002).

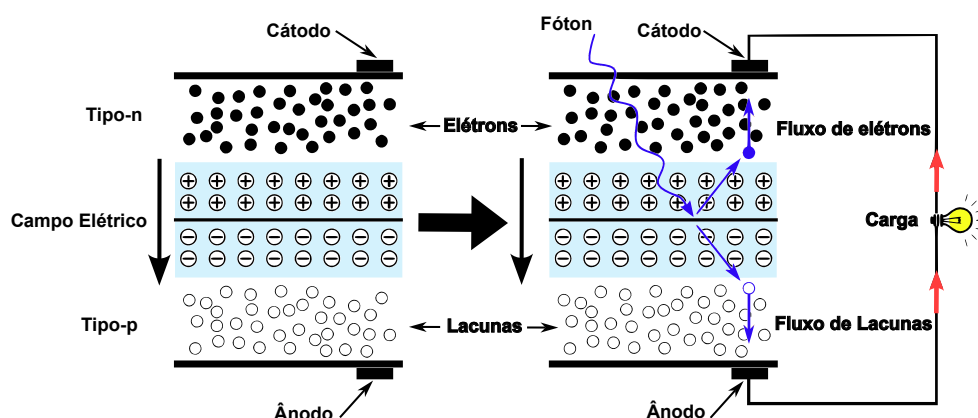
Com o intuito de apresentar o princípio físico por trás destas curvas, deve-se entender que a energia solar pode ser diretamente convertida em energia elétrica em corrente contínua (CC) utilizando materiais semicondutores, que funcionam com base no efeito FV (BLAABJERG, 2018a).

Esse efeito é um processo de conversão de energia em um único passo, em que a energia dos fótons é suficiente para excitar elétrons ligados a materiais sólidos (NELSON, 2003). No entanto, na maioria dos materiais, os elétrons excitados rapidamente retornam ao seu estado

natural. Nos materiais semicondutores, esse efeito é diferente, uma vez que a junção desses materiais, quando excitada, cria uma zona de depleção, devido à migração de elétrons, gerando assim um campo elétrico (WURFEL, 2005; BLAABJERG, 2018a), conforme ilustra a Figura 2.

Logo, as células FV são construídas a partir de materiais semicondutores, tais como Silício (Si), Telureto de Cádmio (CdTe), Disseleneto de Cobre-Índio-Gálio (CIGS), Telureto de Cobre-Índio (CIS), Silício Amorfo (a-Si), entre outros (BLAABJERG, 2018b), havendo assim uma busca constante por materiais melhores, mais eficientes e mais econômicos.

Figura 2 – Processo de conversão de energia em uma célula FV.



Fonte: Adaptado de (BLAABJERG, 2018a).

Dito isto, a corrente elétrica em uma célula FV pode ser entendida como a combinação da corrente proveniente de uma junção PN, representada por um diodo semicondutor, juntamente com a corrente resultante da absorção dos fótons da radiação solar, que é representada por uma fonte de corrente (SILVA, 2019). Cabe ressaltar que a corrente da célula é função da tensão aplicada no elemento semicondutor.

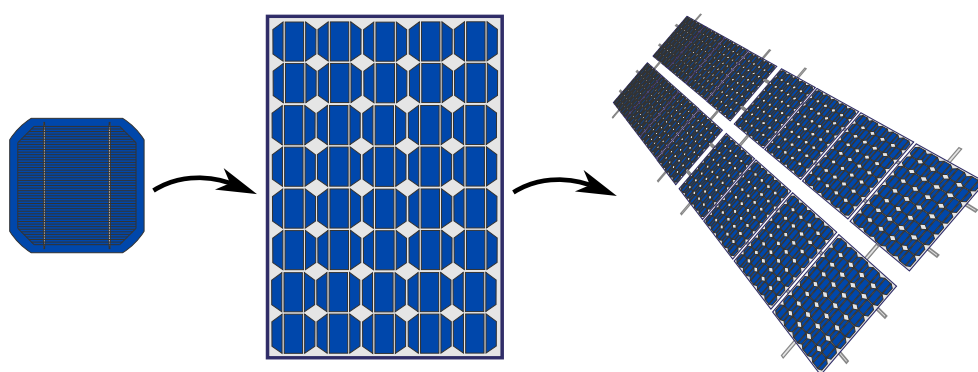
Neste cenário, deve-se trazer à tona que o conceito de corrente de difusão em módulos FV refere-se ao movimento de portadores de carga (elétrons e lacunas) através do material semicondutor, resultante de um gradiente de concentração. Este fenômeno é fundamental para o funcionamento das células solares, pois contribui para a geração de corrente elétrica quando a célula é exposta à luz solar (GREEN, 1982).

Sob outro aspecto, a corrente de fuga, também conhecida como corrente de fuga de junção, é uma corrente indesejada que flui através de um dispositivo semicondutor devido a imperfeições ou defeitos na junção p-n. Essa corrente pode reduzir a eficiência da célula solar ao permitir que a corrente flua através de caminhos não ideais, causando perdas de energia (SCHRODER, 1998).

Neste ponto, é válido salientar que o foco desta seção é apresentar o comportamento da curva I-V com base em seu princípio de funcionamento e, na próxima seção, serão abordados modelos comumente utilizados para representar a operação destas curvas.

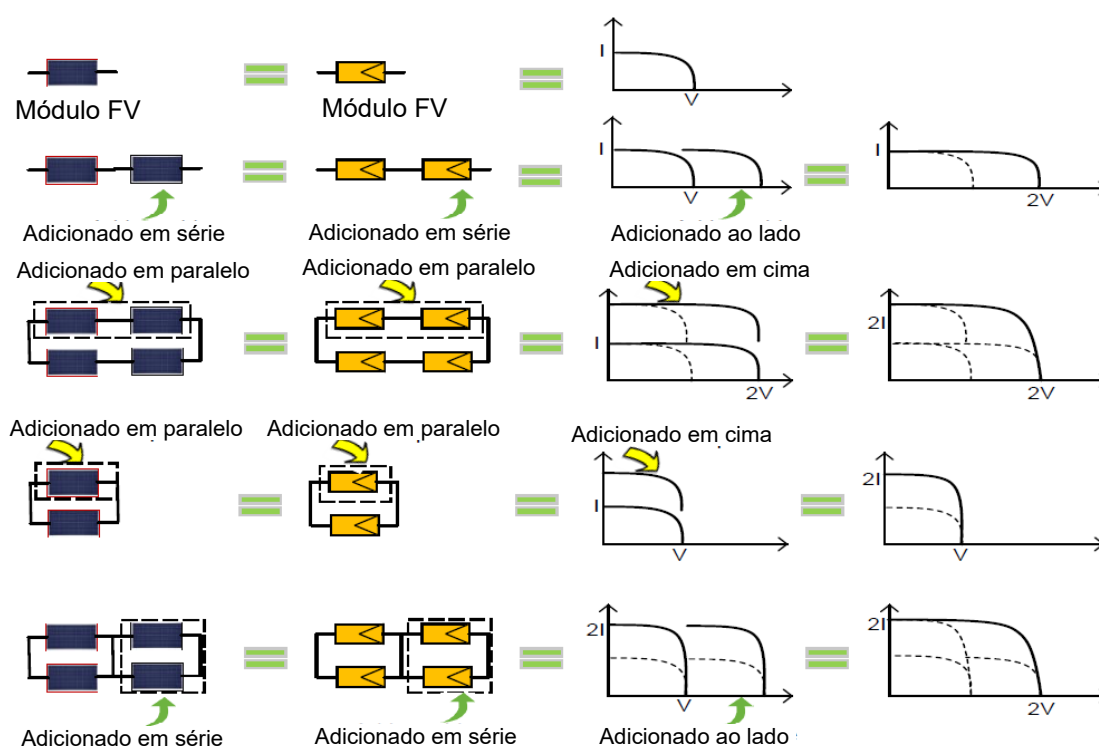
No entanto, deve-se destacar que as células FV são incapazes de gerar, de maneira isolada, potências elevadas. Tipicamente, a tensão fornecida pelas células é da ordem de $0,5\text{ V}$ (BLAABJERG, 2018a). Dito isto, é necessário agrupar células FV em módulos para garantir maiores potências, que por sua vez podem ser organizados em arranjos, como apresenta a Figura 3. Quando esta conexão de módulos é realizada em série (conhecido também como *string*), faz com que haja um aumento da tensão elétrica do arranjo, enquanto a conexão em paralelo eleva a corrente elétrica do mesmo (MATAM, 2020), conforme ilustra a Figura 4.

Figura 3 – Representação do conjunto de células formando um módulo e do conjunto de módulos formando um arranjo.



Fonte: O Autor (2024).

Figura 4 – Configuração de arranjos FV: adição de módulos em série e paralelo.

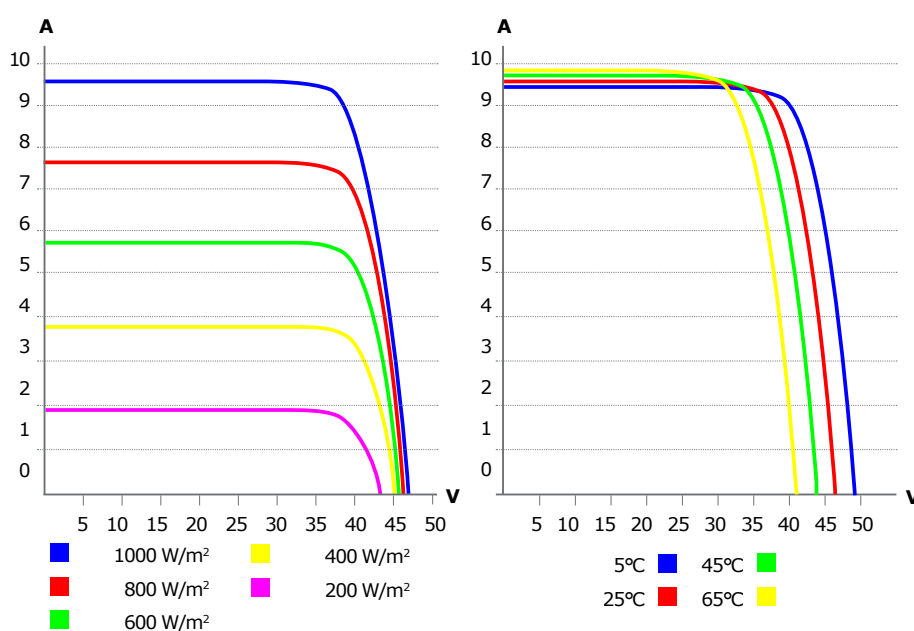


Fonte: Adaptado de (MATAM, 2020).

Neste contexto, é válido mencionar que a curva I-V de um sistema FV não é estática, mas varia de acordo com as condições de operação, como a irradiação solar incidente e a temperatura ambiente (SILVA, 2019), além de fatores construtivos, como resistências dos materiais e qualidade de soldas. Assim sendo, é importante compreender como esses fatores afetam o comportamento da curva I-V, pois isso influencia o desempenho e a eficiência do sistema FV.

Por este motivo, com o intuito de padronizar as informações, os fabricantes de módulos FV costumam apresentar em seus *datasheets* e manuais de operação a curva I-V em apenas um ponto de operação, com temperatura em $25\text{ }^{\circ}\text{C}$ e irradiação em 1000 W/m^2 . Este ponto de operação é conhecido como condições padrão de teste (STC, do inglês *Standard Test Conditions*). Eventualmente, alguns *datasheets* também apresentam outras condições de operação, normalmente fixando a temperatura e variando a irradiação ou fixando a irradiação e variando a temperatura. Como exemplo, a Figura 5 apresenta as curvas I-V do *datasheet* do Módulo Canadian Solar Inc. CS3U-365p.

Figura 5 – Curvas I-V apresentadas no *datasheet* do Módulo Canadian Solar Inc. CS3U-365P.



Fonte: Adaptado de (Canadian Solar, 2020).

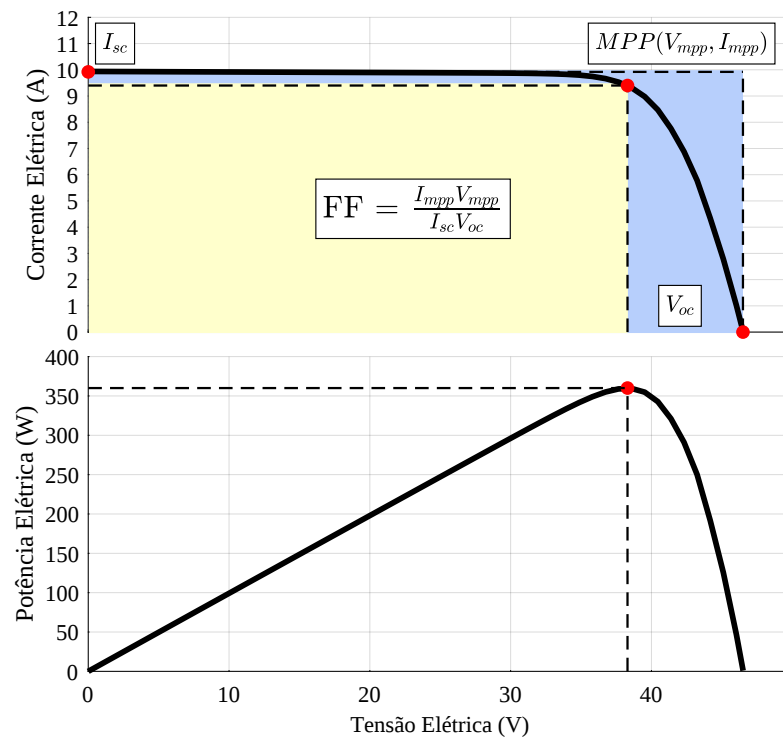
Ademais, é comum que as curvas I-V sejam apresentadas juntamente com as curvas P-V, para que alguns pontos relevantes das curvas fiquem mais visíveis, como, por exemplo o ponto em que a maior potência pode ser transferida para a carga. Além disso, alguns pontos das curvas que podem ser observados e são comumente apresentados nos *datasheets* são eles:

- Tensão de circuito aberto (V_{oc}): É o ponto da curva I-V onde não há fluxo de corrente. É a tensão máxima que a célula ou módulo pode atingir quando nenhum circuito externo é conectado;

- Corrente de curto-circuito (I_{sc}): É o ponto da curva I-V onde a tensão é zero. É a corrente máxima que a célula ou módulo pode fornecer quando a saída está em curto-circuito;
- Ponto de máxima potência (MPP): É o ponto da curva I-V onde a potência elétrica é máxima. Geralmente é representado pela tensão (V_{mpp}) e corrente (I_{mpp}) correspondentes. É um ponto crítico para o dimensionamento e a operação eficiente dos sistemas FV.

Além disso, pode-se também desprender da curva o fator de preenchimento (FF, do inglês *Fill Factor*). O FF é uma medida da qualidade do desempenho de uma célula ou módulo FV e indica a eficiência com que a curva I-V se aproxima de um retângulo. É calculado como a razão entre a potência máxima de saída do dispositivo (P_{mpp}) e o produto da tensão de circuito aberto (V_{oc}) e da corrente de curto-circuito (I_{sc}). Estes pontos relevantes podem ser observados na Figura 6.

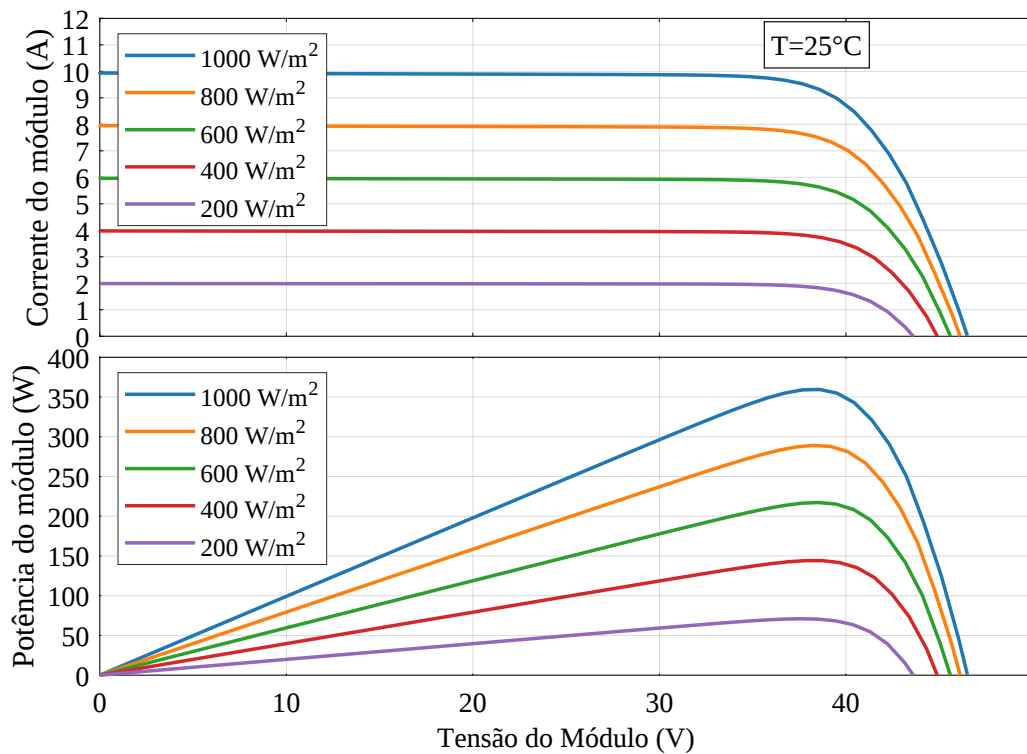
Figura 6 – Exemplo de curva I-V e curva P-V destacando pontos relevantes.



Fonte: O Autor (2024).

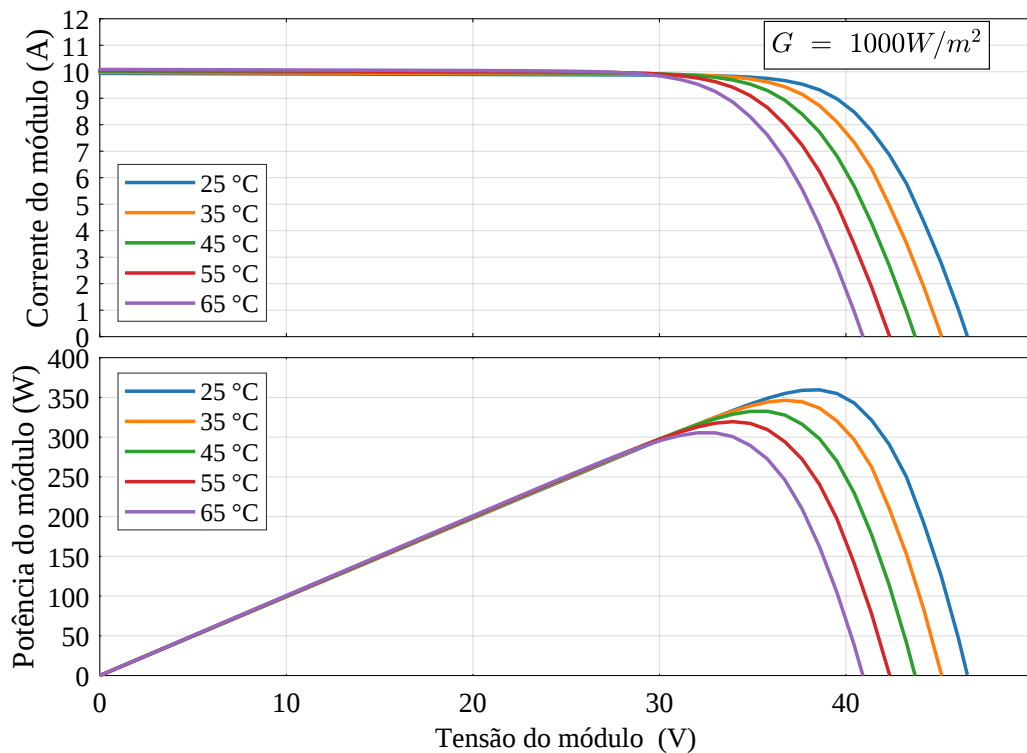
Com o intuito de analisar a influência de fatores ambientais, pode-se apresentar como exemplo os impactos da variação da irradiância e da temperatura na curva I-V. Conforme ilustra a Figura 7, utilizou-se o bloco *PV Array* do MATLAB/Simulink de um módulo Canadian Solar Inc. CS6U-360P, fixou-se o valor da temperatura em 25°C e variou-se a irradiância, enquanto na Figura 8 fixou-se a irradiância em 1000 W/m^2 e variou-se a temperatura. Destaca-se que em ambas as figuras também são apresentadas a curva P-V, para facilitar a compreensão do MPP.

Figura 7 – Comportamento da curva I-V e P-V de um módulo FV com a mudança de irradiância incidente no módulo usando o modelo *PV Array* do MATLAB/Simulink.



Fonte: O Autor (2024).

Figura 8 – Comportamento da curva I-V e P-V de um módulo FV com a mudança da temperatura do módulo usando o modelo *PV Array* do MATLAB/Simulink.



Fonte: O Autor (2024).

Desta forma, pode-se desprender desta análise que a irradiação solar é um fator-chave que afeta a corrente gerada pela célula FV (GREEN, 1982). Em outras palavras, quanto maior a intensidade de radiação solar, maior será a corrente gerada, pois mais fótons são absorvidos e excitam um maior número de elétrons na junção PN (PARIDA; INIYAN; GOIC, 2011). Portanto, ao aumentar a irradiação solar, a curva I-V se desloca para cima, indicando um aumento na corrente para uma determinada tensão aplicada.

No entanto, o aumento da temperatura tende a reduzir a eficiência da célula FV, uma vez que com o este aumento ocorre um aumento na taxa de recombinação de portadores de carga, o que diminui a corrente gerada pela célula. Por isso, a curva I-V se desloca para baixo, indicando uma redução na corrente para uma determinada tensão aplicada (BLAABJERG, 2018b).

Compreender a curva I-V é essencial para a análise detalhada do desempenho dos módulos FV, incluindo a identificação do MPP e possíveis degradações em potência. Essa análise serve de base para a modelagem que será discutida na próxima seção, onde serão apresentados os modelos baseados em diodos. Esses modelos fornecem uma representação mais detalhada e precisa do comportamento elétrico das células FV, permitindo uma melhor compreensão dos sistemas FV como um todo.

2.2 MODELOS BASEADOS EM DIODO

Dando continuidade ao estudo da curva I-V, esta seção foca nos modelos baseados em diodos, que são amplamente utilizados para simular o comportamento elétrico das células FV. Os SDM e DDM serão explorados com mais detalhes, destacando como cada parâmetro desses modelos influencia a curva I-V e, consequentemente, o desempenho dos módulos FV.

Dito isto, é importante abordar os conceitos dos modelos baseados em diodos de módulos FV. Na literatura científica é comum encontrar as abordagens do SDM e DDM, contudo, existem outros modelos, como o de três diodos (TDM, do inglês *Triple Diode Model*) e o de matriz de diodos, também conhecido como modelo multidimensional (GHOLAMI et al., 2022).

No entanto, os SDM e DDM são mais utilizados devido à sua simplicidade e acurácia satisfatória para os casos analisados (GHOLAMI et al., 2022). Por este motivo, este tópico se restringirá a apresentar estes modelos.

2.2.1 Modelo de um diodo (SDM)

No SDM, a célula FV ideal é representada por uma fonte de corrente e um diodo em paralelo, conforme ilustrado na Figura 9. Assim sendo, baseando-se na Equação de Shockley para um diodo ideal (SILVA, 2019), sua corrente de saída pode ser definida por

$$I = I_{ph} - I_d = I_{ph} - I_{sat} \left(e^{\frac{qv}{nkT}} - 1 \right), \quad (1)$$

representando assim a relação de corrente e tensão em uma célula ideal, em que são apresentados:

- I - Corrente elétrica da célula FV ideal [A];
- I_{ph} - Corrente elétrica fotogerada da célula FV [A];
- I_{sat} - Corrente elétrica fotogerada da célula FV [A];
- q - Carga do elétron ($1,6 \cdot 10^{-19} \text{ C}$);
- v - Tensão elétrica de saída da célula FV [V];
- n - Fator de idealidade do diodo;
- k - Constante de Boltzmann ($1,38 \cdot 10^{-23} \text{ m}^2 \text{ kg s}^{-2} \text{ K}^{-1}$);
- T - Temperatura [K].

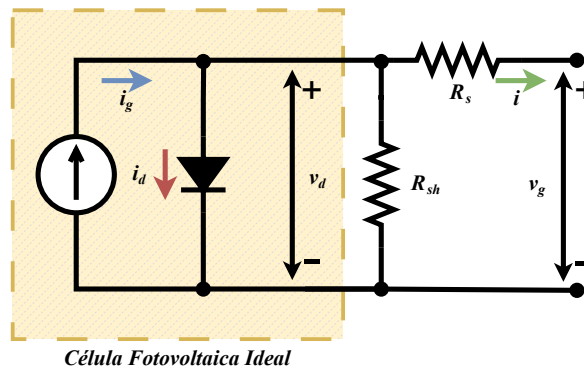
Contudo, conforme já ilustrado na Figura 9, para que o modelo seja mais aderente com o caso concreto, são adicionados ao circuito uma resistência em série (R_s) e uma em paralelo (R_{sh}), possibilitando assim que sejam representadas as perdas do circuito e as correntes de fuga da junção PN (SILVA, 2019). Assim, pode-se desenvolver (1) inserindo tais elementos

$$I = I_{ph} - I_{sat} \left(e^{\frac{q(v+R_s I)}{nkT}} - 1 \right) - \frac{v + R_s I}{R_{sh}}, \quad (2)$$

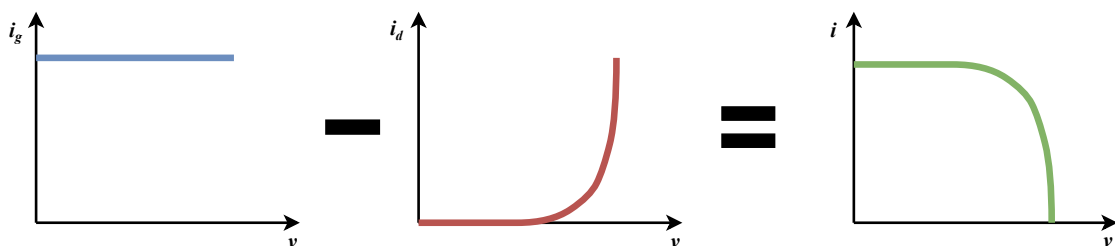
em que, conforme já mencionado, se define R_s com Resistência em série [Ω] e R_{sh} como Resistência em paralelo [Ω].

Figura 9 – Representação do circuito elétrico do SDM.

(a) Circuito elétrico equivalente.



(b) Curva corrente por tensão.



Fonte: O Autor (2024).

Pode-se perceber que alguns parâmetros podem alterar o comportamento da curva I-V. Na seção anterior foram apresentadas a influência da temperatura e da irradiância, parâmetros relacionados ao ambiente e a condições climáticas. A partir de agora serão abordados parâmetros associados a características construtivas dos módulos.

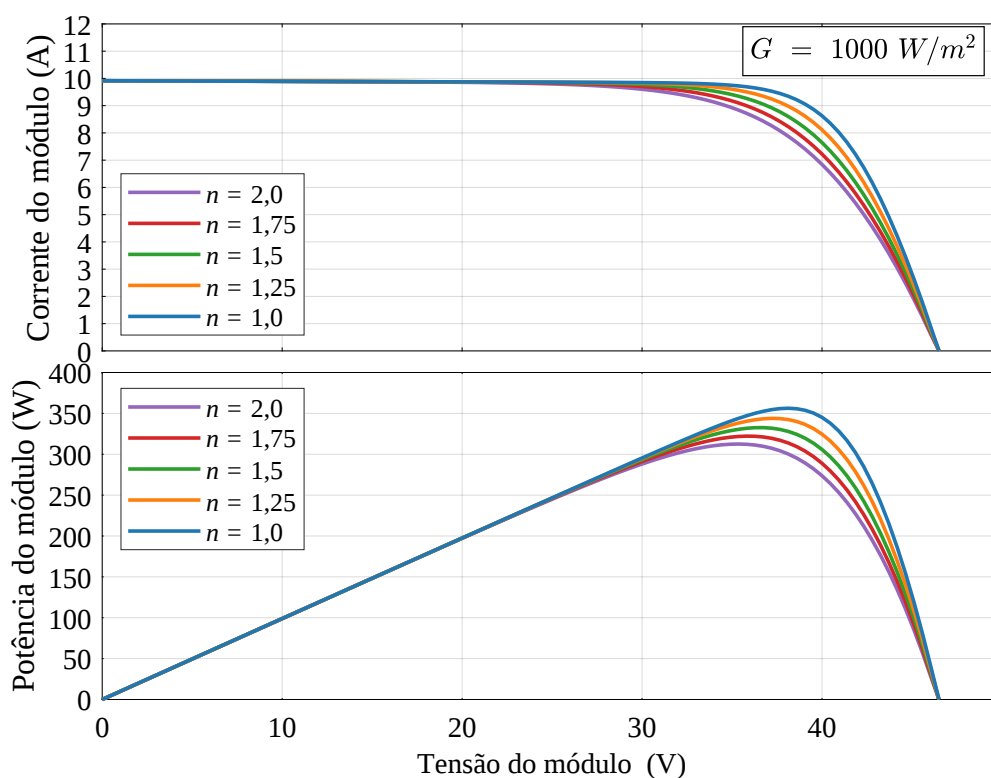
2.2.1.1 Fator de idealidade do diodo (n)

Primeiramente, o n é um parâmetro que relaciona o comportamento real de um diodo com seu comportamento ideal (SILVA, 2019). Deve-se salientar que também é um parâmetro intrinsecamente relacionado ao material semiconductor utilizado na construção das células, mais especificamente ao processo de recombinação da sua junção p-n (JUNIOR, 2018).

Conforme pode-se observar na Figura 10, o n influencia de maneira direta na posição e curvatura do joelho da curva I-V (SILVA, 2019). À medida que o valor de n aumenta, a curva I-V se desvia mais do comportamento ideal. Quando n é mais próximo do valor unitário, a curva I-V mostra uma transição mais acentuada entre a região de corrente constante e a região de decaimento exponencial, próximo à região de MPP.

Por outro lado, para valores maiores de n , a curva se torna mais suave e a corrente decai mais gradualmente com o aumento da tensão, indicando maiores imperfeições do diodo. Isso resulta em uma eficiência menor do módulo. Portanto, um n mais próximo do unitário está associado a um melhor desempenho do módulo FV.

Figura 10 – Comportamento da curva I-V e P-V de um módulo FV com a mudança do n .



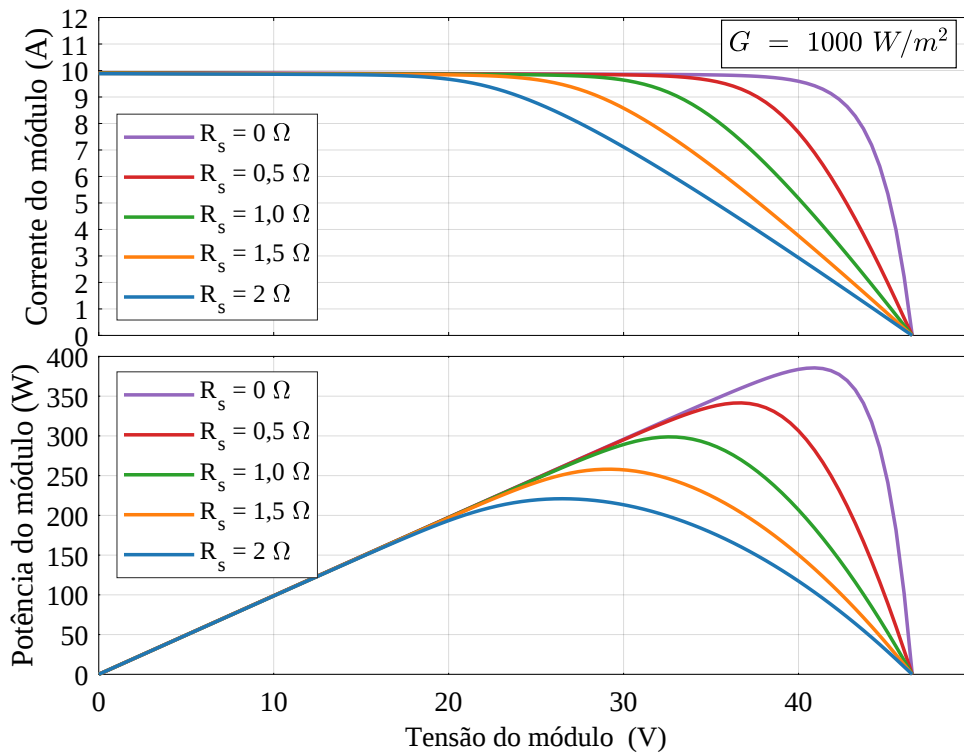
Fonte: O Autor (2024).

2.2.1.2 Resistência em série (R_s)

Outro parâmetro que influencia o formato da curva I-V é a R_s . Esta resistência limita a eficiência do módulo e está correlacionada às resistências dos materiais e dos contatos que formam a célula FV (JUNIOR, 2018). Conforme ilustra a Figura 11, nota-se que com o aumento de R_s a inclinação da curva I-V na região de alta corrente se torna mais acentuada; por outro lado, quando R_s é menor, a curva I-V se aproxima do comportamento ideal, com uma queda abrupta na corrente apenas quando a tensão se aproxima de V_{oc} .

No entanto, para valores maiores de R_s , a corrente diminui mais rapidamente com o aumento da tensão, resultando em uma redução significativa na corrente máxima que o módulo pode fornecer. Isso indica que módulos com R_s mais altas apresentam maiores perdas resistivas, resultando em menor eficiência e desempenho global reduzido.

Figura 11 – Comportamento da curva I-V e P-V de um módulo FV com a mudança de R_s .

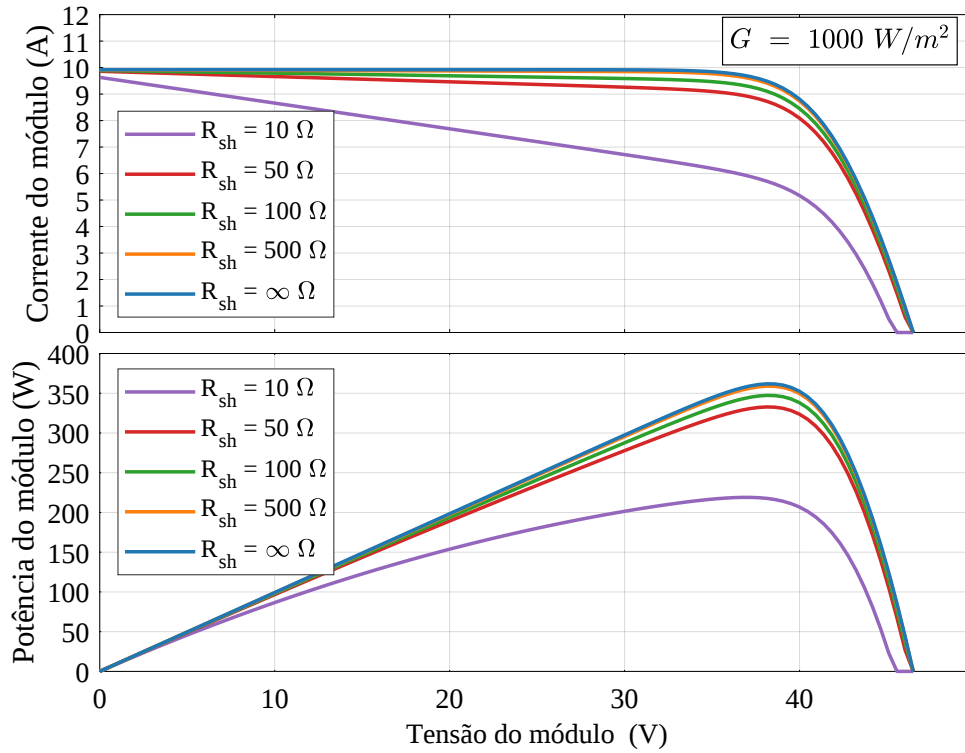


Fonte: O Autor (2024).

2.2.1.3 Resistência em paralelo (R_{sh})

A resistência R_{sh} desempenha um papel essencial na determinação da forma da curva I-V de um módulo FV. Pode-se observar, analisando a Figura 12, que, à medida que o valor de R_{sh} aumenta, a inclinação da curva I-V na região de baixa tensão se torna menos pronunciada. Por exemplo, quando R_{sh} é menor, há uma queda mais acentuada na corrente para pequenos aumentos na tensão, indicando maiores perdas devido a correntes de fuga internas.

Figura 12 – Comportamento da corrente e da potência de um módulo FV com a mudança de R_{sh} .



Fonte: O Autor (2024).

Não obstante, com R_{sh} maiores, a curva I-V se aproxima mais do comportamento ideal, com uma corrente quase constante até que a tensão atinja o ponto de inflexão. Isso sugere que módulos com resistências paralelas mais altas são mais eficientes, pois têm menores perdas de corrente e, conseqüentemente, melhor desempenho.

2.2.2 Modelo de dois diodos (DDM)

Wolf e Rauschenbach (1963) demonstram que a corrente elétrica gerada por uma célula FV não é determinada somente por um diodo, mas também como efeito de diversos diodos distribuídos ao longo da junção PN (HUMADA et al., 2016). Assim sendo, foi desenvolvido um circuito equivalente simplificado com dois diodos, conforme apresentado na Figura 13.

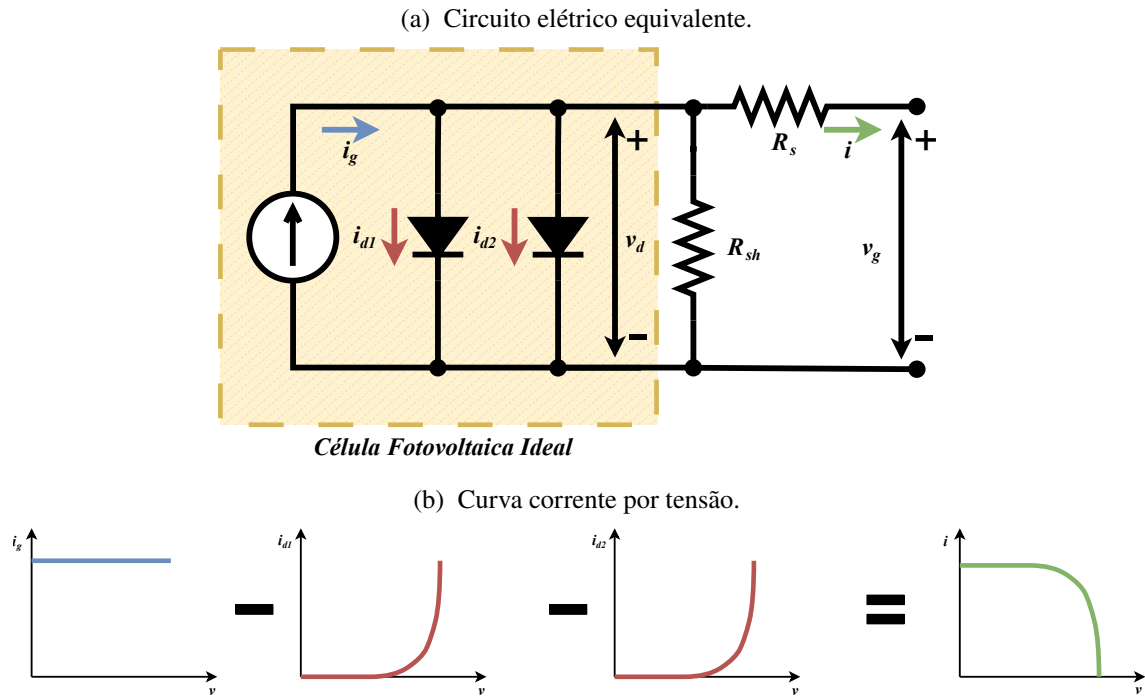
Isto posto, de maneira análoga ao SDM, pode-se desenvolver

$$\begin{aligned} I &= I_{ph} - I_{d1} - I_{d2} \\ &= I_{ph} - I_{sat1} \left(e^{\frac{qv}{n_1 kT}} - 1 \right) - I_{sat2} \left(e^{\frac{qv}{n_2 kT}} - 1 \right), \end{aligned} \quad (3)$$

que considerando as perdas do sistema representadas pelos elementos resistivos fica

$$I = I_g - I_{sat1} \left(e^{\frac{q(v+R_s I)}{n_1 kT}} - 1 \right) - I_{sat2} \left(e^{\frac{q(v+R_s I)}{n_2 kT}} - 1 \right) - \frac{v + R_s I}{R_{sh}}. \quad (4)$$

Figura 13 – Representação do circuito elétrico do DDM.



Fonte: O Autor (2024).

O SDM e o DDM são ambos utilizados para descrever o comportamento elétrico de células FV, mas diferem na complexidade e precisão, pois o DDM incorpora dois diodos, além das mesmas resistências, permitindo uma representação mais detalhada dos mecanismos dentro da célula FV. Essa complexidade adicional torna o DDM mais preciso na modelagem de células solares, especialmente em condições de baixa irradiância, onde os efeitos de recombinação são mais evidentes (de Blas et al., 2002).

Os modelos baseados em diodos fornecem uma base para a análise e otimização de sistemas FV. No entanto, para uma melhor compreensão do desempenho dos módulos FV ao longo do tempo, é necessário considerar a degradação que pode ocorrer nos módulos. A próxima seção aborda esses aspectos, apresentando uma visão mais minuciosa dos mecanismos de degradação e das faltas comuns em sistemas FV.

2.3 DEGRADAÇÃO EM POTÊNCIA EM MÓDULOS FV

A análise dos modelos fornece uma compreensão dos princípios de funcionamento das células FV. Porém, é importante investigar como esses módulos se comportam ao longo do tempo, considerando os efeitos da degradação em potência. Esta seção apresenta um breve resumo das principais causas de defeitos que afetam o desempenho dos módulos FV.

A compreensão adequada do desempenho das diferentes tecnologias FV em condições externas é fundamental para quantificar o desempenho a longo prazo da usina FV, a degradação real em campo, a confiabilidade e a durabilidade dos sistemas FV, bem como para a avaliação de viabilidade e projeto de futuras usinas (BANSAL; JAISWAL; SINGH, 2021).

Entre os anos de 2000 e 2010, em cerca de 200 estudos distintos em 40 países diferentes, registraram-se mais de 11 mil taxas de degradação distintas. Diversos estudos em sistemas FV com mais de 20 anos, e outros com até 30 anos, indicam uma vida útil considerável e valor residual após 25 anos, enquanto outros relatam o declínio prematuro de instalações FV existentes devido a defeitos de fabricação ou práticas de instalação inadequadas (JORDAN et al., 2016).

Neste ponto, deve-se citar que as definições e as caracterizações de falha não são um consenso na literatura científica, pois não há uma definição clara para o termo falha (KOESTER et al., 2022). O IEA apresenta em sua definição a falha (em inglês, *failure*) em módulos FV como um efeito que degrada a potência do módulo sem reversão ou causa um problema de segurança. Assim, questões que não afetam a potência de maneira irreversível ou impactam na segurança não são consideradas falhas. Além disso, problemas causados por manuseio inadequado ou condições ambientais extremas, como sujidade ou sombreamento, não são classificados como falhas de módulo, pois não se relacionam com o desempenho normal esperado do módulo (KÖNTGES et al., 2014; International Energy Agency, 2023).

Então, de acordo com a definição da IEA, o sombreamento não é considerado uma falha no módulo, pois, mesmo que degrade a potência do módulo, não é uma situação irreversível, tampouco causa problemas de segurança.

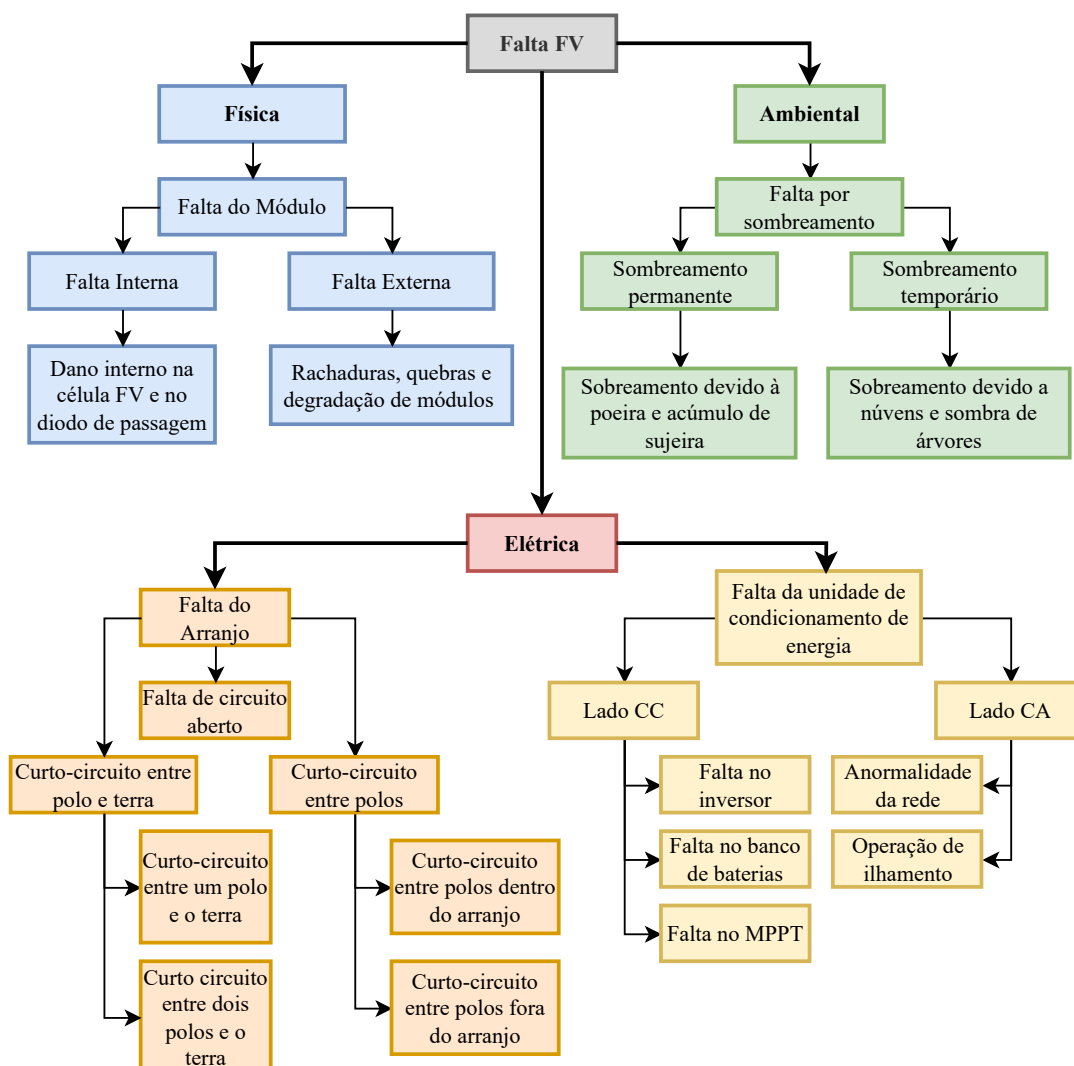
Alguns autores, tais como (LINDIG et al., 2018), descrevem que o termo falha em dispositivos eletrotécnicos é definido como a perda da capacidade de um item realizar uma função requerida. Embora essa definição seja clara para a maioria dos dispositivos, a falha de um módulo FV é um pouco mais complexa. Por exemplo, mesmo que um módulo FV continue tecnicamente em funcionamento, sua produção de energia pode ser tão baixa que inviabiliza sua operação do ponto de vista econômico. Na mesma linha, (JORDAN et al., 2016) relata a dificuldade de definir o termo falha e reforça a definição de falha como a perda da capacidade de um item realizar a função requerida. No entanto, o referido autor também destaca que essa definição não é tão clara para módulos FV.

Outro termo que é recorrente na literatura é falta (em inglês, *fault*), que, assim como falha, não tem uma definição objetiva quando aplicado a sistemas FV. Apesar disso, diversos artigos apresentam categorizações e classificações de faltas. Um exemplo, pode ser apresentado na classificação de faltas utilizada por (PILLAI et al., 2018), conforme desprende-se da Figura 14, pode-se partir das suas origens e efeitos no desempenho dos sistemas.

Destaca-se que falta aqui tem um sentido muito mais amplo que o conceito de falha apresentado pelo IEA, pois aqui apresenta-se o sombreamento como uma possível causa de faltas assim com problemas em outros elementos do sistema, tais como inversores, banco de baterias e até mesmo o lado CA da rede.

Inicialmente, de acordo com (PILLAI et al., 2018), as faltas podem ser divididas em três grandes categorias: físicas, ambientais e elétricas. As faltas físicas incluem as faltas internas e externas dos módulos, enquanto as ambientais são causadas por sombreamento, que pode ser

Figura 14 – Classificação de faltas em sistemas FV de acordo com (PILLAI et al., 2018).



Fonte: Adaptado de (PILLAI et al., 2018).

permanente ou temporário, e, por fim, as faltas elétricas podem ser problemas com os arranjos ou com o sistema de condicionamento de potência.

Conforme mencionado, as faltas físicas dos módulos são subdivididas em faltas internas e externas. Faltas internas referem-se a danos dentro das células FV e nos diodos de passagem (*by-pass*). Por outro lado as faltas externas incluem rachaduras nos módulos, módulos quebrados e degradados.

Em outro prisma, as faltas ambientais são principalmente causadas por sombreamento, que pode ser permanente, devido a acumulações de sujeira, ou temporário, causado por movimento de nuvens ou sombras de árvores.

Não obstante, existem as faltas elétricas que podem ser circuitos abertos, ou curto-circuitos entre um ou dois polos e o terra, entre polos de um mesmo arranjo ou entre polos de arranjos distintos. Neste ponto, cabe fazer o adendo que o termo polo utilizado aqui em

circuitos CC se refere ao terminal positivo e negativo do sistema. Por fim, ainda existem as falta no condicionamento de energia, como, por exemplo, faltas no conversor, no MPPT, no banco de baterias ou até mesmo anormalidades na rede elétrica.

Outros autores apontam classificações semelhantes àquela utilizada por (PILLAI et al., 2018), tais como (KURUKURU et al., 2020), (MEMON et al., 2022), (DHOUNDIYAL et al., 2022), (MERIBOUT et al., 2023). Deve-se também mencionar que o IEA apresenta uma abordagem similar em para esta classificação (International Energy Agency, 2021).

Não obstante, também é importante mencionar uma questão semântica e linguística, pois os termos falta e falha português são sinônimos a depender do contexto. De acordo com o dicionário Houaiss, a palavra falha pode ser definida como interrupção de funcionamento normal e ter como sinônimos defeito, falência ou falta.

No entanto, em sistemas elétricos, uma falta refere-se a um curto-circuito ou uma interrupção inesperada no fluxo de corrente elétrica. Esse tipo de ocorrência pode ser devido a falhas no isolamento, defeitos em equipamentos ou condições climáticas adversas, como raios e ventos fortes. As faltas podem causar danos significativos aos equipamentos e interrupções no fornecimento de energia (GLOVER et al., 2011).

Assim, por conta dessas divergências, (JORDAN et al., 2016) opta por utilizar o termo "modos de degradação" para referir-se às alterações observáveis na aparência, desempenho e segurança de um módulo.

Desta forma, considerando que este trabalho é focado na análise de curvas I-V e, para evitar conflitos semânticos, entende-se como adequado se desprender dos conceitos de falhas e faltas, e utilizar a terminologia degradação em potência, definida como qualquer evento que reduza de maneira permanente ou temporária a potência elétrica gerada pelo módulo FV. Então, o sombreamento de módulo FV será considerado uma degradação em potência, mesmo que não seja caracterizado como uma falha no módulo FV, nos moldes da definição do IEA.

Assim sendo, entende-se que a categorização de degradações em potência é essencial para o diagnóstico e a mitigação eficiente de problemas em sistemas FV, garantindo a eficiência e a longevidade dos sistemas. Diversos estudos na literatura têm se dedicado a categorizar os mecanismos pelos quais os módulos FV degradam, resultando em perda de capacidade de geração de potência (JORDAN et al., 2016; NDIAYE et al., 2013; DHOUNDIYAL et al., 2022).

A compreensão dos mecanismos de degradação em potência de módulos FV é fundamental para o desenvolvimento de estratégias de manutenção e para a melhoria da confiabilidade dos sistemas FV. Com isso, pode-se avançar para a próxima seção, que introduz os MGNL. Esses modelos adaptam seus parâmetros em resposta às condições ambientais, oferecendo uma abordagem mais precisa para a avaliação do desempenho dos módulos, podendo ser utilizados como ferramenta para avaliação de degradações de funcionamento de sistemas FV.

2.4 MODELOS GLOBAIS NÃO LINEARES

Após analisar a degradação em potência nos módulos FV, é essencial explorar métodos de modelagem que considerem essas variáveis dinâmicas. Nesta seção, discutimos os MGNL, que utilizam parâmetros adaptativos para prever com precisão o comportamento elétrico dos módulos FV sob diversas condições ambientais. Esses modelos representam uma evolução significativa em relação aos modelos tradicionais.

Os MGNL são modelos de módulo FV capazes de representar o comportamento elétrico do módulo FV sob diversas condições ambientais, como temperatura e irradiância variáveis. Diferente de modelos clássicos de circuitos, os MGNL adaptam seus parâmetros em resposta às variações ambientais, proporcionando maior precisão na simulação de desempenho dos módulos FV (JUNIOR, 2023; JUNIOR, 2018; SILVA, 2019).

Os MGNL podem utilizar os mesmos circuitos básicos dos modelos tradicionais, como o SDM, mas com parâmetros que variam conforme a temperatura e a irradiância. Essa adaptação é feita através de equações não-lineares específicas para cada parâmetro, que permitem uma estimativa mais precisa do comportamento dos módulos FV sob diferentes condições (JUNIOR, 2023; JUNIOR, 2018).

Um exemplo é apresentado em (SILVA, 2019), onde um MGNL incorpora 15 parâmetros, incluindo R_s , R_{sh} , n , I_{sc} e V_{oc} , dependentes de irradiância e/ou temperatura. No entanto, estes parâmetros são obtidos por algoritmos de otimização que ajustam os parâmetros com base em um conjunto limitado de curvas I-V obtidas sob diferentes condições ambientais (SILVA, 2019).

Neste ponto, deve-se mencionar que o método proposto por (SILVA, 2019) depende de um banco de curvas I-V. Assim sendo, para sanar essa necessidade, o autor utilizou o banco de dados do Laboratório Nacional de Energias Renováveis (NREL, do inglês *National Renewable Energy Laboratory*), que são obtidos por medições de campos em plantas distintas espalhadas pelo EUA.

Os MGNL são particularmente úteis em aplicações onde a precisão e a flexibilidade são cruciais, como na detecção de degradação em potência, manutenção preditiva e maximização da eficiência dos sistemas FV. Por exemplo, ao implementar o MGNL em controladores de inversores FV, é possível otimizar o MPP e detectar anomalias como sombreamento parcial em módulos (SILVA, 2019).

No entanto, deve-se mencionar que diversas metodologias foram propostas para obtenção de MGNL, cada qual com sua especificidade, vantagens e desvantagens. Pode-se exemplificar com os seguintes trabalhos publicados (ALI et al., 2017), (WANG et al., 2021), (ZHANG et al., 2022a), (ZHANG et al., 2022b), e o trabalho (GHOLAMI et al., 2022) apresenta um compêndio de publicações sobre o tema.

2.4.1 MGNL proposto por (SILVA, 2019)

O MGNL apresentado em (SILVA, 2019) oferece uma modelagem de módulos FV, abordando simultaneamente os efeitos da irradiância e temperatura. Diferentemente de modelos anteriores que utilizavam expressões puramente matemáticas sem base física, este MGNL integra equações de extrapolação de parâmetros diretamente no algoritmo de otimização, utilizando múltiplas curvas I-V como dados de entrada. Isso resulta em um modelo mais preciso, capaz de prever o comportamento dos módulos sob diversas condições ambientais sem a necessidade de repetidas estimações de parâmetros.

As equações que descrevem o MGNL são formuladas para incluir a variação da R_s e R_{sh} com a irradiância e temperatura, além de outros parâmetros importantes como I_{sc} e V_{oc} . Desta forma, (SILVA, 2019) apresenta as equações que regem o comportamento do modelo conforme:

$$R_s = R_{s,ref2} [1 + k_{Rs}(T - T_{ref})] + R_{s,ref1} \left(\frac{G}{G_{ref}} \right)^{\gamma_{Rs}}, \quad (5)$$

$$R_{sh} = R_{sh,ref} [1 + k_{Rsh}(T - T_{ref})] \left(\frac{G}{G_{ref}} \right)^{\gamma_{Rsh}}, \quad (6)$$

$$n = n_{ref}, \quad (7)$$

$$I_{sc} = [I_{sc,ref} + \alpha_i(T - T_{ref})] \left(\frac{G}{G_{ref}} \right), \quad (8)$$

$$V_{oc} = V_{oc,ref} + \beta_T(T - T_{ref}) + \beta_S V_t \ln \left(\frac{G}{G_{ref}} \right), \quad (9)$$

$$I_g = I_{sc} \left(1 + \frac{R_s}{R_{sh}} \right), \quad (10)$$

$$I_{sat} = \frac{I_g - \frac{V_{oc}}{R_{sh}}}{e^{\frac{V_{oc}}{V_t}} - 1}. \quad (11)$$

Para exemplificar, R_s é modelada como uma função de referência e temperatura, ajustada pela irradiância. Este detalhamento matemático permite que o MGNL mantenha a precisão física dos parâmetros sob diferentes condições operacionais.

O processo de estimação dos parâmetros do MGNL é realizado através de um algoritmo de otimização de busca por padrões (PS, do inglês *Pattern Search*), cabendo aqui a ressalva de que outros algoritmos foram testados em (JUNIOR, 2023). A estimação é feita uma única vez usando um banco de dados de curvas I-V, o que reduz a necessidade de reestimativas constantes, uma limitação comum em modelos anteriores. Esta abordagem não só melhora a eficiência computacional, mas também aumenta a aplicabilidade prática do modelo.

A validação do MGNL foi realizada comparando suas previsões com dados experimentais de módulos FV de diferentes tecnologias, sendo importante ressaltar que este modelo foi validado também em trabalhos como (BARBOSA, 2023), (JUNIOR, 2023) e (FERNANDES, 2024). Assim sendo, os resultados mostram que o MGNL pode descrever com precisão o comportamento dos módulos para condições ambientais diversas. Além disso, quando comparado a outros modelos adaptativos, o MGNL demonstrou uma precisão semelhante, mas com menor complexidade computacional, destacando-se como uma solução eficaz ao problema.

Por fim, o MGNL representa uma evolução na modelagem de módulos FV, oferecendo uma solução que combina precisão, flexibilidade e aplicabilidade prática. No entanto, para validar a eficácia desses modelos e comparar diferentes abordagens, é necessário utilizar critérios de avaliação bem definidos. A seção seguinte se dedica à apresentação dos critérios de avaliação utilizados neste trabalho para medir a precisão e eficácia dos modelos discutidos.

2.5 CRITÉRIOS DE AVALIAÇÃO UTILIZADOS

Após a apresentação do MGNL, torna-se necessário estabelecer critérios para a avaliação da eficácia desses modelos. Esta seção apresenta os métodos de avaliação utilizados neste trabalho, o Erro Médio Absoluto em Potência (EMAP) e o Desvio Médio Quadrático (RMSD, do inglês *Root Mean Square Deviation*). Esses critérios são aplicados para validar a precisão dos modelos e garantir que eles possam ser aplicados de maneira confiável.

A priori, a avaliação pode partir do erro absoluto ($erro_a$), conforme

$$erro_a = |P_{curva} - P_{modelo}|, \quad (12)$$

onde o subscrito *curva* indica o valor real da curva e o subscrito *modelo* referencia os valores obtidos através do modelo proposto.

Assim sendo, para o caso concreto da avaliação de curvas P-V, pode-se definir o EMAP por

$$EMAP = \frac{\sum_{j=1}^{N_{pontos}} erro_a}{N_{pontos}}, \quad (13)$$

em que $erro_a$ é calculado para todos os pontos, conforme (12), da tensão indo de zero até a condição de circuito aberto, e N_{pontos} é o número de pontos extraídos da curva I-V real do módulo FV. Em síntese, é a soma de todos os erros absolutos ao longo da curva P-V (SILVA, 2019).

No entanto, desta forma, não é viável comparar duas curvas P-V de módulos de ordens de potência distintas; dessa forma, pode-se normalizar pela potência máxima do sistema em análise, dando origem assim ao Erro Médio Absoluto em Potência Normalizado (EMAPN) (JUNIOR, 2023), que pode ser definido como

$$\text{EMAPN} = \frac{\text{EMAP}}{P_{mp}} \times 100\%. \quad (14)$$

Além disso, outra figura de mérito que é aderente à comparação de curvas I-V é o RMSD (MAHMOUD; XIAO; ZEINELDIN, 2013). O RMSD é uma métrica de erro que quantifica a média das diferenças quadráticas entre os valores observados e os valores previstos por um modelo. Para aplicação em pauta, optou-se por avaliar o desvio em corrente elétrica, isto é, calculado por:

$$\text{RMSD} = \sqrt{\frac{1}{N_{curva}} \sum_{i=1}^{N_{curva}} (I_{curva} - I_{modelo})^2}. \quad (15)$$

De maneira semelhante, também é válido normalizar o valor deste desvio, dando origem ao Desvio Médio Quadrático Normalizado (NRMSD, do inglês *Normalized Root Mean Square Deviation*)

$$\text{NRMSD} = \frac{\text{RMSD}}{I_{sc}} \times 100\%. \quad (16)$$

No entanto, para esta normalização, utiliza-se a I_{sc} como referência de normalização.

2.6 CONCLUSÃO

Neste capítulo, é apresentada uma fundamentação teórica que reforça a necessidade da compreensão das características de funcionamento dos módulos FV, destacando alguns pontos de avaliação da curva I-V. Além disso, são apresentados os modelos baseados em diodos e o modelo MGNL.

Destaca-se a influência da variação de alguns parâmetros dos modelos apresentados, como n , R_s , R_{sh} , na formação da curva I-V, assim como as relações desses parâmetros com variações de irradiância e temperatura.

Assim, ressalta-se a importância dos modelos para a modelagem de módulos FV, bem como a escolha adequada de critérios que podem ser utilizados para avaliação da acurácia desses modelos, sendo sugeridos o EMAPN e o NRMSD, por apresentarem boa aderência às características desejadas pelo modelo.

3 DESAFIOS NA MODELAGEM DE SISTEMAS FOTOVOLTAICOS COM BLOCO PV ARRAY DO MATLAB

O MATLAB/Simulink é uma ferramenta amplamente utilizada no campo da engenharia e ciências aplicadas. O MATLAB, um software com uma linguagem de programação de alto desempenho, oferece uma ampla gama de funcionalidades para análise numérica, visualização de dados e desenvolvimento de algoritmos. Por outro lado, o Simulink é uma extensão do MATLAB que permite a modelagem, simulação e análise de sistemas dinâmicos por meio de diagramas de blocos, proporcionando uma abordagem visual para sistemas de engenharia (MATHWORKS, 2024). Esta ferramenta é particularmente útil na modelagem de sistemas FV devido à sua capacidade de lidar com a complexidade e variabilidade dos dados.

No entanto, a modelagem de sistemas FV apresenta diversos desafios. A variabilidade da irradiância solar e da temperatura ambiental, bem como a complexidade intrínseca dos modelos físicos e matemáticos, são alguns dos principais obstáculos. Entre os blocos presentes no Simulink, a partir da versão do MATLAB R2015b, destaca-se o bloco *PV Array*, que pode ser utilizado na modelagem de sistemas de geração solar FV. Este bloco representa módulos FV e oferece uma interface gráfica para inserir parâmetros definidos pelo usuário.

Este capítulo tem como objetivo explorar e discutir as limitações encontradas na modelagem de sistemas FV utilizando o bloco *PV Array* do MATLAB/Simulink. Serão abordadas as principais restrições e desafios, bem como possíveis soluções para mitigar essas limitações e melhorar a precisão dos modelos de simulação.

No desenvolvimento de pesquisas sobre caracterização, modelagem e detecção de degradações em potência em módulos FV, é comum utilizar este bloco específico como referência. Em essência, parte-se do pressuposto de que o comportamento desse bloco representa o comportamento esperado para o sistema em análise, seja uma célula, um módulo ou um arranjo de módulos. Portanto, é essencial investigar o funcionamento e o comportamento desse bloco, a fim de determinar sua adequação como referência e compreender quaisquer limitações que possa apresentar.

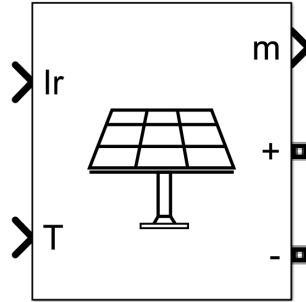
Um exemplo pode ser observado em (HARROU et al., 2018), que propõe uma avaliação das diferenças entre os MPP (I_{mpp} , V_{mpp} e P_{mpp}) do arranjo real e do modelo simulado, para assim determinar a existência ou não de degradações em potência. Porém, é fácil perceber que, caso o modelo utilizado não represente de maneira fidedigna o comportamento do arranjo saudável, pode-se apresentar vícios na detecção das degradações em potência.

3.1 BLOCO *PV ARRAY* DO MATLAB

Inicialmente, entende-se como relevante uma breve apresentação e contextualização do funcionamento do bloco *PV Array*, para assim estender a sua análise e compreender os detalhes de

sua operação. Dito isto, pode-se observar na Figura 15 como é disposto o bloco para ambiente de simulação. Desprende-se assim que este possui duas entradas, uma para irradiância (I_r , conforme bloco) e outra para temperatura (T , conforme bloco), o terminal positivo (+) e negativo (-), além de uma conexão para medição (m).

Figura 15 – Bloco *PV Array*.



Fonte: (MATHWORKS, 2024).

Isto posto, em consulta à central de ajuda do MATLAB (MATHWORKS, 2024), o bloco *PV Array* é um modelo baseado no SDM, e por isso possui cinco parâmetros, utilizando a corrente fotogerada (I_{ph}), resistência série (R_s) e resistência *shunt* (R_{sh}), fazendo com que a curva I-V seja dependente da irradiância (G) e da temperatura (T) do sistema especificado. Conforme já mencionado é baseado no SDM, cuja corrente, para um único módulo, pode ser correlacionada com a corrente de saturação (I_{sat}) e a tensão no diodo (V_d), através de:

$$I_d = I_{sat} \left[e^{\left(\frac{V_d}{V_t}\right)} - 1 \right], \quad (17)$$

enquanto a tensão térmica (V_t), pode ser escrita da seguinte forma:

$$V_t = \frac{KT}{q} \cdot n \cdot N_s, \quad (18)$$

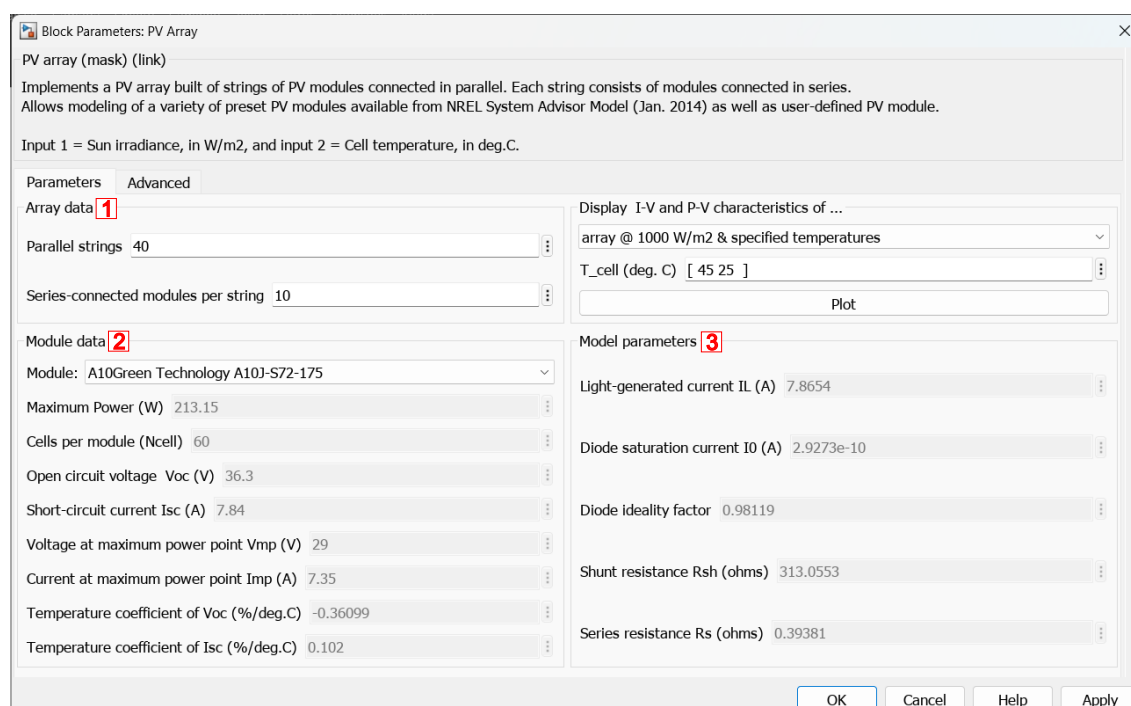
onde n é o fator de idealidade do diodo, N_s é o número de células em série, K é a constante de Boltzmann ($1,38 \cdot 10^{-23} J/K$), q é a carga de elétron ($1,602 \cdot 10^{-19} C$), e T é a temperatura em Kelvin.

É válido também avaliar quais parâmetros podem ser configurados no bloco *PV Array*, conforme ilustra a Figura 16. Na seção *Array data* [1] (dados do arranjo, em português), pode-se alterar a quantidade de *strings* (séries de módulos solares) em paralelo e de módulos conectados em série. Como exemplo, quando se deseja avaliar somente um módulo, deve-se preencher ambos os campos com o valor unitário. Enquanto isso, na seção *Module data* [2] (dados do módulo, em português) podem ser parametrizados os módulos. Neste ponto, indica-se pesquisar o modelo do módulo desejado no banco de dados, assim todos os demais parâmetros são determinados de maneira automática. Porém, caso não exista o módulo desejado dentre os disponíveis, deve-se utilizar o *user-defined* (definido pelo usuário, em português). Se for o caso, deve-se assim preencher os valores de:

- Células por módulo;
- Tensão de circuito aberto - V_{oc} (V);
- Corrente de curto circuito - I_{sc} (A);
- Tensão no ponto de máxima potência - V_{mp} (V);
- Corrente no ponto de máxima potência - I_{mp} (A);
- Coeficiente de temperatura do V_{oc} - ($\%/^{\circ}C$);
- Coeficiente de temperatura do I_{sc} - ($\%/^{\circ}C$).

Vale-se aqui salientar que, todos os dados a serem preenchidos devem ser em STC.

Figura 16 – Parâmetros Bloco *PV Array* no MATLAB/Simulink R2023a.



Fonte: (MATHWORKS, 2024).

Desta forma, a partir da definição dos dados do módulo, a seção *Model parameters* ³ (parâmetros do modelo, em português) é calculada e preenchida. Ressalta-se a necessidade da escolha adequada dos dados do módulo, sejam esses dados oriundos do *datasheet* ou do próprio banco de dados do MATLAB.

Em resumo, o bloco *PV Array* do MATLAB/Simulink é uma excelente ferramenta para a modelagem e simulação de sistemas FV, permitindo uma análise detalhada das características e desempenho dos módulos. Apesar disso, é um bloco que foi criado na versão MATLAB R2015b, e ao longo de suas versões, sofreu poucas alterações significativas em suas características construtivas. Suas atualizações se restringiram à manutenção do banco de dados e melhorias de interface com o usuário.

3.2 MÉTODO DE TRANSLAÇÃO DE CURVAS

Deve-se mencionar a metodologia utilizada para translação das curvas I-V, na qual o bloco *PV Array* aplica cinco equações para relacionar os parâmetros do modelo com irradiância e temperatura, sendo elas:

$$I_{ph} = \frac{G}{G_{ref}} [I_{ph,ref} + \alpha_{I_{sc}} (T - T_{ref})], \quad (19)$$

$$n = n_{ref} \left(\frac{T}{T_{ref}} \right), \quad (20)$$

$$R_{sh} = R_{sh,ref} \left(\frac{G_{ref}}{G} \right), \quad (21)$$

$$R_s = R_{s,ref}, \quad (22)$$

$$I_{sat} = I_{sat,ref} \left(\frac{T}{T_{ref}} \right)^3 e^{\frac{q}{k} \left(\frac{E_{g,ref}}{T_{ref}} - \frac{E_g}{T} \right)}, \quad (23)$$

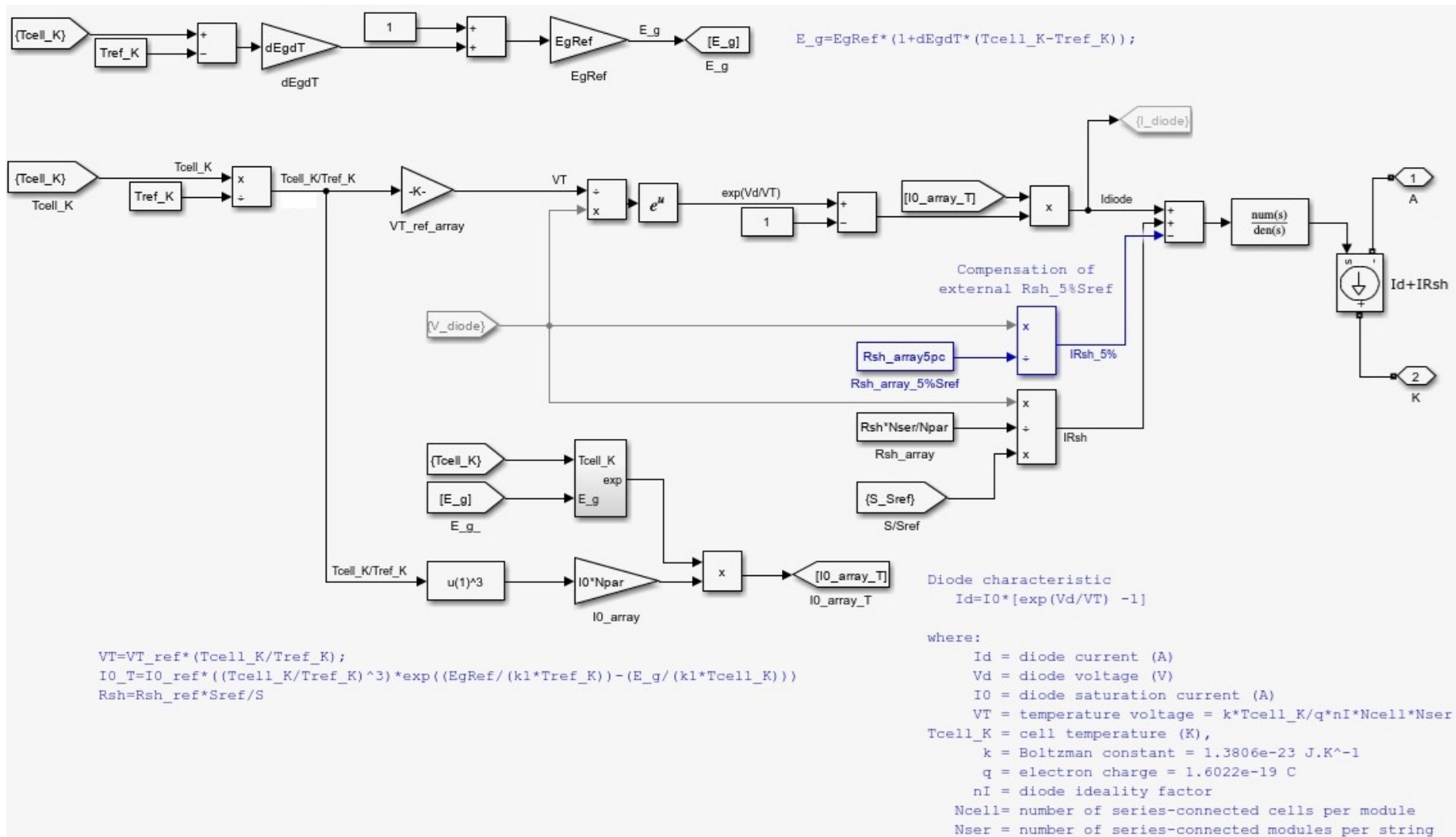
além das equações de translação de I_{sc} e V_{oc}

$$I_{sc} = I_{sc,ref} [1 + \alpha_{I_{sc}} (T - T_{ref})], \quad (24)$$

$$V_{oc} = V_{oc,ref} [1 + \beta_{V_{oc}} (T - T_{ref})], \quad (25)$$

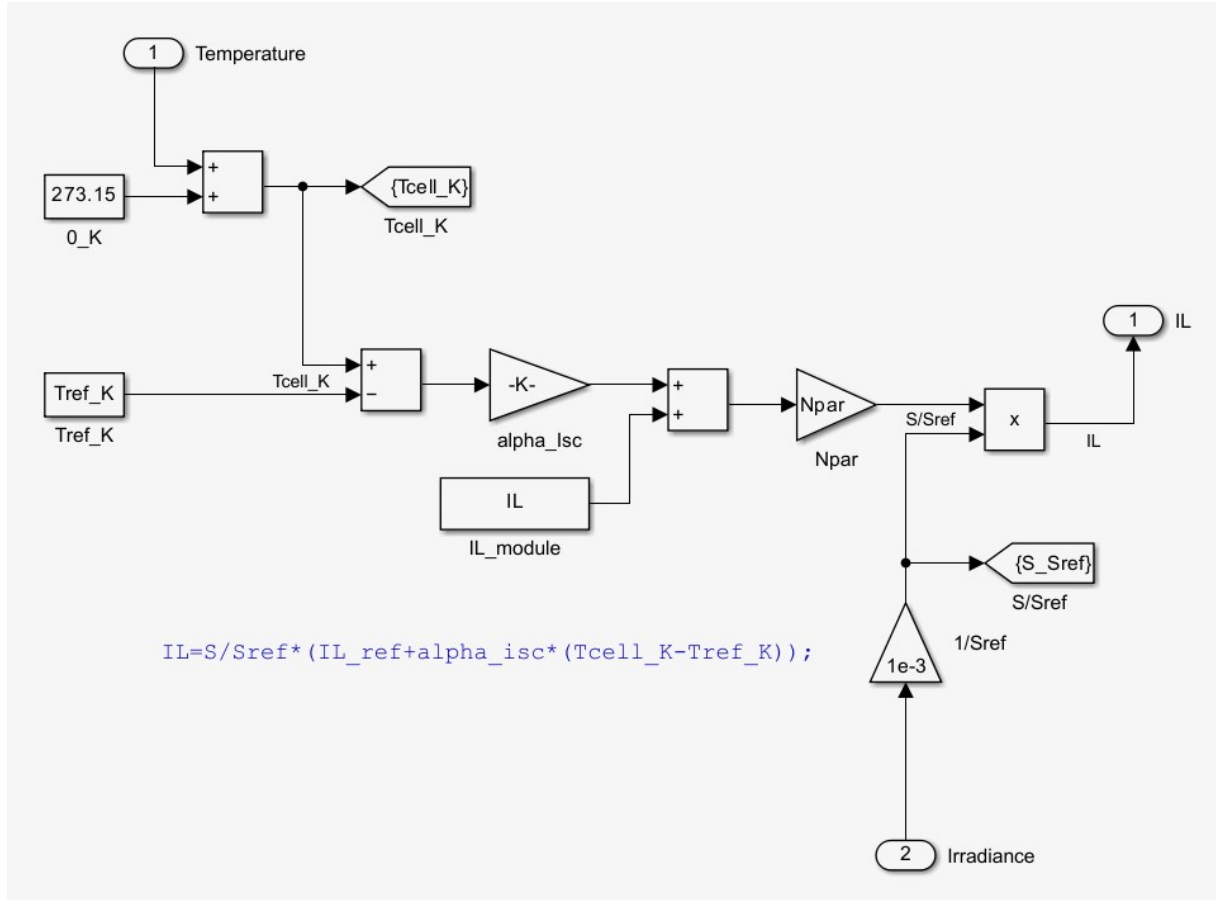
onde o subscrito *ref* é atribuído para valores de referência. Estas equações foram compiladas por (de SOTO et al., 2006) e amplamente aplicadas na modelagem de módulos FV, podendo-se mencionar alguns exemplos (CHOUDER et al., 2012), (HUANG et al., 2020), (WANG et al., 2021), (ZHANG et al., 2022a), entre outros. Conforme pode ser observado nas Figuras 17 e 18.

Figura 17 – Parâmetros internos do Bloco *PV Array* no MATLAB/Simulink R2023a.



Fonte: (MATHWORKS, 2024).

Figura 18 – Parâmetros internos do Bloco *PV Array* no MATLAB/Simulink R2023a.



Fonte: (MATHWORKS, 2024).

Neste ponto, é importante mencionar que na literatura científica é possível encontrar dezenas de procedimentos distintos de translação de curva I-V (PILIOUGINE et al., 2024), ressaltando que a norma IEC 60891:2021 descreve quatro procedimentos distintos, alguns deles com versões simplificadas (IEC, 2021).

Destaca-se que todos os métodos têm como objetivo principal corrigir os parâmetros dos módulos FV de STC para outras condições de operação. No entanto, possuem complexidades e abordagens distintas. O método de (de SOTO et al., 2006) é conhecido por sua simplicidade e aplicação direta de equações específicas para cada parâmetro.

Além disso, enquanto o método de (de SOTO et al., 2006) é eficaz para pequenas correções de irradiância, os procedimentos da IEC 60891:2021 fornecem um conjunto padronizado de técnicas amplamente utilizado na indústria, capaz de lidar com uma variedade mais ampla de condições e ajustes de parâmetros.

Assim, aproveita-se para aprofundar a análise acerca destas equações aqui apresentadas. É recorrente encontrar na literatura outras abordagens para as variações dos parâmetros em função da temperatura e da irradiância. Como exemplo, pode-se avaliar a resistência série R_s , que de acordo com (22), é independente de irradiância e temperatura.

No entanto, em (ZHANG et al., 2022b), o autor apresenta uma possível dependência de R_s , conforme

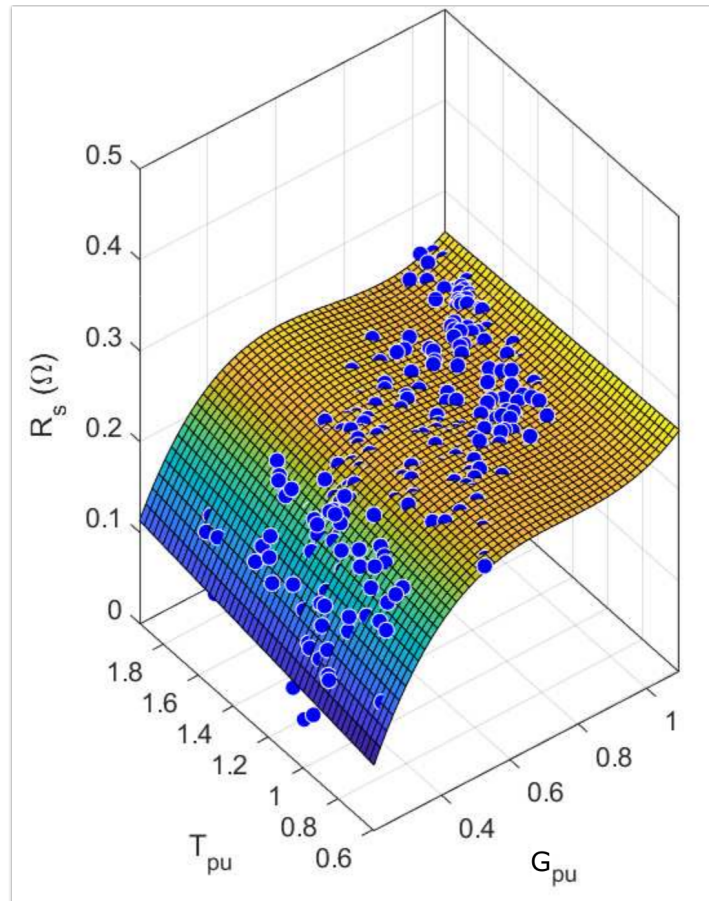
$$R_s = R_{s,ref} + \varepsilon_{R_s}(T - T_{ref}) + \gamma_{R_s}(G - G_{ref}), \quad (26)$$

onde ε_{R_s} é o coeficiente de temperatura de R_s e γ_{R_s} é o coeficiente de irradiância. Por outro lado, (WANG et al., 2021) utiliza a correlação

$$R_s = R_{s,ref} - \left[\frac{n}{I_{sat}} e^{\left(\frac{-V_{oc}}{n}\right)} \right]. \quad (27)$$

Não obstante, entende-se que os resultados obtidos em (JUNIOR, 2023) corroboram com a ideia de que R_s é dependente de irradiância e temperatura. O autor propõe a utilização de um polinômio para avaliar o comportamento dos parâmetros do SDM. Assim sendo, para exemplificar, pode-se apresentar a Figura 19, em que é possível perceber a variação dos valores de R_s para diferentes valores de irradiância e temperatura. Deve-se explicar que a Figura 19, apresentada por (JUNIOR, 2023), utiliza a notação por unidade (pu), em que os valores base são T e G em STC; em outras palavras, $T_{pu} = 1$ representa $25^\circ C$ e $G_{pu} = 1$ representa $1000 W/m$. Salienta-se que em (JUNIOR, 2023) os valores são obtidos através dos bancos de dados da NREL.

Figura 19 – Plano de R_s para módulos de silício.



Fonte: Adaptado de (JUNIOR, 2023).

3.3 ENERGIA DE BANDA PROIBIDA (E_g)

Pode-se retomar a corrente de saturação, Equação (23), em que a energia de banda proibida E_g apresenta uma pequena dependência com a temperatura, sendo representada no Bloco *PV Array*, conforme

$$E_g = E_{g,ref}(1 - 0,0002677[T - T_{ref}]). \quad (28)$$

Esta equação também fora apresentada por (de SOTO et al., 2006) e também é recorrente encontrá-la em métodos de modelagem de módulos FV,

Em (JUNIOR, 2018) opta-se por utilizar a equação, conhecida como Efeito Varshni (VARSHNI, 1967),

$$E_g(T) = E_g(0) - \frac{\alpha_{Eg}T^2}{T + \beta_{Eg}}, \quad (29)$$

em que $E_g(0)$ é a energia em 0 K ($-273,15^\circ\text{C}$) e α_{Eg} e β_{Eg} são constantes que dependem do material utilizado. Como exemplo, em (SINGH et al., 2012) são apresentados os valores da Tabela 2, com os parâmetros de $E_g(0)$, α_{Eg} e β_{Eg} . Além disso, acrescentou-se a $E_g(STC)$, que representa a energia de banda proibida em temperatura de STC, ou seja, 25°C , para os materiais semicondutores Silício (Si), Arseneto de gálio (GaAs), Telureto de Cádmio (CdTe) e Sulfeto de Cádmio (CdS).

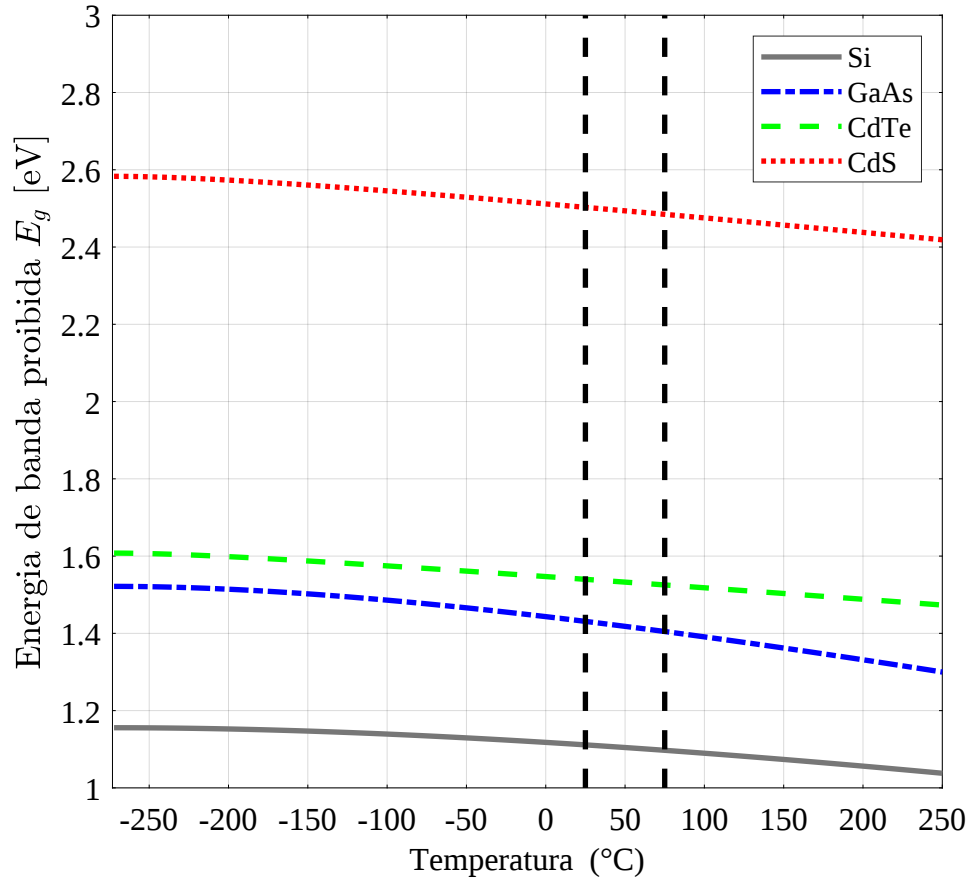
Tabela 2 – Parâmetros da energia de banda proibida para materiais semicondutores Si, GaAs, CdTe e CdS.

Material	$E_g(0)$ [eV]	α_{Eg} [eV/K] $\cdot 10^{-4}$	β_{Eg} [K]	$E_g(STC)$ [eV]
Si	1,1557	7,021	1108	1,1113
GaAs	1,5216	8,871	572	1,4310
CdTe	1,6077	3,100	108	1,5399
CdS	2,5830	4,020	147	2,5027

Fonte: Adaptado de (SINGH et al., 2012).

Assim sendo, utilizando os valores da Tabela 2 em (29), pode-se representar, conforme a Figura 20, os valores da energia de banda proibida em função da temperatura. Inicialmente, optou-se por avaliar uma variação de temperatura mais ampla, entre $-273,15^\circ\text{C}$, para contemplar o valor de $E_g(0)$, e 250°C , por considerar aplicações tanto terrestres quanto espaciais de módulos FV (SINGH et al., 2012). No entanto, também se destaca na referida figura a faixa de temperatura entre 25°C e 75°C , por serem temperaturas mais usuais.

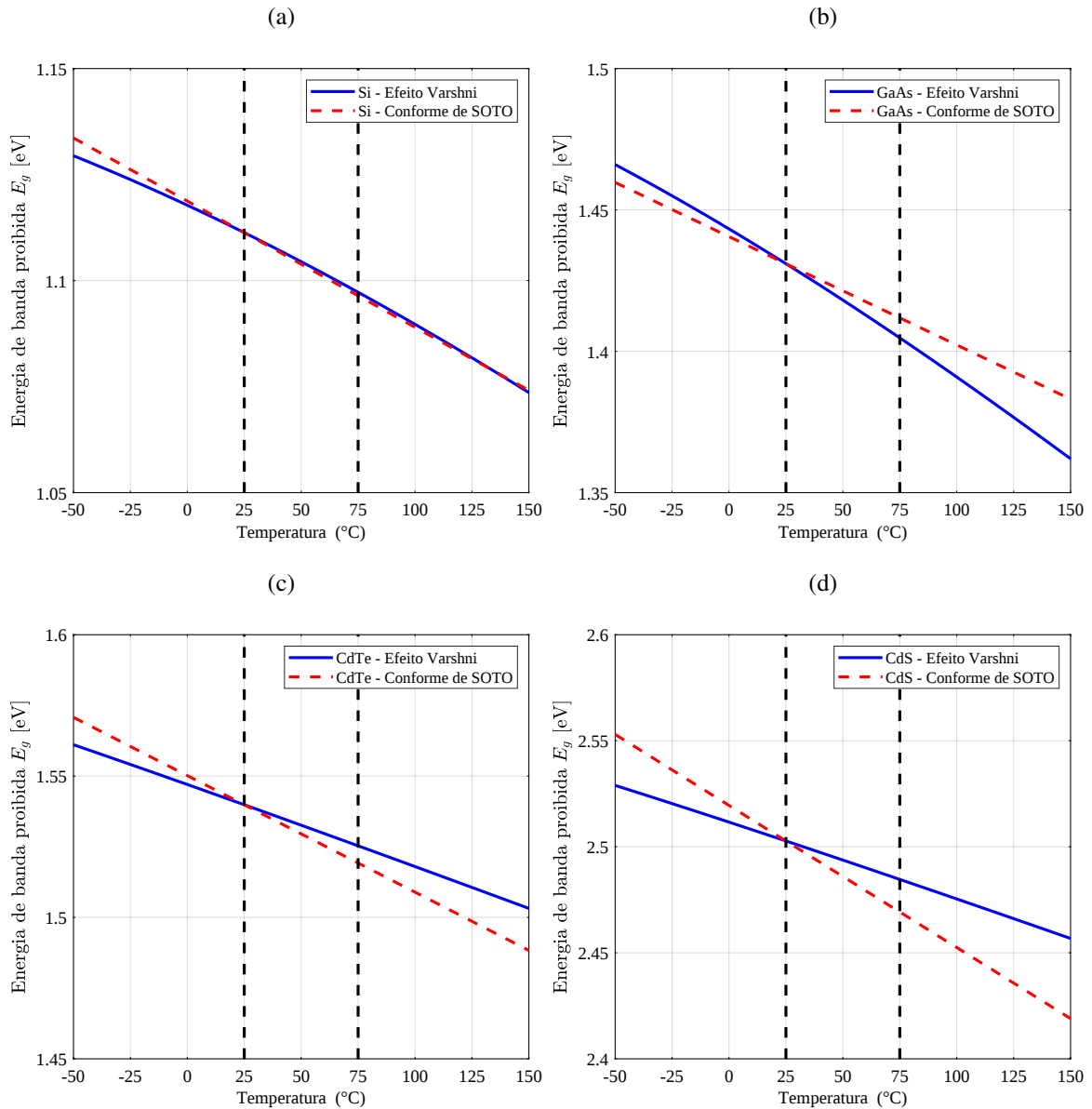
Figura 20 – Comportamento da energia de banda proibida de acordo com o Efeito Varshni para materiais semicondutores Si, GaAs, CdTe e CdS em função da variação de temperatura.



Fonte: O Autor (2024).

Com o intuito de comparar a simplificação apresentada em (de SOTO et al., 2006), Equação (28), com o Efeito Varshni (VARSHNI, 1967), Equação (29), considerou-se que a energia de banda proibida de referência ($E_{g,ref}$) de (28) é igual ao valor de $E_g(STC)$ apresentado na Tabela 2, resultando assim na Figura 21.

Figura 21 – Energia de banda proibida comparando Efeito Varshni em (29) e a linearização apresentada por (de SOTO et al., 2006) em (28) para materiais semicondutores (a) Si, (b) GaAs, (c) CdTe e (d) CdS em função da variação de temperatura.



Fonte: O Autor (2024).

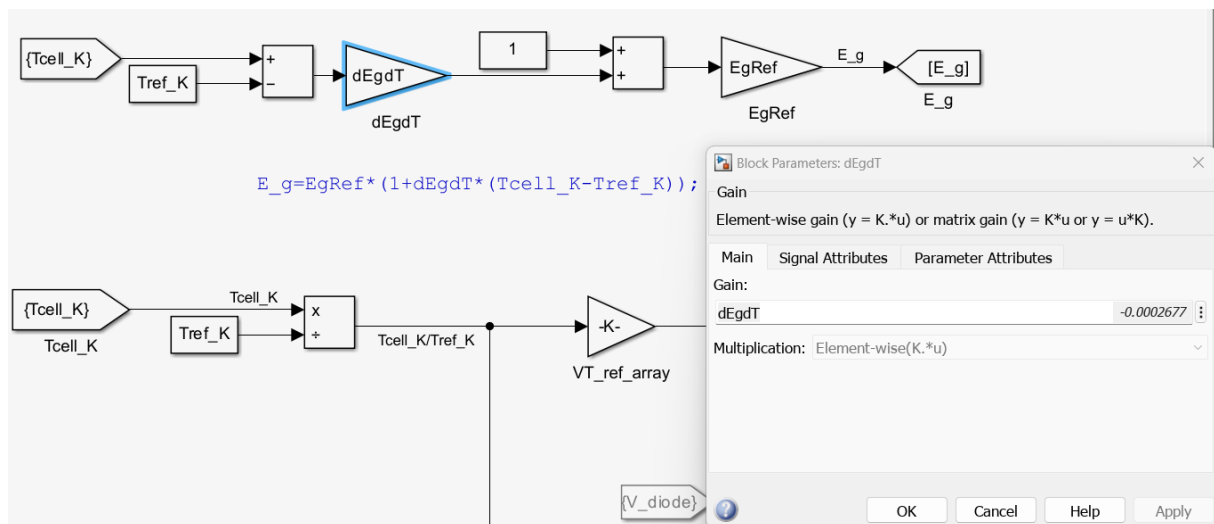
Nota-se assim que a linearização apresentada em (de SOTO et al., 2006) pode ser considerada adequada para módulos de Si, enquanto para módulos de outras tecnologias não é tão aderente, podendo assim representar uma limitação de utilização do Bloco *PV Array* para modelar módulos e sistemas que possuam uma tecnologia diferente de Silício.

Como exemplo, pode-se observar que o módulo FS-4122A-3 do fabricante First Solar, presente no banco de dados do Bloco *PV Array*, é um módulo concebido com células de CdTe, conforme ilustra o seu *datasheet* na Figura 22. Pode-se também apresentar a captura da tela do Simulink, conforme Figura 23, confirmando assim a utilização do valor de $-0,0002677$, em (28). Nota-se, assim, que a energia de banda proibida é regida por (28), sem qualquer adaptação, corroborando assim para as limitações apresentadas pela utilização do Bloco *PV Array*.

Figura 22 – Datasheet do módulo First Solar - FS-4122A-3.

MECHANICAL DESCRIPTION		MODULE NUMBERS AND RATINGS AT STANDARD TEST CONDITIONS (1000W/m², AM 1.5, 25°C) ⁵								
Length	1200mm	NOMINAL VALUES		FS-4110-3 FS-4110A-3	FS-4112-3 FS-4112A-3	FS-4115-3 FS-4115A-3	FS-4117-3 FS-4117A-3	FS-4120-3 FS-4120A-3	FS-4122-3 FS-4122A-3	
Width	600mm									
Weight	12kg	Nominal Power ⁶ (-0/+5W)	P _{MPP} (W)	110.0	112.5	115.0	117.5	120.0	122.5	
Thickness	6.8mm	Voltage at P _{MAX}	V _{MPP} (V)	67.8	68.5	69.3	70.1	70.8	71.5	
Area	0.72m²	Current at P _{MAX}	I _{MPP} (A)	1.62	1.64	1.66	1.68	1.70	1.71	
Individual Leadwire	2.5mm², 657mm (minimum from strain relief to connector mating surface)	Open Circuit Voltage	V _{OC} (V)	86.4	87.0	87.6	88.1	88.7	88.7	
		Short Circuit Current	I _{SC} (A)	1.82	1.83	1.83	1.83	1.84	1.85	
		Module Efficiency	%	15.3	15.6	16.0	16.3	16.7	17.0	
Connectors	MC4 or MC4-EVO 2 ⁹	Maximum System Voltage	V _{SYS} (V)	1500 ^{7,8}						
Bypass Diode	None	Limiting Reverse Current	I _R (A)	4.0						
Cell Type	Thin-film CdTe semiconductor, up to 216 cells	Maximum Series Fuse	I _{CF} (A)	4.0						
RATINGS AT NOMINAL OPERATING CELL TEMPERATURE OF 45°C (800W/m², 20°C air temperature, AM 1.5, 1m/s wind speed) ⁵										
Frame Material	None	Nominal Power	P _{MPP} (W)	83.2	85.1	87.0	89.0	90.8	92.7	
Front Glass	3.2mm heat strengthened	Voltage at P _{MAX}	V _{MPP} (V)	63.5	64.5	64.9	65.9	66.3	67.2	
	Series 4A™ Includes anti-reflective coating	Current at P _{MAX}	I _{MPP} (A)	1.31	1.32	1.34	1.35	1.37	1.38	
		Open Circuit Voltage	V _{OC} (V)	81.6	82.1	82.7	83.2	83.7	83.7	
		Short Circuit Current	I _{SC} (A)	1.47	1.47	1.48	1.48	1.48	1.49	
Back Glass	3.2mm tempered	TEMPERATURE CHARACTERISTICS								
Encapsulation	Laminate material with edge seal	Module Operating Temperature Range	(°C)	-40 to +85						
Load Rating	2400Pa ¹⁰	Temperature Coefficient of P _{MPP}	T _K (P _{MPP})	-0.28%/°C [Temperature Range: 25°C to 75°C]						
		Temperature Coefficient of V _{OC}	T _K (V _{OC})	-0.28%/°C						
		Temperature Coefficient of I _{SC}	T _K (I _{SC})	+0.04%/°C						

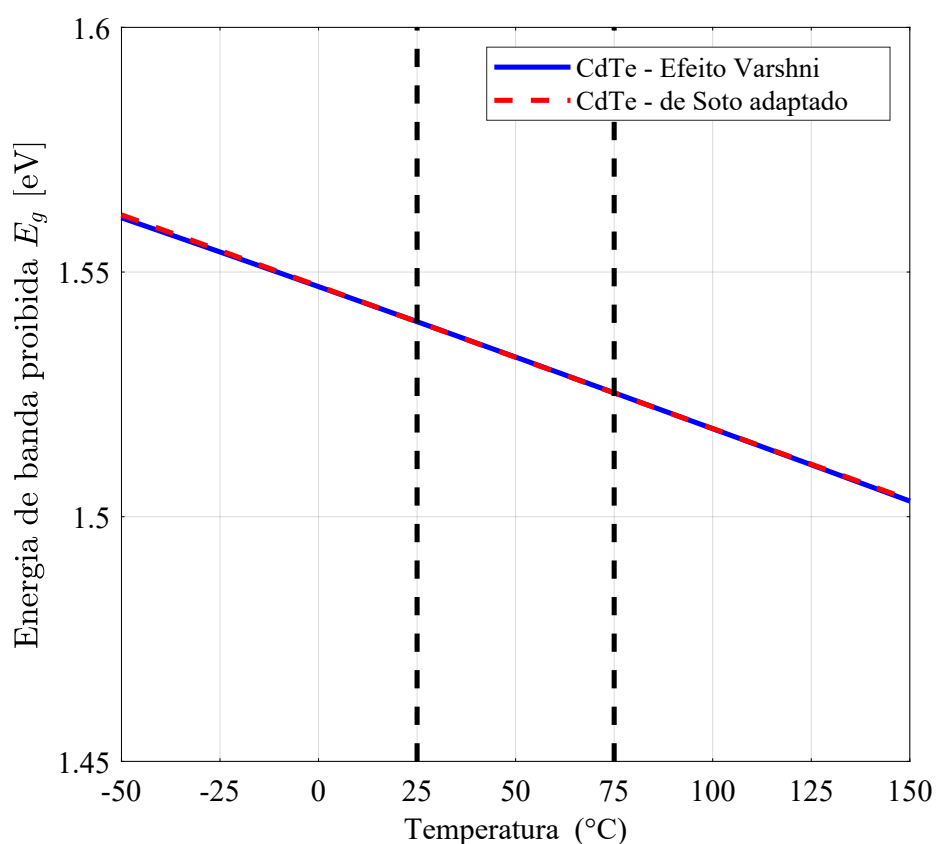
Fonte: (First Solar, 2018).

Figura 23 – Implementação da energia de banda proibida no bloco *PV Array* para o módulo FS-4122A-3.

Fonte: (MATHWORKS, 2024).

Cabe aqui esclarecer que é factível alterar o valor desta variável. Como exemplo, através da análise da curva apresentada na Figura 24, compreende-se que o valor de $-0,00018904$ seria mais aderente aos módulos de CdTe. Portanto, fazendo a correção mencionada, é obtida a adaptação apresentada na Figura 24.

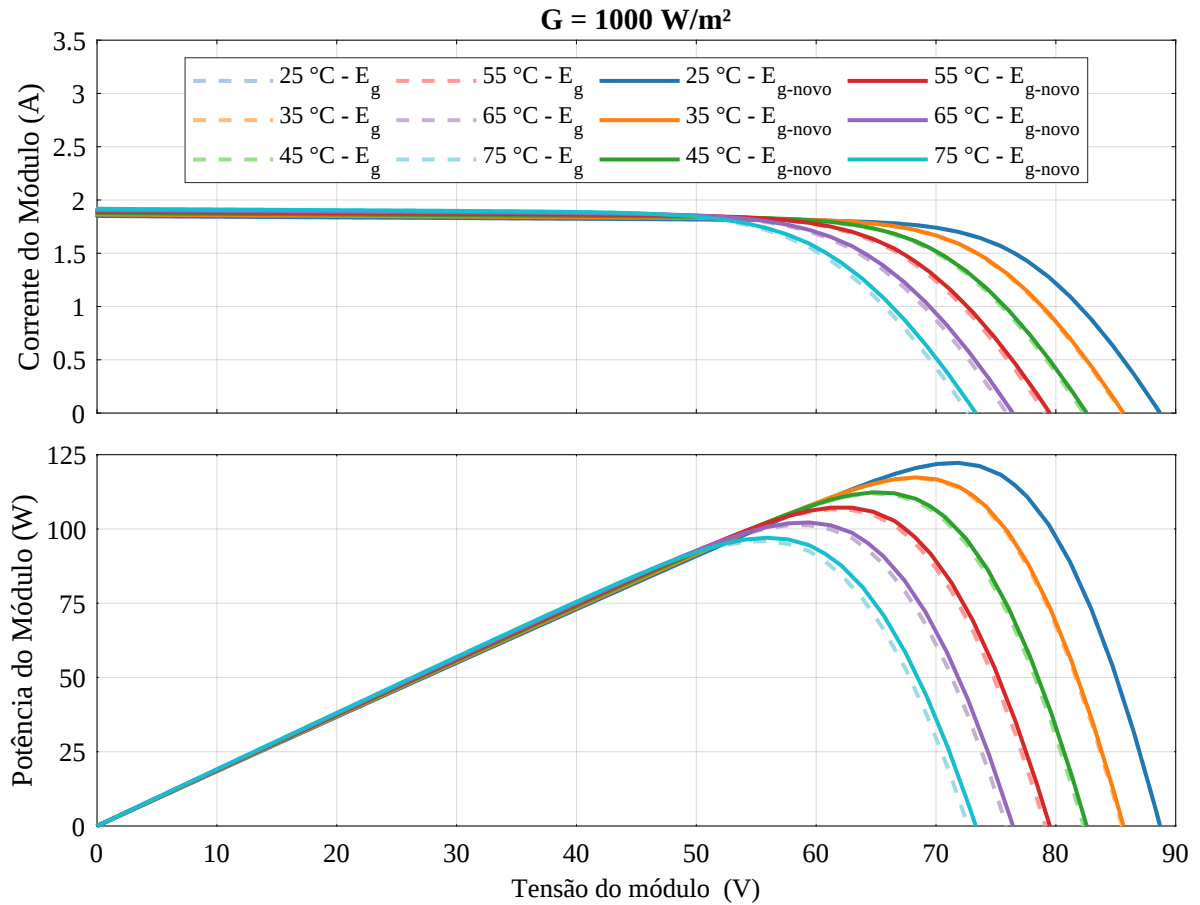
Figura 24 – Energia de banda proibida comparando Efeito Varshni em (29) e a adaptação da linearização apresentada por (de SOTO et al., 2006) em (28) para módulos de CdTe.



Fonte: o Autor (2024).

Isto posto, pode-se simular as curvas I-V com a correção deste valor e sem a alteração deste valor, salientando aqui que a alteração é mais visível quando há variações na temperatura do módulo. Assim, com a análise da Figura 25, percebe-se o descasamento entre as curvas I-V à medida que a temperatura vai sendo maior que 25 °C.

Figura 25 – Comparativo das curvas I-V e P-V do módulo FS-4122A-3 simulado usando o bloco *P-V array* sem considerar a correção da variação de E_g e considerando esta correção (E_{g-novo}).



Fonte: O Autor (2024).

3.4 A ESCOLHA DOS PARÂMETROS DE REFERÊNCIA

É interessante discutir os parâmetros apresentados nos *datasheets* dos fabricantes de módulos FV. Infelizmente, alguns fabricantes não apresentam as informações dos módulos de maneira assertiva, ou até mesmo conflitante dentre informações do próprio *datasheet*. Além disso, eventualmente, devido a problemas no processo produtivo, pode-se ter alguma discrepância entre os dados do *datasheet* e o efetivamente apresentados pelo módulo.

Como exemplo, pode-se observar as informações apresentadas no *datasheet* do módulo RSM-100P (RESUN, 2024), conforme apresentado na Figura 26. De acordo com a planilha das condições de funcionamento em STC, a I_{sc} é 6,04 A (Figura 26(a)). No entanto, observando a curva I-V apresentada no mesmo *datasheet* (Figura 26(b)), percebe-se que para todas as irradiâncias, inclusive 1000 W/m^2 , a corrente não atinge valores superiores a 6 A. Enquanto observando a Figura 26(c), os valores parecem se aproximar de 6,04 A.

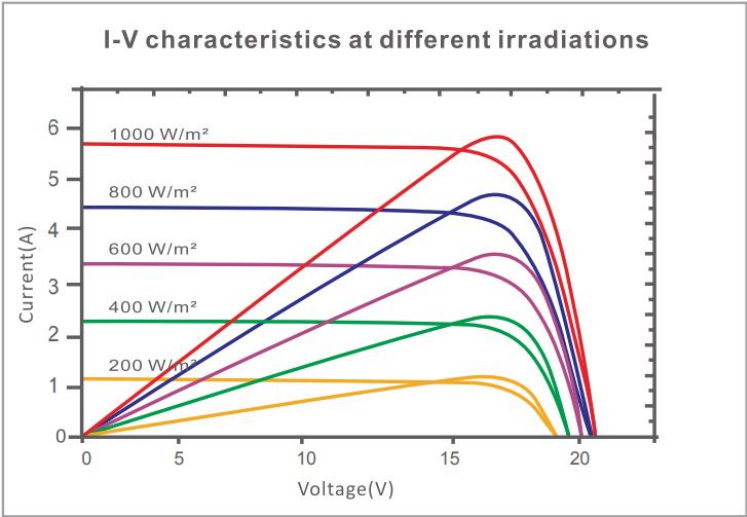
Figura 26 – Informações do *datasheet* do módulo RSM-100P do fabricante Resun. (a) Dados elétricos em STC. (b) Curva I-V para diferentes irradiâncias. (c) Curva I-V para diferentes temperaturas.

(a)

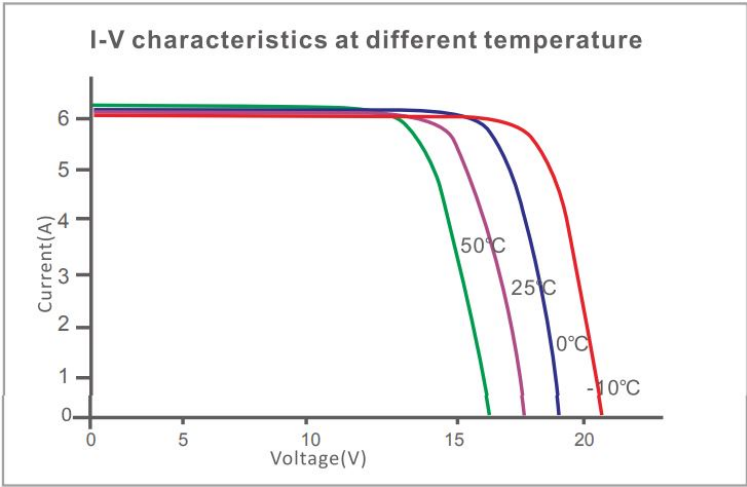
ELECTRICAL DATA(STC)	
Rated Power in Watts-Pmax(Wp)	100W
Open Circuit Voltage-Voc(V)	21.58V
Short Circuit Current-Isc(A)	6.04A
Maximum Power Voltage-Vmp(V)	17.40V
Maximum Power Current-Imp(A)	5.75A
Module Efficiency (%)	15.44%

STC: Irradiance 1000 W/m², Cell Temperature 25°C, Air Mass AM1.5 according to EN 60904-3.

(b)



(c)



Fonte: (RESUN, 2024).

Não obstante, outro ponto passível de análise é a aderência das informações apresentadas no *datasheet* com o banco de dados do módulo do MATLAB. Para ilustrar, pode-se avaliar os dados apresentados no *datasheet* do módulo YL265C-30b da YINGLI Solar, conforme Figura 27, cabendo aqui a constatação de que essa é a versão americana do *datasheet* e que a versão alemã, Figura 28, do mesmo módulo apresenta valores distintos.

Figura 27 – *Datasheet* do Módulo YINGLI - YL265C-30b - Americano.

ELECTRICAL PERFORMANCE

Electrical parameters at Standard Test Conditions (STC)

Module type			YLxxxC-30b (xxx=P _{max})				
Power output	P _{max}	W	280	275	270	265	260
Power output tolerances	ΔP _{max}	W	0 / + 5				
Module efficiency	η _m	%	17.1	16.8	16.5	16.2	15.9
Voltage at P _{max}	V _{mpp}	V	31.3	30.9	30.5	30.1	29.7
Current at P _{max}	I _{mpp}	A	8.96	8.91	8.85	8.79	8.74
Open-circuit voltage	V _{oc}	V	39.1	38.8	38.6	38.3	38.1
Short-circuit current	I _{sc}	A	9.50	9.47	9.43	9.37	9.35

STC: 1000W/m² irradiance, 25°C cell temperature, AM1.5g spectrum according to EN 60904-3.
Average relative efficiency reduction of 3.5% at 200W/m² according to EN 60904-1.

Fonte: (YINGLI, 2016).

Figura 28 – *Datasheet* do Módulo YINGLI - YL265C-30b - Alemão.

ELEKTRISCHES VERHALTEN

Elektrische Parameter bei Standard-Testbedingungen (STC)

Modultyp			YLxxxC-30b (xxx=P _{max})						
Leistung	P _{max}	W	280	275	270	265	260	255	250
Leistungstoleranz	ΔP _{max}	W	0 / +5						
Modulwirkungsgrad	η _m	%	17,1	16,8	16,5	16,2	15,9	15,6	15,3
Spannung bei P _{max}	V _{mpp}	V	31,1	30,8	31,1	31,0	30,8	30,6	30,5
Strom bei P _{max}	I _{mpp}	A	9,01	8,94	8,68	8,55	8,46	8,33	8,20
Leerlaufspannung	V _{oc}	V	38,9	38,6	39,0	39,0	38,6	38,3	38,1
Kurzschlussstrom	I _{sc}	A	9,61	9,55	9,06	8,93	8,91	8,85	8,71

STC: 1000W/m² Einstrahlung, 25°C T_{zeller}, AM 1,5 Spektrum gemäß EN 60904-3.
Mittlere Verringerung des relativen Wirkungsgrads von 3,5% bei 200W/m² gemäß EN 60904-1.

Fonte: (YINGLI, 2016).

Ambas as versões são ligeiramente diferentes das informações do banco de dados do MATLAB, conforme a Figura 29. Com o intuito de facilitar a avaliação, os dados são apresentados na Tabela 3.

Figura 29 – Parâmetros do MATLAB para o módulo YL265C-30b.

Module data

Module: Yingli Energy (China) YL265C-30b

Maximum Power (W) 265.2174

Cells per module (Ncell) 60

Open circuit voltage Voc (V) 38.28

Short-circuit current Isc (A) 9.35

Voltage at maximum power point Vmp (V) 30.38

Current at maximum power point Imp (A) 8.73

Temperature coefficient of Voc (%/deg.C) -0.3

Temperature coefficient of Isc (%/deg.C) 0.044

Fonte: (MATHWORKS, 2024).

Tabela 3 – Comparação entre dados do módulo YL265C-30b de diferentes Fontes.

Modelo	YL265C-30b	YL265C-30b	YL265C-30b
Fonte dos dados	<i>Datasheet</i> Americano	<i>Datasheet</i> Alemão	MATLAB
Nº de Celulas	60	60	60
$P_{\max}$ (W)	265	265	265,2174
V_{oc} (V)	38,3	39	38,28
I_{sc} (A)	9,37	8,93	9,35
V_{mp} (V)	30,1	31	30,38
I_{mp} (A)	8,79	8,55	8,73
α_{Isc} (A)	-*	-*	-0,3
β_{Voc} (A)	-*	-*	0,044

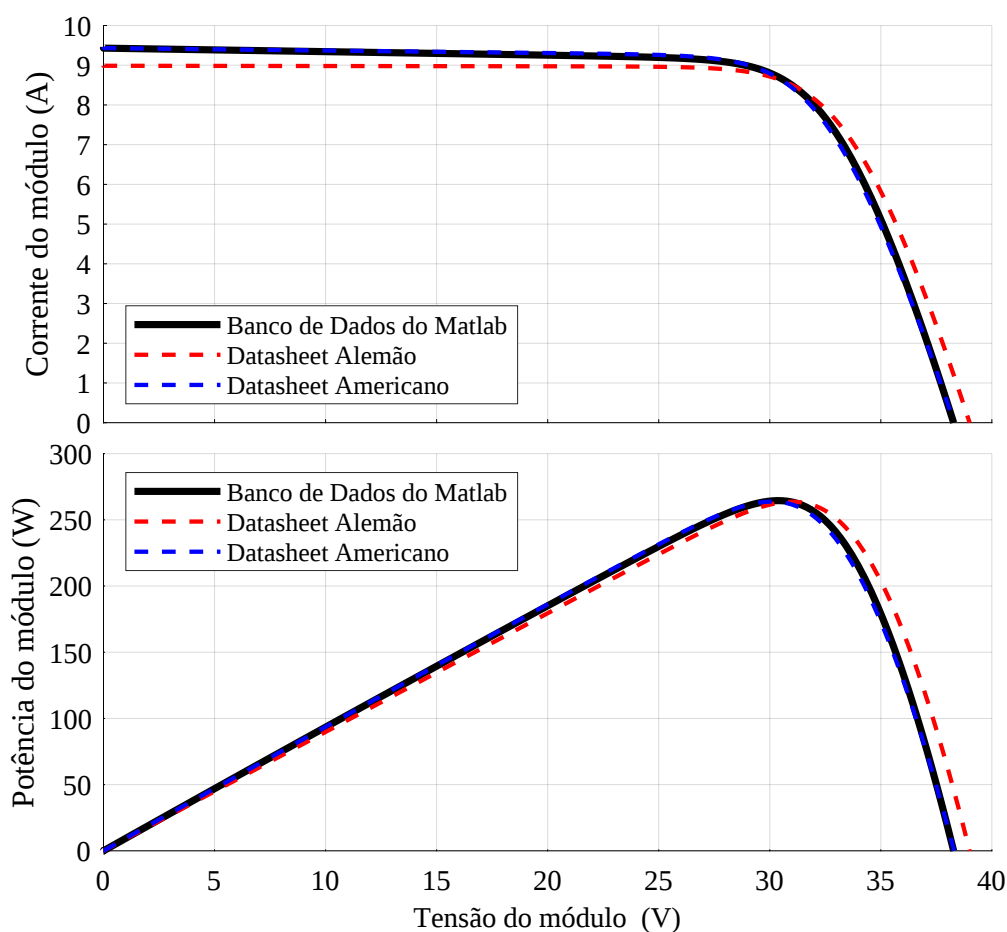
* - valores não apresentados nos *Datasheet*.

Fonte: O Autor (2024).

Assim sendo, caso fossem utilizados os dados apresentados para parametrização do bloco *PV Array*, é possível que seja inserido um erro na origem dos dados. Para melhor avaliar essa informação, as curvas I-V e P-V do módulo YL265C-30b são apresentadas na Figura 30.

Pondera-se aqui que é evidente que para o caso apresentado o erro relativo é pequeno, porém, caso o objetivo da análise seja avaliar cenários de produção de grandes plantas de geração de energia, ajustar algum mecanismo de busca de MPP ou até mesmo testar algum algoritmo de detecção de degradações em potência em módulos, essa fonte de erro pode influenciar nos resultados.

Figura 30 – Comparação entre as curvas I-V e P-V do Módulo YL265C-30b para diferentes fontes de dados.



Fonte: O Autor (2024).

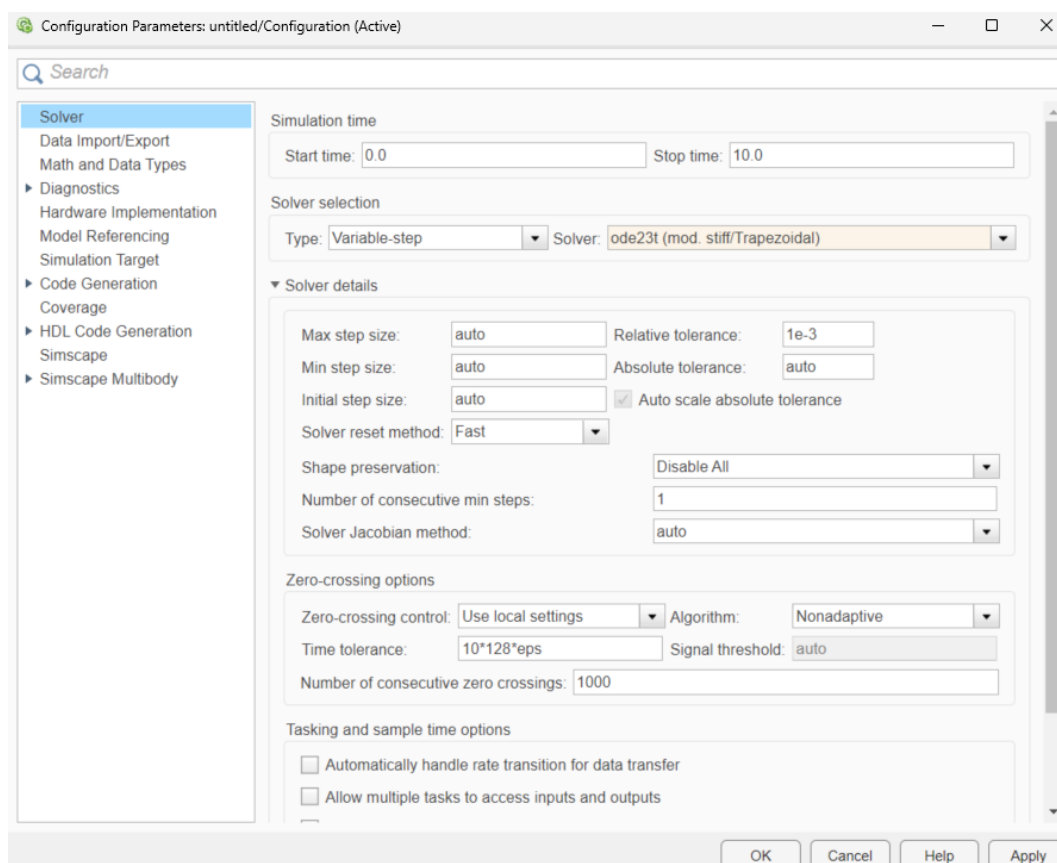
Entende-se que, apesar dos esforços da indústria, é facilmente observada alguma variação nas características elétricas de módulos FV que deveriam ser idênticas. Isso fica evidenciado principalmente pela incerteza associada à máxima potência, que em módulos comerciais é comum apresentar uma tolerância de algo em torno de 5%, fato este que pode ser observado em alguns *datasheets* comerciais (PAVAN et al., 2014).

3.5 OPERAÇÃO DO PV ARRAY DURANTE TRANSITÓRIOS

Cabe também apresentar que eventuais limitações de cálculo numérico podem ser um ponto sensível durante a utilização do bloco *PV Array*. Essas limitações podem decorrer do processo iterativo necessário para resolver as equações que descrevem o comportamento dos módulos. Em outras palavras, durante a simulação, o bloco *PV Array* precisa realizar cálculos numéricos para encontrar as grandezas elétricas resultantes em cada ponto de operação, com base nas condições de irradiância e temperatura fornecidas. No entanto, a precisão desses cálculos depende da definição adequada de tolerâncias para a convergência das soluções.

A tolerância no MATLAB/Simulink é um parâmetro que afeta a precisão das simulações, que define o erro máximo relativo permitido para os estados do sistema em cada passo de cálculo e é necessário para controlar a precisão do cálculo de passo de variável. O valor padrão da tolerância é 10^{-3} , como apresenta a Figura 31, indicando o valor padrão da tolerância e como alterá-lo.

Figura 31 – Configuração de Parâmetros do MATLAB/Simulink.

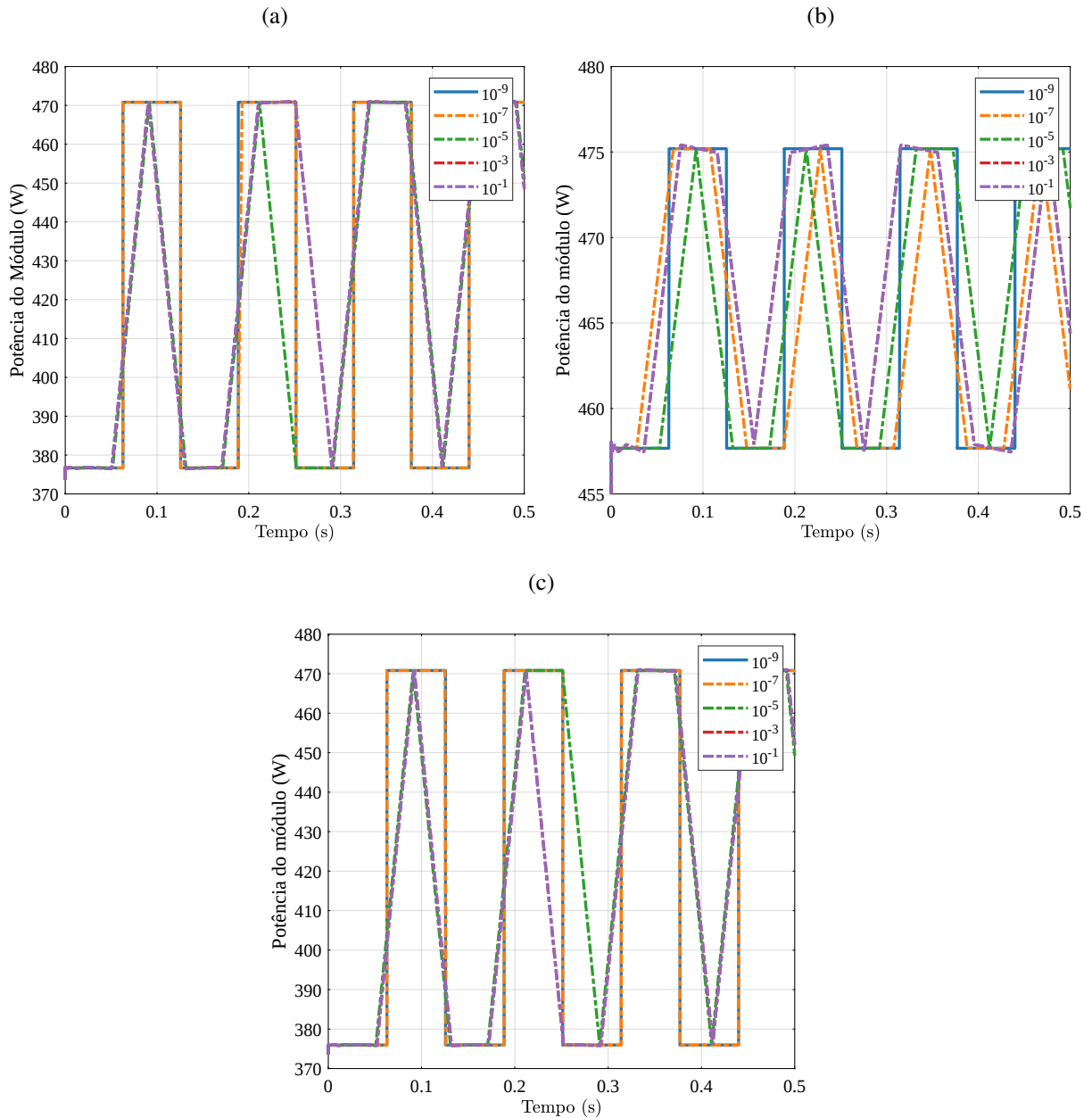


Fonte: (MATHWORKS, 2024).

Com o intuito de evidenciar um pouco melhor esta limitação, utilizou-se o bloco *PV Array* parametrizados com o Módulo Canadian Solar CS6U-360P de 360 W em diversas simulações com variações em degrau de irradiância, temperatura e tensão elétrica. Para a irradiância variou-se entre 1000 W/m^2 e 900 W/m^2 (queda de 10% da irradiância de STC), para a temperatura variou-se entre $25 \text{ }^\circ\text{C}$ e $75 \text{ }^\circ\text{C}$ (elevação de duas vezes da temperatura de STC), enquanto para a tensão elétrica variou-se entre 41,85 V e 46,5 V (queda de 10% V_{oc}), todas em uma frequência de 50 Hz. Os resultados são apresentados na Figura 32.

Observando os resultados apresentados na Figura 32, nota-se que, para variações de irradiância, tolerâncias superiores a 10^{-5} apresentam discrepâncias (Figura 32(a)), para variações de temperatura, tolerâncias superiores a 10^{-7} também apresentam desvios (Figura 32(b)), por fim, para variações em tensão elétrica a tolerâncias superiores a 10^{-5} apresentam erros (Figura 32(c)). Assim sendo, pode-se notar que valores de tolerância superiores 10^{-5} apresentam problemas de representação fiel do módulo durante transitórios de parâmetros.

Figura 32 – Avaliação do comportamento transitório da potência elétrica para diferentes variações em degrau dos parâmetros: (a) irradiância (b) temperatura (c) tensão.



Fonte: O Autor (2024).

Salienta-se aqui que o objetivo não é abordar aspectos numéricos ou quantitativos destes erros, somente avaliar que a depender da tolerância pode-se propagar erros, caso o objeto da avaliação seja intermitente ou dinâmico. Além disso, a depender da aplicação utilizada, estes erros podem ser significativos ou não.

Ressalta-se que ao definir as tolerâncias de convergência, é importante considerar o equilíbrio entre precisão e eficiência computacional. Se as tolerâncias forem definidas muito baixas, o algoritmo de cálculo pode exigir um número excessivo de iterações para alcançar a convergência, resultando em tempos de simulação mais longos e maior esforço computacional.

Por outro lado, tolerâncias muito altas podem levar a soluções aproximadas que não capturam com precisão o comportamento real dos módulos. Isso pode afetar diretamente a precisão das análises realizadas.

Neste ponto, também é válido mencionar que o bloco apresenta limitações quanto à alteração do passo de cálculo, caso se deseje utilizar o passo de cálculo fixo. Nos casos em que o passo configurado foi superior a $2,27 \mu s$, a execução da simulação foi interrompida por erros no cálculo numérico. Sendo assim, recomenda-se utilizar valores inferiores ao supracitado para o passo de cálculo.

3.6 CONCLUSÃO

Este capítulo aborda as limitações de modelagem em sistemas FV, com foco especial no bloco *PV Array* do MATLAB. A partir de uma análise detalhada, fica evidente que embora o bloco *PV Array* seja amplamente utilizado para modelar sistemas FV, ele apresenta limitações que podem afetar a precisão dos resultados. Por exemplo, a necessidade de escolher cuidadosamente os parâmetros a serem utilizados.

Além disso, a metodologia de translação de curvas utilizada pelo bloco *PV Array* também pode introduzir incertezas nos resultados. Outra questão importante discutida é a variação nas características elétricas dos módulos FV, mesmo quando são supostamente idênticos. Essas variações podem introduzir erros significativos na modelagem de sistemas FV, especialmente em cenários de produção em larga escala.

Em síntese, destaca-se a importância de uma abordagem cuidadosa ao modelar utilizando o bloco *PV Array* do MATLAB. Embora essa ferramenta seja poderosa, é essencial reconhecer e mitigar suas limitações para obter resultados precisos e confiáveis na análise de sistemas FV.

4 APLICAÇÃO DO MGNL COM EXTRAÇÃO DE DADOS DE MÓDULOS FV

Dadas as limitações apresentadas pelo Bloco *PV Array* do MATLAB/Simulink no capítulo anterior, é importante buscar alternativas que contornem os problemas apresentados. Neste contexto, pode-se propor a utilização de outro princípio para modelagem de módulos FV.

O MGNL pode ser uma solução para mitigar alguns problemas apresentados. No entanto, para que o referido modelo funcione de maneira adequada, é necessário extrair algumas curvas I-V do módulo em análise e utiliza-las como um banco de dados para suprir o modelo. Desta forma, entende-se que o primeiro passo para a utilização adequada de curvas I-V extraídas de forma experimental para uso no MGNL é o desenvolvimento de uma metodologia de obtenção dos dados das curvas I-V de maneira adequada e confiável.

Assim sendo, este capítulo é dividido em sete seções: primeiramente é apresentado o desenvolvimento da bancada experimental utilizada para extração das curvas I-V; em seguida são explicitados os procedimentos adotados para o tratamentos dos dados obtidos com o extrator de curvas; posteriormente são abordadas as técnicas de obtenção de parâmetros para aplicação do MGNL; logo depois é apresentada a concepção do MGNL em ambiente MATLAB/Simulink; em seguida comparam-se os resultados com o Bloco *PV Array* do MATLAB/Simulink; e é apresentada uma alternativa de ajustes válida para a utilização do Bloco *PV Array*; por fim, é feita uma breve avaliação da utilização da metodologia proposta para um arranjo de quatro módulos.

4.1 BANCADA EXPERIMENTAL PARA EXTRAÇÃO DE CURVAS I-V

É importante voltar ao contexto do funcionamento do MGNL. Conforme já mencionado, o MGNL proposto por (SILVA et al., 2017) foi o precursor no desenvolvimento de modelos com essas características. Posteriormente outros modelos foram apresentados à comunidade científica, como (JUNIOR, 2023), (ZHANG et al., 2022a), (WANG et al., 2021), (LANG; ZHANG, 2020), entre outros.

Isto posto, conforme já mencionado, uma das etapas impreteríveis para o correto funcionamento da otimização destes modelos é a disposição de curvas I-V, sendo desejável que estas representem de maneira clara o funcionamento do sistema em análise. Portanto, estas curvas, preferencialmente em diferentes pontos de operação, serão utilizadas na etapa de treinamento. Desta forma é possível extrair os parâmetros necessários para a modelagem do sistema.

Desprende-se assim, a importância de um banco de dados com uma boa variabilidade de parâmetros e que representem de maneira coerente o comportamento do sistema em análise. Isto posto, esta seção visa apresentar de maneira clara e progressiva o desenvolvimento da metodologia de extração de curvas I-V.

4.1.1 Extrator de curvas I-V

Primeiramente, com o intuito de obter curvas I-V de diferentes modelos de módulos FV, utilizou-se o extrator de curva *PV Analyzer I-V Curve Tracer - PVA 1000S*, em conjunto com o sensor de irradiância, temperatura e inclinação sem fio *SolSensor*, ambos do fabricante *Solmetric*, ilustrados na Figura 33. Este conjunto dispõe da capacidade de medir as curvas I-V e P-V e possui as propriedades expostas na Tabela 4.

Figura 33 – Extrator de curvas - *PV Analyzer I-V Curve Tracer - PVA 1000S* e o sensor de irradiância, temperatura e inclinação sem fio *SolSensor*.



Fonte: O Autor (2024).

Tabela 4 – Propriedades do extrator de curva *PVA-1000S*.

Propriedade	PVA-1000S
Tensão Elétrica	20 V – 1000 V
Corrente Elétrica	0 A – 20 A
Irradiância	100 W/m ² – 1500 W/m ²
Precisão da Tensão	±0,5 V% ± 0,25 V
Precisão da Corrente	±0,5 A% ± 0,04 A
Precisão da Irradiância	± 2,5%
Quantidade de pontos	100 ou 500
Tempo de varredura	0,05 s – 2 s

Fonte: Adaptado de (BARBOSA, 2023).

Neste cenário, é relevante apresentar outros fabricantes e modelos de extratores de curvas, conforme discutido em (TRETER, 2023). Estes dispositivos são fundamentais para a caracterização precisa de módulos FV, permitindo uma análise detalhada do desempenho. No entanto, é importante destacar que há uma variação nos valores de tensão e corrente medidos, com imprecisões que podem variar entre 0,2% e 2%, dependendo do fabricante e do modelo

específico. Estas variações podem influenciar significativamente a avaliação da eficiência e do comportamento dos módulos FV. A Tabela 5 ilustra esta diversidade de soluções comerciais, evidenciando as diferenças nas especificações técnicas e nos níveis de precisão oferecidos pelos diferentes produtos. Destaca-se assim que o extrator utilizado apresenta uma precisão somente inferior ao DS-100 do fabricante Daystar.

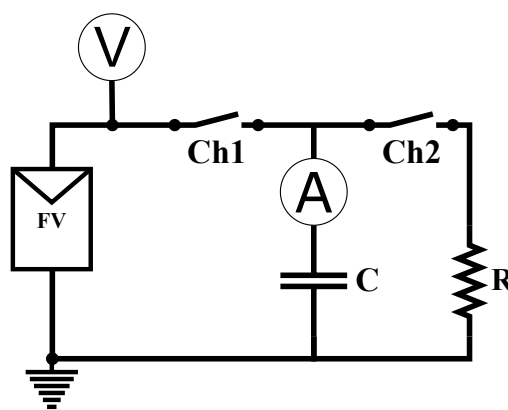
Tabela 5 – Soluções comerciais para a aquisição de curvas I-V.

Fabricante / modelo	Faixa de medição	Precisão (tensão; corrente)
Amprobe / SOLAR-4000	1000 V, 15 A	$\pm 0,5\%$ / $\pm 0,5\%$
Daystar / DS-1000	1000 V, 100 A	0,2~0,4% / 0,2~0,9%
EKO Instruments / MP-11	1000 V, 30 A	$\pm 1\%$ / $\pm 1\%$
Halm Electronic / cetisPV	1000 V, 100 A	Não informado
HT Instruments / I-V 500W	1500 V, 15 A	$\pm 0,5\%$ / $\pm 0,5\%$
Kewell / IVT-200-1000	1000 V, 200 A	$\pm 1\%$ / $\pm 1\%$
Pv Engineering / PVPM1500X	1500 V, 20 A	$\pm 1\%$ / $\pm 1\%$
Pordis / 140A-III	1000 V, 15 A	Não informado
Qunling / PV-8150	1000 V, 150 A	$\pm 2\%$ / $\pm 2\%$
NOYO Technology / PV1800	1000 V, 20 A	$\pm 1\%$ / $\pm 1\%$
Solmetric / PVA-1500V2	1000 V, 30 A	$\pm 0,5\%$ / $\pm 0,5\%$
Solmetric / PVA-1000S	1000 V, 20 A	$\pm 0,5\%$ / $\pm 0,5\%$

Fonte: Adaptado de (TRETER, 2023).

Conforme demonstrado em (SOLMETRIC, 2016), este traçador de curvas atua com o método da carga capacitiva, isto é um capacitor é carregado, varrendo os pontos entre curto-circuito e circuito aberto. Este método pode ser representado pelo circuito elétrico disposto na Figura 34 (TRETER et al., 2018).

Figura 34 – Circuito elétrico do método da carga capacitiva.



Fonte: Adaptado de (TRETER et al., 2018).

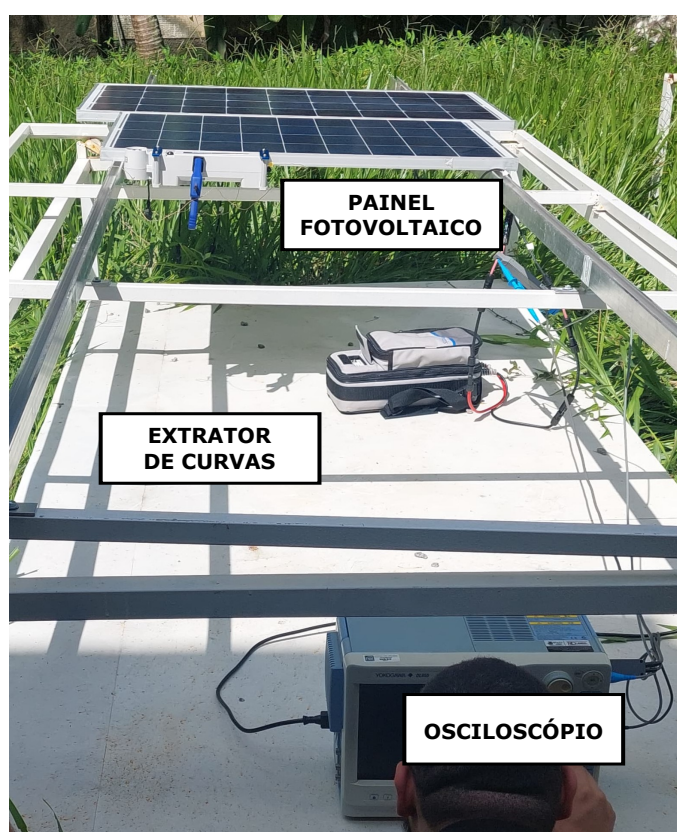
Para utilizá-lo, o capacitor deve iniciar em curto-circuito, posteriormente, a chave Ch2 é aberta e a chave Ch1 é fechada para iniciar o carregamento. Quando a carga do capacitor

aumenta, a corrente diminui e a tensão aumenta. Ao completar a carga, a corrente do módulo se torna zero, alcançando o circuito aberto. Assim sendo, para obter uma curva I-V precisa, são necessários capacitores de alta qualidade e baixas perdas. Adicionar uma fonte em paralelo ao capacitor, antes de conectá-lo ao módulo, ajuda a reduzir as perdas de tensão. A sequência correta de ativação das chaves e a descarga prévia do capacitor são fundamentais para iniciar uma nova medição (DURAN et al., 2008).

Neste ponto, cabe destacar que optou-se por utilizar o extrator de curvas na configuração de 500 pontos, visando assim uma maior clareza e definição dos dados analisados. Assim, o extrator tende a amostrar os pontos de interesse com uma menor periodicidade.

É necessário mencionar que a fonte controlada pode ser um elemento indutor de erros no processo, uma vez que a fonte controlada não representa de maneira eficiente o comportamento dinâmico do módulo (JUNIOR et al., 2024). Dito isto, para avaliar o comportamento do módulo e ter uma confirmação do funcionamento do extrator, pode-se executar uma montagem de maneira semelhante, utilizando um módulo (RSM-100P) como fonte de energia, conforme observado na Figura 35.

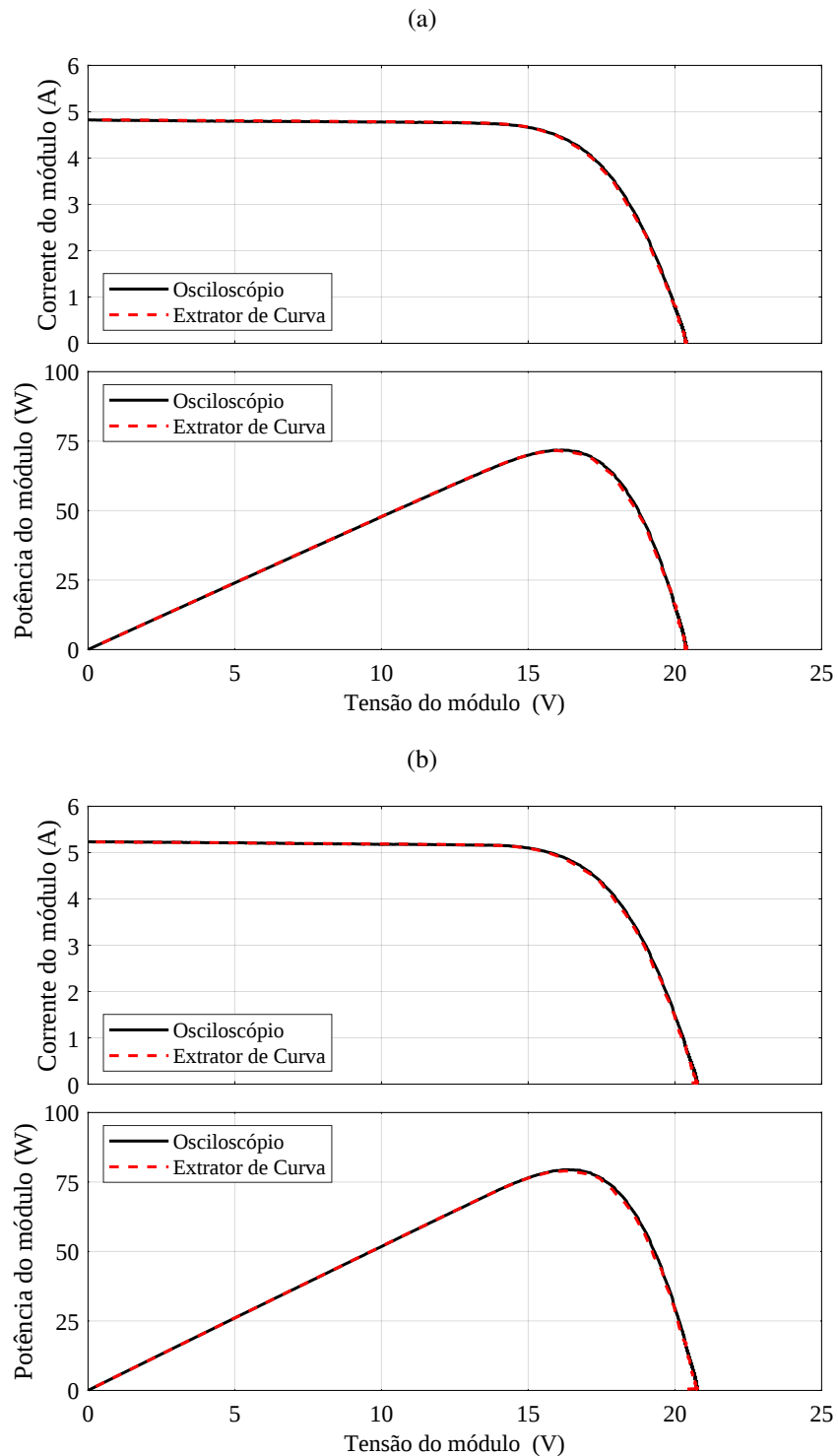
Figura 35 – Bancada experimental para verificação do extrator de curva com o painel fotovoltaico.



Fonte: O Autor (2024).

Por conseguinte, pode-se extrair curvas I-V distintas para o mesmo ensaio, a fim de comparar os resultados obtidos com o extrator de curvas com os resultados do osciloscópio, conforme exposto na Figura 36.

Figura 36 – Comparativo de curvas I-V obtidas com extrator de curvas e com osciloscópio.



Fonte: O Autor (2024).

Desprende-se assim, que os resultados obtidos com o extrator de curvas é bem aderente aos resultados medidos com o osciloscópio, embora o osciloscópio não possua precisão suficiente para validar o extrator de curvas, corrobora para a confiança nos resultados obtidos.

4.1.2 Montagem da bancada experimental

Vislumbrando aproximar a montagem das condições encontradas em sistemas FV existentes, foi concebida uma montagem de uma bancada experimental externa. Destaca-se também, que uma vez sendo desejável validar o uso do MGNL para avaliação de sistemas FV em condições reais de operação, entende-se como apropriado os ensaios em ambiente externo e sujeitos a variações das condições operacionais.

Isto posto, visando a extração experimental de curvas I-V, a referida bancada experimental foi montada conforme apresentado na Figura 37. E, utilizando esta estrutura, foram ensaiados três modelos distintos de módulos. As especificações desses módulos são apresentadas na Tabela 6.

Neste ponto, é válido mencionar que foram ensaiados ao menos 100 vezes cada módulo para obter um banco de dados razoável de curvas I-V. Para tanto, foi desenvolvida uma rotina automatizada utilizando o *software Power Automate* da *Microsoft* para realizar as medições utilizando o *Solmetrics PV Analyzer 4.0*, *software* do extrator de curvas, de forma contínua e em intervalos de tempo fixos. Salienta-se que a rotina criada pode ser adaptada para outras características de ensaio. Por exemplo, pode-se determinar as faixas de valores de irradiância e temperatura desejadas e fazer as medições somente quando o valor desejado for atingido.

Figura 37 – Bancada experimental externa para ensaios de módulos FV.



Fonte: O Autor (2024).

Tabela 6 – Características dos módulos ensaiados

Modelo	Fabricante	Potência Máxima	Tecnologia do Módulo
RSM-100P	RESUN SOLAR ENERGY	100 W	Silício Monocristalino
RS6E-155M	RESUN SOLAR ENERGY	155 W	Silício Policristalino
YL265C-30b	YINGLI SOLAR	265 W	Silício Monocristalino

Fonte: O Autor (2024).

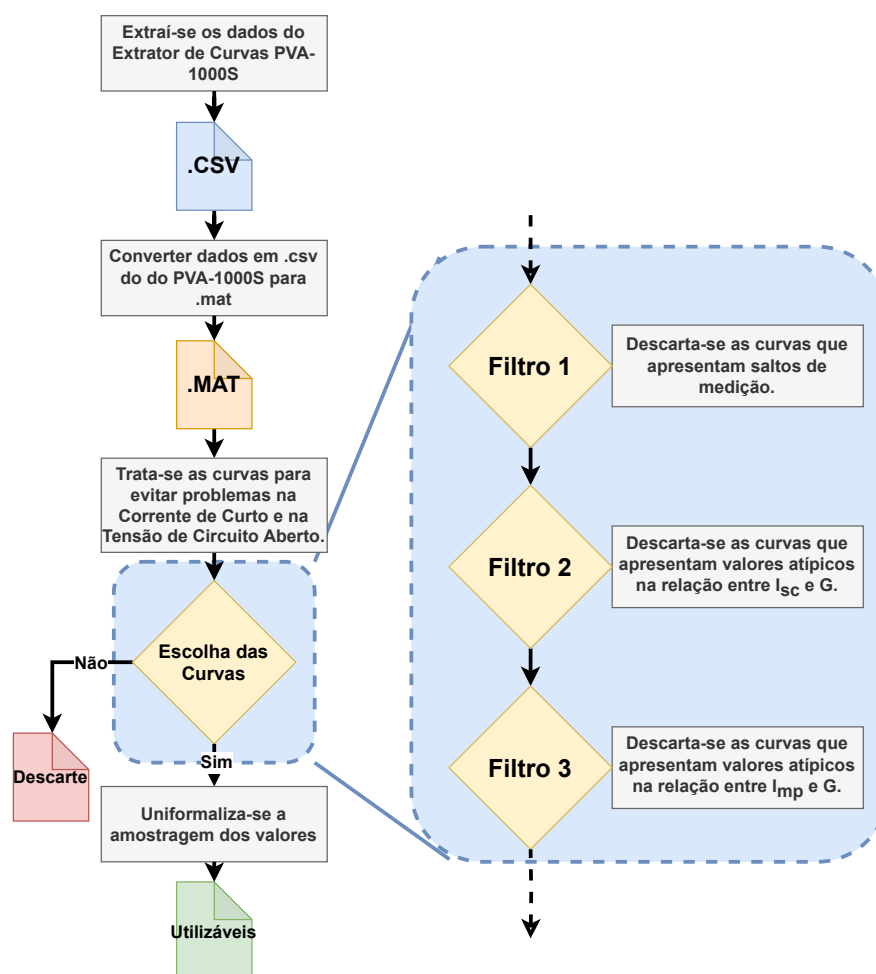
4.2 TRATAMENTO DOS DADOS OBTIDOS

Com o intuito de validar os valores obtidos com a bancada experimental, avaliou-se o comportamento das curvas I-V. Observou-se assim a necessidade da realização de um tratamento dos dados, e até mesmo descarte de algumas curvas que poderiam conduzir o MGNL a um resultado impreciso.

Partindo desta premissa, e com base em resultados obtidos, foi desenvolvida uma metodologia de tratamento de dados em quatro etapas, vislumbrando assim a obtenção de resultados mais adequados a utilização do MGNL. Em suma, pode-se apresentar o processo descrito para obtenção e tratamento das curvas I-V a serem utilizadas com o fluxograma da Figura 38. E entende-se que esse processo é primordial para garantir o funcionamento do MGNL.

Destaca-se que a Figura 38 apresenta um detalhamento dos processos de seleção de curvas utilizado, realçando as três etapas de filtragem, sendo a primeira o descarte das curvas que apresentam saltos de medição, assim como dois filtros consecutivos que desconsideram as curvas que apresentam valores atípicos nas relações $I_{sc} \times G$ e $I_{mp} \times G$, respectivamente.

Figura 38 – Fluxograma proposto para tratamento e filtro de curvas I-V a serem utilizadas como banco de dados do MGNL.



Fonte: O Autor (2024).

4.2.1 Tratamento inicial - Problemas na região de I_{sc} e V_{oc}

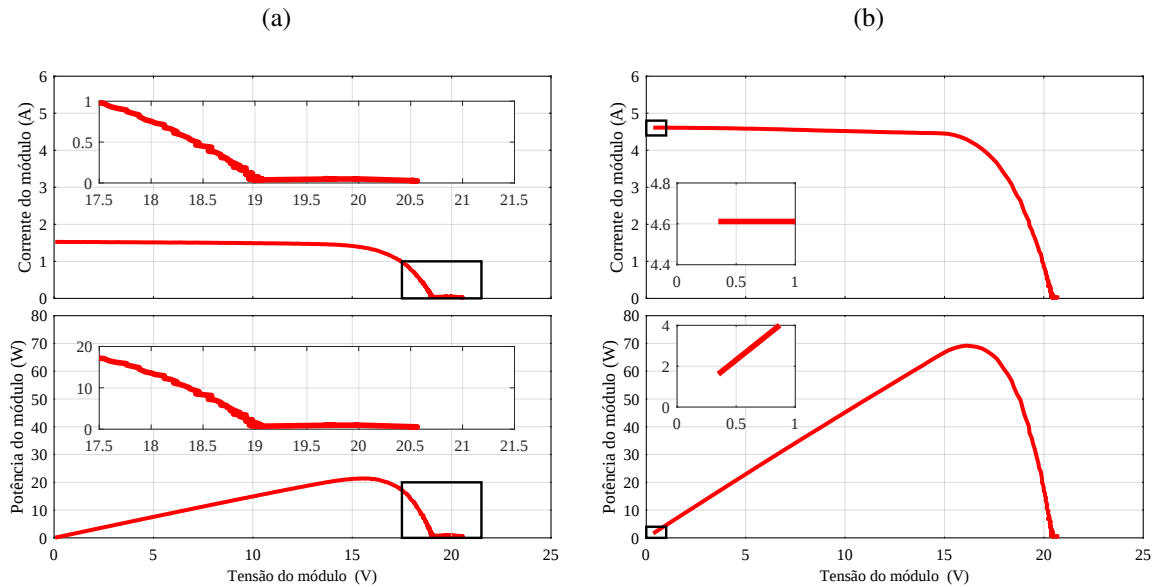
Inicialmente, ao realizar uma análise preliminar das curvas medidas, é possível identificar problemas relacionados aos pontos de I_{sc} e de V_{oc} em determinadas curvas. A Figura 39 exemplifica essas questões, apresentando medições do módulo RSM-100P com discrepâncias que podem ser corrigidas, permitindo a utilização adequada da curva.

Especificamente, na Figura 39(a), observa-se um deslocamento próximo ao ponto V_{oc} , onde a curva forma um platô após atingir este valor. Essa anomalia se torna mais evidente com a visualização ampliada da região, indicando uma falha na medição que pode afetar a precisão da curva.

Por outro lado, a Figura 39(b) mostra uma ausência de medição quando $V = 0$, ou seja, o valor de I_{sc} não está representado corretamente. Esta falha também é mais perceptível na visualização ampliada da região correspondente.

Ambas as falhas de medição podem levar a erros nos resultados do MGNL, uma vez que este modelo utiliza diretamente os parâmetros I_{sc} e V_{oc} . Portanto, é essencial identificar e corrigir essas discrepâncias para garantir a precisão das análises e simulações realizadas com base nas curvas medidas.

Figura 39 – Exemplos de falhas de medição do módulo RSM-100P: (a) erro próximo de V_{oc} , (b) erro próximo de I_{sc} .



Fonte: O Autor (2024).

Por conseguinte, dado que essas falhas específicas não comprometem os demais dados da curva I-V, decidiu-se desenvolver um método para tratamento dessas curvas visando mitigar os impactos dessas falhas. A Figura 40 ilustra esse processo de tratamento.

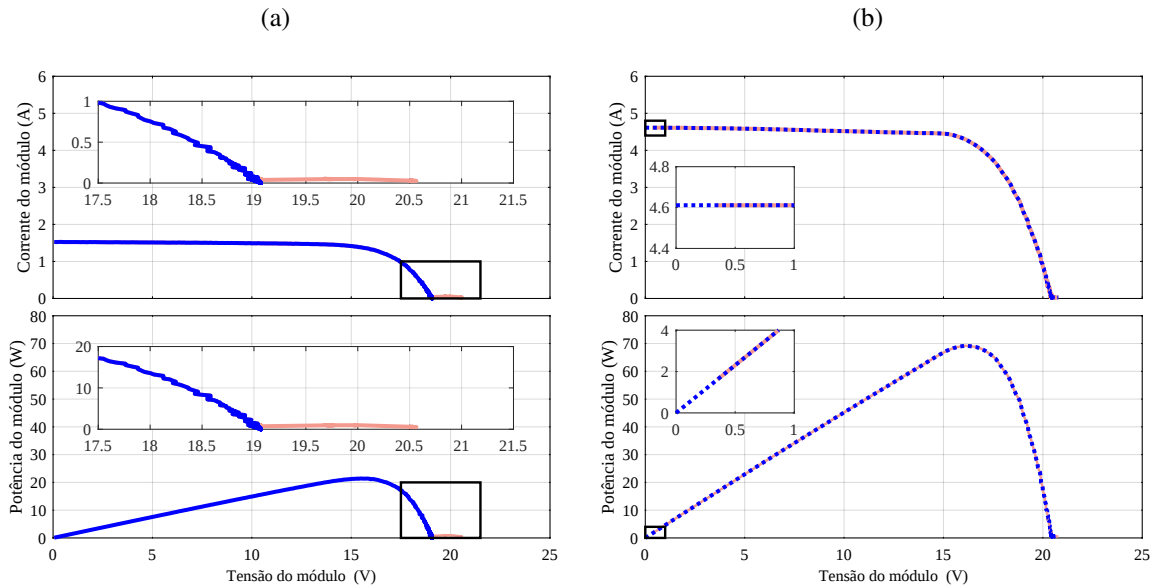
Na primeira parte do tratamento da curva, os valores da curva I-V após alcançar o ponto V_{oc} são descartados, conforme ilustrado na Figura 40(a). Essa abordagem vislumbra eliminar os

dados incorretos que formam um platô após V_{oc} , garantindo que a curva não represente valores de corrente após atingir o valor de V_{oc} .

Na segunda etapa do tratamento, define-se o valor da corrente quando a tensão é nula como I_{sc} . Em outras palavras, assume-se que $I(V = 0) = I_{sc}$, como mostrado na Figura 40(b). Esta correção é necessária para assegurar que a curva represente corretamente o ponto de I_{sc} , eliminando a falha de medição observada.

Esses passos garantem que as curvas I-V tratadas sejam mais confiáveis para aplicação do MGNL, permitindo análises e simulações mais acuradas com base nos dados corrigidos, resultando assim melhores resultados quando analisada métricas de erro.

Figura 40 – Correção de falhas de medição do módulo RSM-100P: (a) correção próxima de V_{oc} (b) correção próxima de I_{sc} .



Fonte: O Autor (2024).

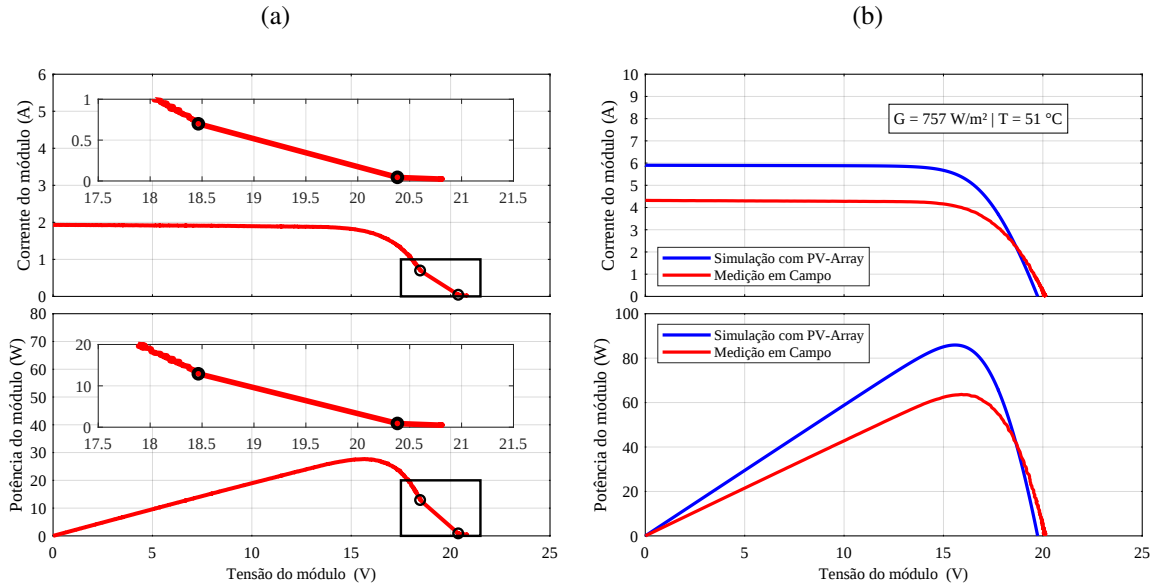
4.2.2 Triagem de curvas

Existem imprecisões observadas que inviabilizam a utilização de algumas curvas I-V específicas, como mostrado na Figura 41. Uma dessas incorreções é a ausência de medições em determinados trechos da curva. Por exemplo, na Figura 41(a), há um salto de medição entre 18,4641 V e 20,3816 V, sem qualquer medição de corrente nesse intervalo. Isso resulta em uma mudança abrupta na corrente medida, de 0,6996 A para 0,0446 A. Cabe ressaltar que estes eventos são mais recorrentes quando há uma perda de comunicação momentânea entre o computador e o extrator de curva.

Além disso, é importante mencionar que problemas de medição podem ocorrer devido a limitações físicas da bancada experimental externa. A medição de irradiância pode não estar perfeitamente alinhada com a extração da curva I-V. Como exemplo, o sensor de irradiância pode estar sombreado enquanto o módulo FV não está causando uma disparidade entre as curvas

extraídas e os valores esperados. Para representar comparativamente, os valores esperados são determinados pela translação das curvas com base nos dados do *datasheet*, como ilustrado na Figura 41(b).

Figura 41 – Exemplos de falhas de medição do módulo RSM-100P: (a) ausência de medição, (b) incompatibilidade entre a curva I-V e o valor de G medido.



Fonte: O Autor (2024).

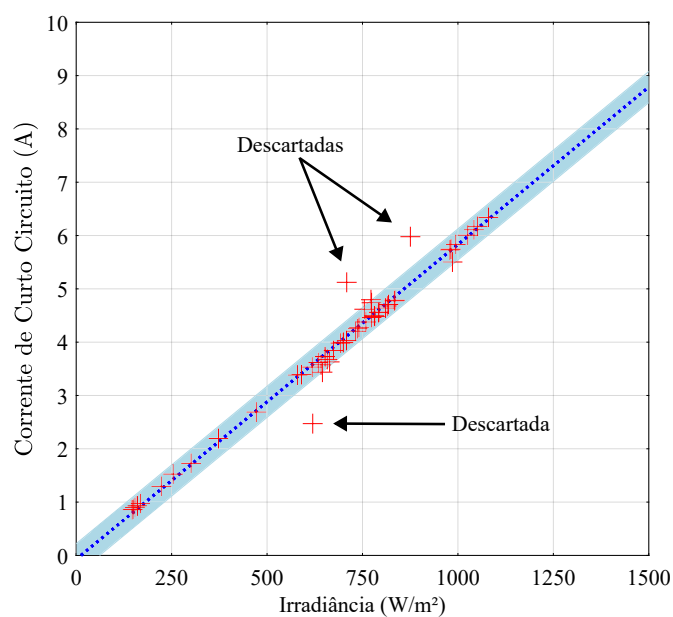
Esses problemas destacam a necessidade de um tratamento cuidadoso dos dados de medição para garantir a precisão e a confiabilidade das análises realizadas com base nas curvas I-V.

Assim sendo, a primeira abordagem para a seleção das curvas I-V consistiu no descarte das curvas que apresentaram degraus na medição de tensão superiores a 1% do valor de V_{oc} no terço final da curva, ou seja, entre $\frac{2}{3}V_{oc}$ e V_{oc} . Este segmento é crítico, pois é a parte menos linear da curva I-V, e a presença de degraus de medição nesse trecho pode causar distorções significativas na análise e interpretação dos dados.

Visando mitigar os efeitos da disparidade entre a G e a I_{sc} , foi implementado um polinômio de primeiro grau ajustado com base nos parâmetros de G e I_{sc} de todas as curvas I-V extraídas de um módulo em análise. De forma empírica, utilizou-se uma tolerância de 5% para mais e para menos, estabelecendo assim uma margem de tolerância para os valores obtidos. As curvas I-V que apresentaram valores fora dessa margem foram descartadas. Este processo de triagem é ilustrado graficamente na Figura 42.

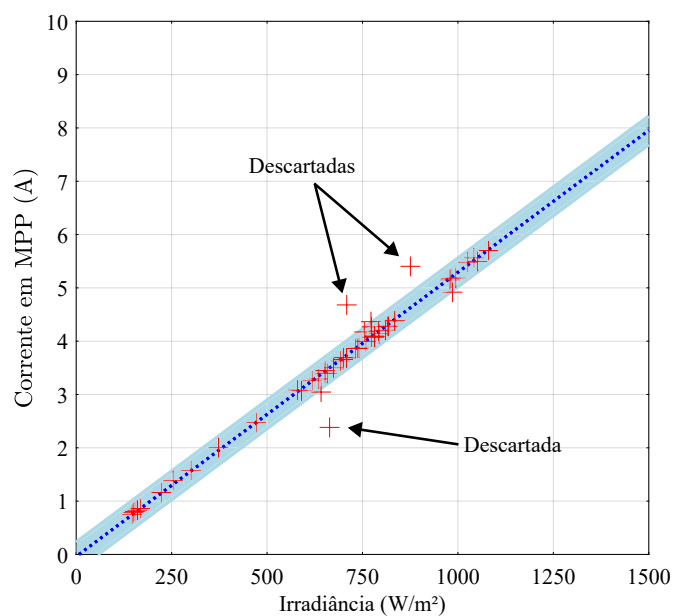
No mesmo diapasão, correlaciona-se a G e a I_{mpp} , conforme ilustra a Figura 43.

Figura 42 – Exemplo de relação entre I_{sc} (A) e G (W/m^2) do módulo RSM-100P utilizada como método de filtragem.



Fonte: O Autor (2024).

Figura 43 – Exemplo de relação entre I_{mpp} (A) e G (W/m^2) do módulo RSM-100P utilizada como método de filtragem.



Fonte: O Autor (2024).

Estes dois critérios integrados proporcionam o descarte de curvas irregulares, inclusive descartando curvas que apresentam sombreamentos e valores muito discrepantes para este banco de curvas.

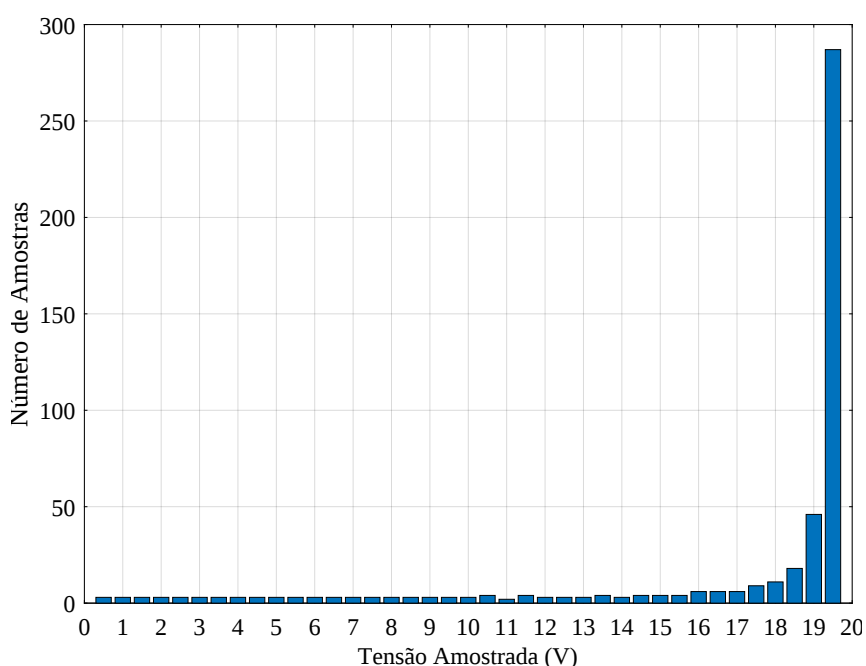
4.2.3 Uniformização das amostras

É importante notar que as medições no início da curva I-V possuem menos valores amostrados, enquanto mais próximo ao final da curva I-V a amostragem é significativamente maior.

Para ilustrar esse fato de forma mais clara, pode-se exemplificar a distribuição das amostras de uma curva extraída do módulo RSM-100P, agrupando a quantidade de amostras em intervalos específicos de tensão. Em caráter ilustrativo, a cada $0,5V$, quantificam-se as amostras, como mostrado na Figura 44. Por exemplo, entre $11,25V$ e $11,75V$ existem 4 amostras, enquanto entre $19,25V$ e $19,75V$ existem 287 amostras. Esta disparidade na distribuição das amostras evidencia a variação significativa na densidade de amostragem ao longo da curva I-V.

Assim sendo, esse comportamento sugere uma distribuição desigual das amostras ao longo da curva, o que pode impactar a precisão da análise nos diferentes segmentos da curva I-V, principalmente a depender das metodologias utilizadas para a avaliação dos resultados, uma vez que, quando se têm muitas amostras em um determinado ponto pode-se calcular os erros inerentes àquele ponto diversas vezes.

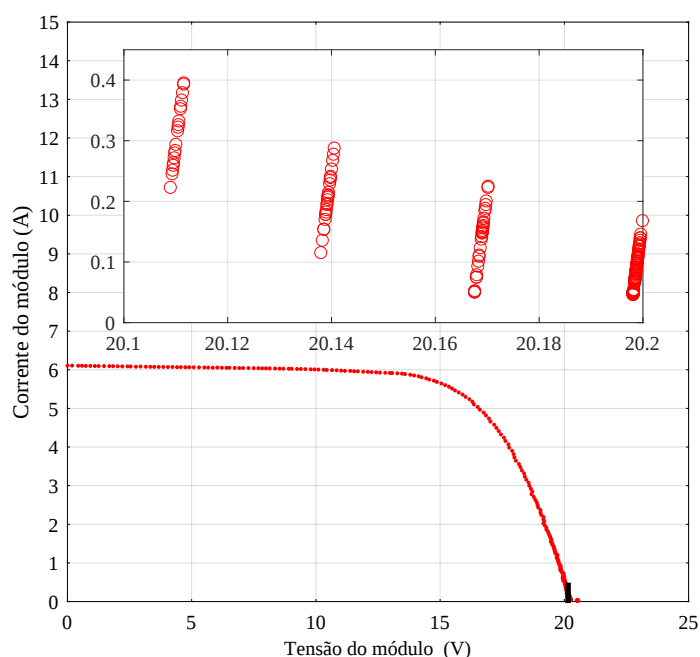
Figura 44 – Distribuição de amostras ao longo da curva I-V para o módulo RSM-100P.



Fonte: O Autor (2024).

Fato este pode ser compreendido pelo princípio de funcionamento do extrator, que é baseado na descarga de um capacitor. Por essa características de coleta de dados do extrator de curvas, pode-se perceber que eventualmente são apresentadas diversas medições de corrente em pontos próximos de tensão em valores próximos ao V_{oc} , como pode ser observado na Figura 45, em especial na aproximação apresentada.

Figura 45 – Exemplo de problema de curva I-V com múltiplas medições em um pequeno intervalo de tensão de um módulo RSM-100P.



Fonte: O Autor (2024).

No exemplo em pauta, conforme Figura 45, dentro de uma janela amostral de 100 mV (entre $20,1\text{ V}$ e $20,2\text{ V}$) pode-se notar 176 medições de corrente distintas. Embora essa curva I-V não apresente grandes discrepâncias do ponto de vista visual, a perspectiva muda quando consideramos as métricas utilizadas para determinar a eficiência, como EMAPN e NRMSD. Estas métricas são calculadas ponto a ponto e seus valores são acumulados. Assim, a presença de múltiplas medições em uma pequena faixa de tensão pode interferir significativamente nos resultados, afetando a precisão e a confiabilidade das análises de eficiência.

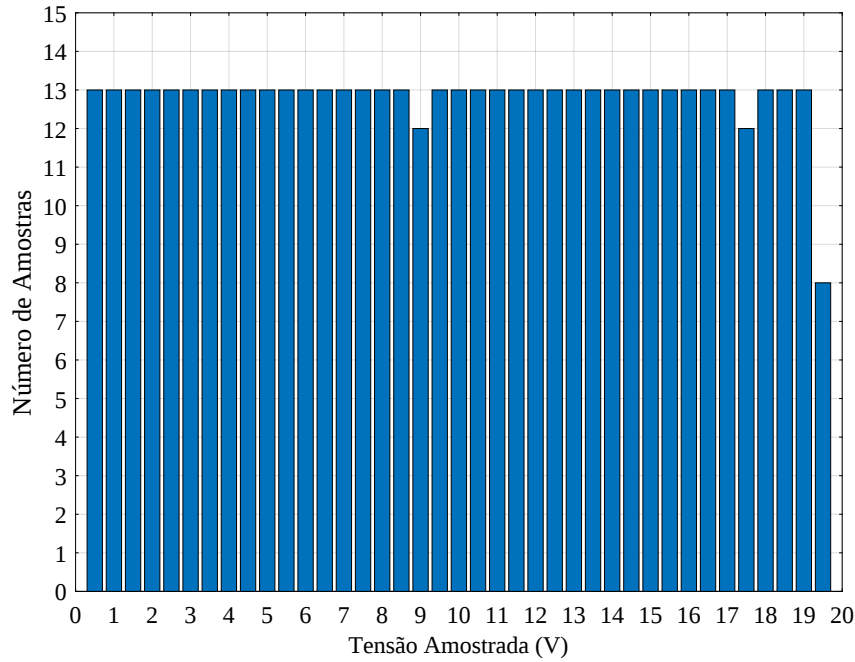
Nesse diapasão, optou-se em fixar o número de amostras e distribuir de maneira igualitária entre 0 V e V_{oc} , ou seja, deixar todas as amostras igualmente distribuídas, conforme Figura 46. Visando mitigar os problemas relacionados às características do método de modelagem a ser utilizado, este tratamento do dado reduz significativamente a quantidade de amostras em um intervalo amostral, sem descaracterizar a curva I-V, visto que entende-se que estas múltiplas medições são intrínsecas ao modo de operação do extrator, e não medições factuais. Apresenta-se de maneira mais visual este método na Figura 47.

4.3 OBTENÇÃO DOS PARÂMETROS E APLICAÇÃO DO MGNL

4.3.1 Análise dos dados

É importante mencionar que os dados obtidos dos módulos ensaiados foram avaliados previamente, antes de dar continuidade aos procedimentos que serão descritos. Assim, inicialmente, foram ensaiados três módulos RSM-100P, três módulos RS6E-155M e um

Figura 46 – Distribuição de amostras ao longo da curva I-V para o módulo RSM-100P após tratamento.



Fonte: O Autor (2024).

módulo YL265C-30b. No entanto, como os módulos RS6E-155M não apresentaram resultados satisfatórios, optou-se por ensaiar mais três módulos deste modelo, mantendo-se insatisfatórios, conforme pode ser observado na Figura 48. Desta forma, concluiu-se que o lote deste modelo não traria benefícios aos desenvolvimentos das métricas e estudos deste trabalho.

De certa forma, este achado corrobora com informações desprendidas do capítulo anterior, e entende-se que deve-se trazer um olhar mais cético aos dados oriundos dos manuais dos fabricantes, assim como da qualidade dos materiais produzidos. De toda forma, dar-se-á continuidade à análise dos dados somente dos módulos RSM-100P e YL265C-30b.

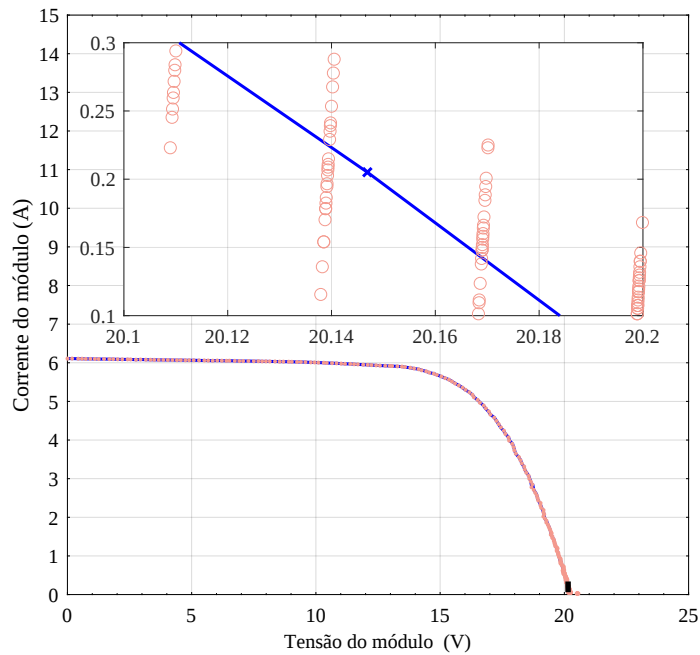
4.3.2 MGNL utilizado

Entende-se por pertinente observar que o modelo completo apresentado em (SILVA, 2019) possui oito parâmetros ($R_{s,ref1}$, $R_{s,ref2}$, k_{Rs} , γ_{Rs} , $R_{sh,ref}$, k_{Rsh} , γ_{Rsh} , e n_{ref}) que são efetivamente estimados. Assim, os dois parâmetros finais do SDM são determinados explicitamente em relação aos demais parâmetros (JUNIOR, 2023), com suas equações representadas por:

$$I_{ph} = I_{sc} \left(1 + \frac{R_s}{R_{sh}} \right), \quad (30)$$

$$I_{sat} = \frac{I_g - \frac{V_{oc}}{R_{sh}}}{e^{\frac{V_{oc}}{V_t}} - 1}. \quad (31)$$

Figura 47 – Exemplo de tratamento da curva I-V com múltiplas medições em um pequeno intervalo de tensão de um módulo RSM-100P.



Fonte: O Autor (2024).

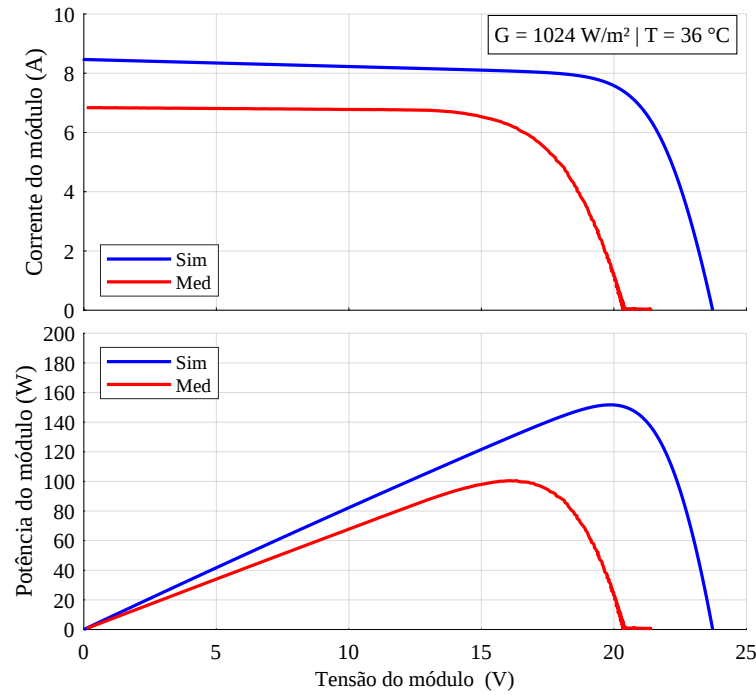
Assim, é fundamental destacar que durante os processos de otimização são adotados os seguintes critérios:

- Número máximo de iterações: 10.000;
- Número máximo de funções avaliadas: 10.000;
- Tolerância do término no tamanho da malha parâmetro do PS: $2,22 \cdot 10^{-16}$;
- Tolerância de término no valor das variáveis de entrada: $2,22 \cdot 10^{-16}$;
- Tolerância de término no valor da função: $2,22 \cdot 10^{-16}$.

Por fim, a Figura 49 ilustra a representação gráfica das etapas para realizar o modelo de (SILVA et al., 2017).

Cabe-se aqui ressaltar que, com base nos resultados apresentados em (FERNANDES, 2024), optou-se por alterar os limites superiores e inferiores dos parâmetros apresentados em (SILVA, 2019). Após a análise da saturação dos parâmetros em seus limites superior e inferior, ajustaram-se os limites de γ_{Rs} e γ_{Rsh} (superior) e k_{Rsh} (inferior) conforme apresenta a Tabela 7. Com essa modificação, garante-se que os valores encontrados pelo algoritmo de otimização estejam dentro do intervalo de busca e não saturados ao utilizar o modelo de Silva (FERNANDES, 2024).

Figura 48 – Comparativo entre o valor da Curva I-V medido em bancada experimental e valores esperados baseado em método de translação de curva.



Fonte: O Autor (2024).

Tabela 7 – Limites inferiores e superiores utilizados na busca de parâmetros para o modelo de (SILVA et al., 2017) modificado.

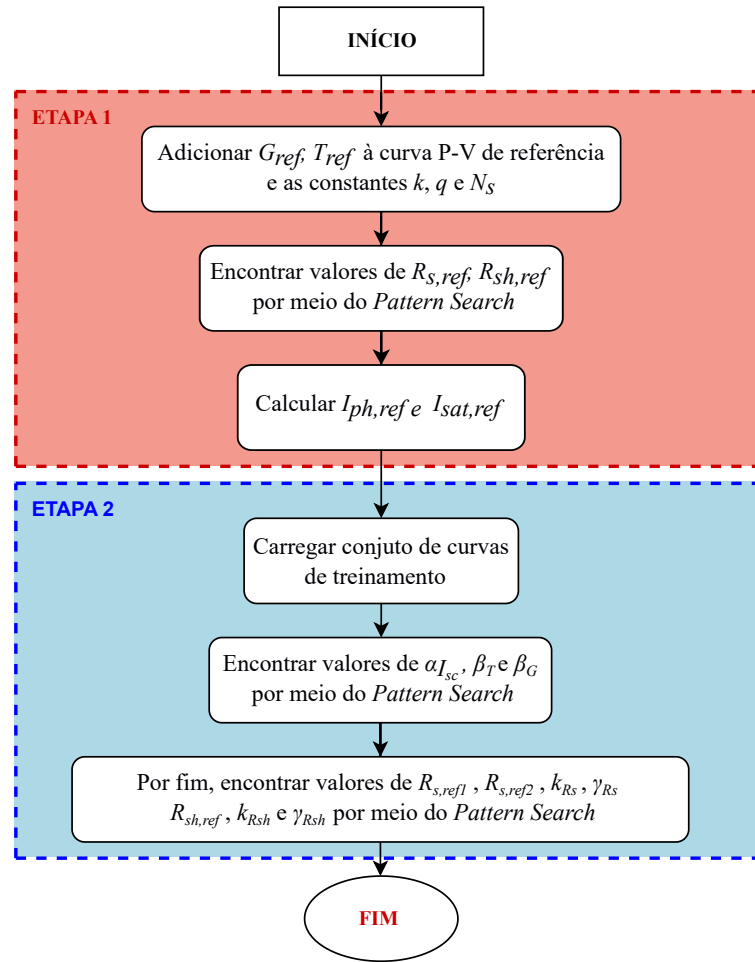
Parâmetro	Mínimo	Máximo
$R_{s,ref}$	0	$ R_{s,max} $
$R_{sh,ref}$	$R_{sh,min}$	10000
$k_{R_{sh}}$	$- k_{R_{sh},min} $	$ k_{R_{sh},min} $
k_{R_s}	$- k_{R_s,max} $	$ k_{R_s,max} $
$\gamma_{R_{sh}}$	$ \gamma_{R_{sh},min} $	$- \gamma_{R_{sh},min} $
γ_{R_s}	$ \gamma_{R_s,min} $	$- \gamma_{R_s,min} $
n	0	10
$\alpha_{I_{sc}}$	10^{-2}	10^{-4}
$\beta_{V_{oc}}$	-1	10^{-3}

Fonte: Adaptado de (FERNANDES, 2024)

4.3.3 Parâmetros extraídos

De posse das curvas tratadas e filtradas, pode-se incorporá-las no MGNL de Silva e, assim, obter os valores dos parâmetros desejados. Portanto, foram experimentados, dessa forma, três módulos distintos do modelo RSM-100P do fabricante Resun Solar, além de um módulo do YL265C-30b do fabricante Yingli Solar, que embora não seja um módulo recém adquirido, não estava em uso contínuo, encontrava-se armazenado em ambiente sem exposição as intempéries do tempo.

Figura 49 – Fluxograma básico do MGNL de Silva.

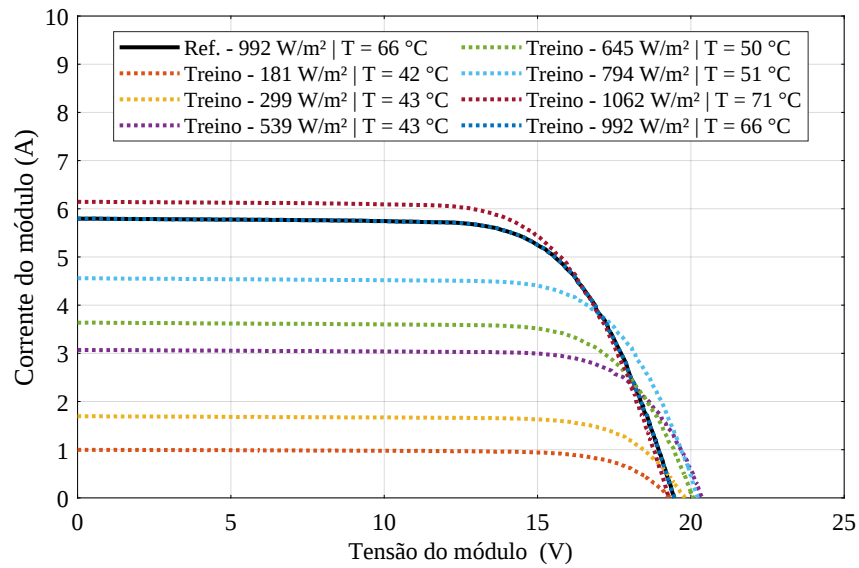


Fonte: Adaptado de (CAVALCANTE et al., 2023).

4.3.3.1 RSM-100P

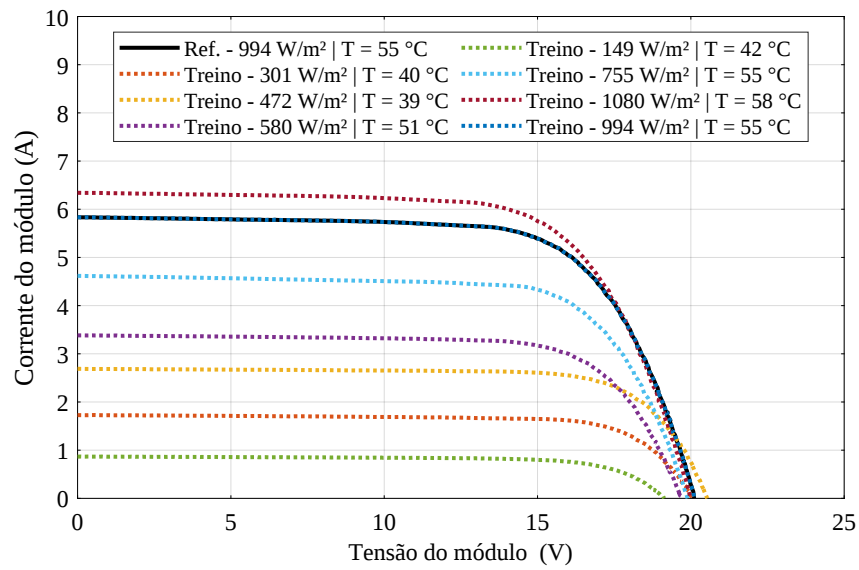
Para cada um dos três módulos ensaiados, foram selecionadas diferentes curvas I-V como referência e como treino, conforme observado nas Figuras 50, 51 e 52, e descrito na Tabela 8.

Figura 50 – Curvas I-V selecionadas como referência e como treinamento para o primeiro módulo ensaiado RSM-100P (P1).



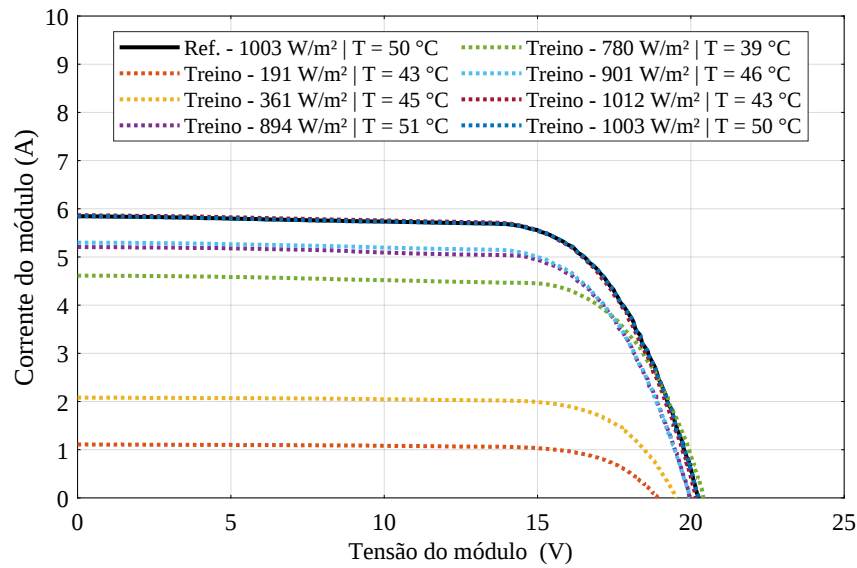
Fonte: O Autor (2024).

Figura 51 – Curvas I-V selecionadas como referência e como treinamento para o segundo módulo ensaiado RSM-100P (P2).



Fonte: O Autor (2024).

Figura 52 – Curvas I-V selecionadas como referência e como treinamento para o terceiro módulo ensaiado RSM-100P (P3).



Fonte: O Autor (2024).

Tabela 8 – Curvas de referência e treinamento utilizadas no modelo de Silva para cada módulo do *RSM-100P*.

Curvas de referência e treinamento para o MGNL					
Módulo 1		Módulo 2		Módulo 3	
$G(W/m^2)$	$T(^{\circ}C)$	$G(W/m^2)$	$T(^{\circ}C)$	$G(W/m^2)$	$T(^{\circ}C)$
992*	66*	994*	55*	1003*	50*
181	42	301	40	191	43
299	43	472	39	361	45
539	43	580	51	894	51
645	50	149	42	780	39
794	51	755	55	901	46
1062	71	1080	58	1012	43
992	66	994	55	1003	50

* Valores de Referência

Fonte: O Autor (2024).

Assim, de posse das referidas curvas de referência e de treinamento, foi possível realizar as etapas do MGNL de Silva e obter os resultados apresentados na Tabela 9. Estes parâmetros obtidos foram confrontados com todas as curvas disponíveis no banco de dados daquele módulo, obtendo assim os valores de média dos EMAPN e dos NRMSD apresentados na Tabela 9.

Deve-se aqui observar que há uma discrepância entre os valores obtidos pelos três módulos, considerando que pertencem ao mesmo fabricante e são oriundos do mesmo lote, é importante destacar os padrões de qualidade do processo de fabricação. Desta forma, com o intuito de investigar a estrutura da caixa de junção de um destes módulos, conforme apresenta a Figura 53.

Tabela 9 – Parâmetros obtidos a partir dos módulos do modelo RSM-100P ensaiados em ambiente externo.

Parâmetros	Módulo P1	Módulo P2	Módulo P3
$R_{s,ref1}$	0,15 Ω	0,11 Ω	0,14 Ω
$R_{s,ref2}$	0,15 Ω	0,11 Ω	0,14 Ω
$R_{s,ref}$	0,29 Ω	0,23 Ω	0,29 Ω
$R_{sh,ref}$	247,13 Ω	115,04 Ω	98,73 Ω
k_{Rs}	1,46 %/ $^{\circ}C$	-0,78 %/ $^{\circ}C$	-3,28 %/ $^{\circ}C$
k_{Rsh}	-1,38 %/ $^{\circ}C$	0,20 %/ $^{\circ}C$	-0,08 %/ $^{\circ}C$
$k_{v,T}$	-0,06 V/ $^{\circ}C$	-0,07 V/ $^{\circ}C$	-0,00 V/ $^{\circ}C$
$k_{v,G}$	0,87	0,78	0,77
k_i	6,68 mA/ $^{\circ}C$	9,33 mA/ $^{\circ}C$	0,00 mA/ $^{\circ}C$
γ_{Rs}	-0,28	1,92	1,02
γ_{Rsh}	0,19	-1,12	-0,72
n_{ref}	1,12	1,33	1,07
$I_{g,ref}$	5,80 A	5,85 A	5,86 A
$I_{sat,ref}$	401,62 nA	2063,93 nA	38,40 nA
Média dos EMAPNs	0,87 %	0,76 %	1,39 %
Média dos NRMSDs	1,22 %	1,26 %	2,19 %

Fonte: O Autor (2024).

Figura 53 – Detalhe da caixa de junção de um dos módulos ensaiados do tipo RSM-100P.



Fonte: O Autor (2024).

A partir da análise da Figura 53, nota-se que as soldas são grosseiras e, além disso, observa-se que a conexão entre os dois diodos de passagem (*by-pass*, em inglês) não está conectada a nada, levantando assim o questionamento acerca da necessidade destes dois diodos. Não obstante, deve-se mencionar que o texto do *datasheet* deste módulo apresenta expressamente que a caixa de junção é formada por seis diodos de passagem, conforme Figura 54, fato este que pode ser facilmente refutado na Figura 53.

Figura 54 – Trecho do *datasheet* do RSM-100P que versa sobre as características mecânicas do módulo.

MECHANICAL DATA	
Solar cells	Poly-crystalline 156x91mm 4 or 5 Bus bars
Cell configuration	36 cells(3x12)
Module dimensions	1020x670x30 mm
Weight	7KGS
Front Cover	3.2mm Tempered Glass
Frame Material	Anodized Aluminum Alloy
J-BOX	IP65 or IP67, 6 Diodes
Cable	4mm2(IEC)/12AWG(UL),900mm
Connectors	MC4 or MC4 Comparable
Standard Packaging	4pcs/suite box

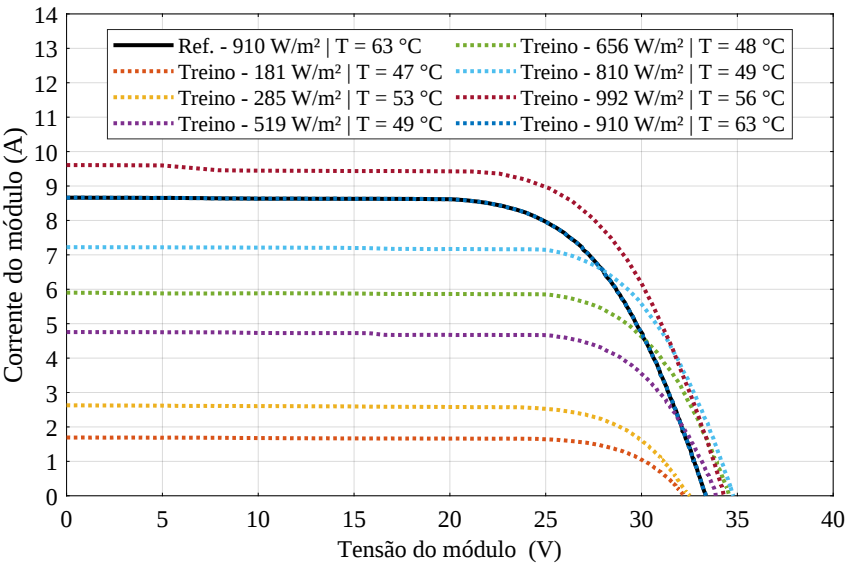
Fonte: (RESUN, 2024).

Desta forma, com esses parâmetros, pode-se dar continuidade à modelagem do módulo em ambiente simulado do MATLAB/Simulink.

4.3.3.2 YL265C-30b

De maneira análoga, foram realizados os mesmos procedimentos com o módulo YL265C-30b, empregando as curvas de referência e de treinamento, conforme Figura 55 e Tabela 10, e obtendo assim os parâmetros conforme Tabela 11.

Figura 55 – Curvas I-V selecionadas como referência e como treinamento para o módulo ensaiado YL265C-30b.



Fonte: O Autor (2024).

Tabela 10 – Curvas de referência e treinamento utilizadas no modelo de (SILVA, 2019) para o módulo *YL265b-30C*.

Curvas de referência e treinamento para o MGNL	
Módulo	
$G(W/m^2)$	T(°C)
910*	63*
181	47
285	53
519	49
656	48
810	49
992	56
910	63

* Valores de Referência

Fonte: O Autor (2024).

Tabela 11 – Parâmetros obtidos a partir do módulo do modelo YL265b-30C ensaiado.

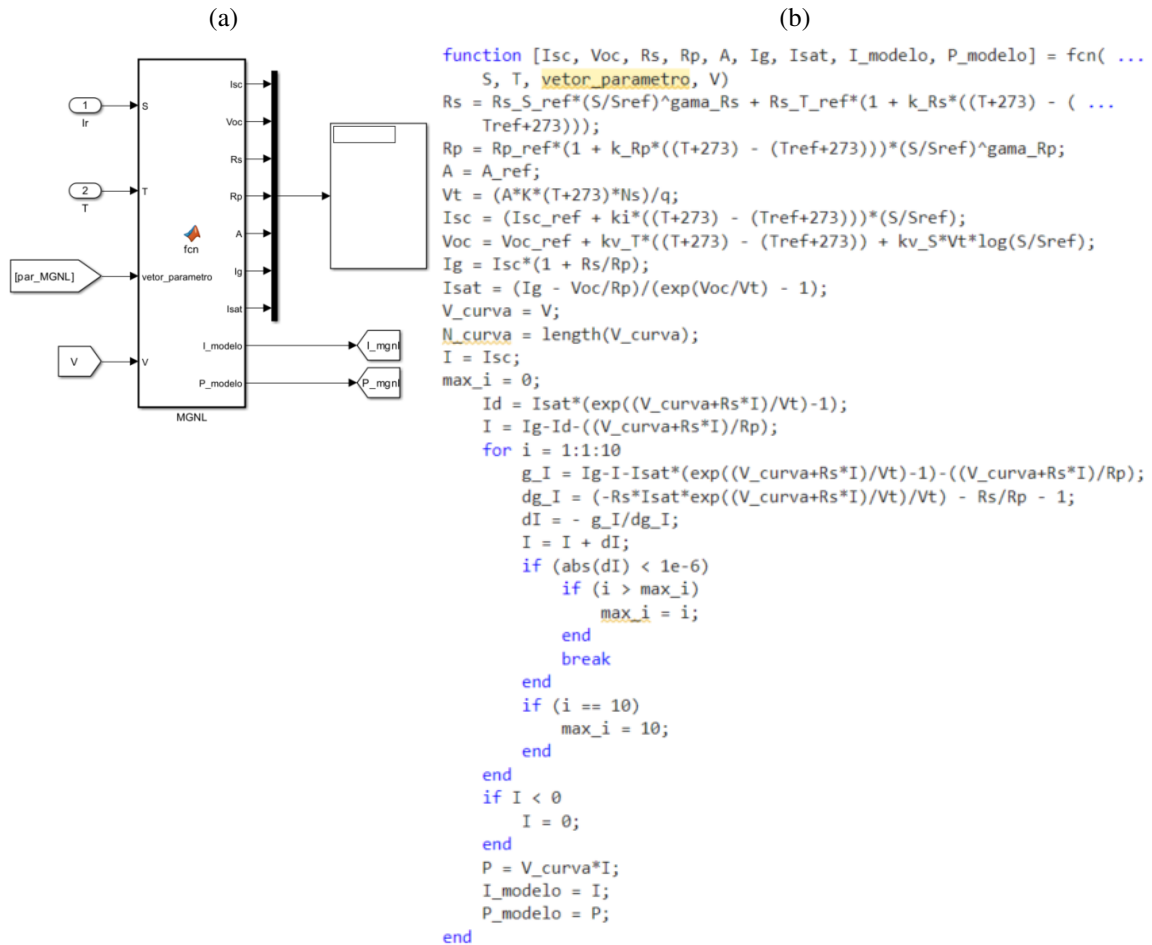
Parâmetros	Módulo
$R_{s,ref1}$	0,17 Ω
$R_{s,ref2}$	0,17 Ω
$R_{s,ref}$	0,33 Ω
$R_{sh,ref}$	1631,73 Ω
k_{Rs}	1,46 %/°C
k_{Rsh}	-5.35 %/°C
$k_{v,T}$	-0,12 V/°C
$k_{v,G}$	0,88
k_i	31,67 mA/°C
γ_{Rs}	0,59
γ_{Rsh}	-2,71
n_{ref}	1,29
$I_{g,ref}$	8,66 A
$I_{sat,ref}$	3075,81 nA
Média dos EMAPNs	0,98 %
Média dos NRMSDs	1,47 %

Fonte: O Autor (2024).

4.4 CONCEPÇÃO DO MGNL NO MATLAB/SIMULINK

Inicialmente, após a obtenção dos parâmetros necessários para o MGNL, foi desenvolvido um bloco em ambiente MATLAB/Simulink para sua implementação, conforme demonstrado na Figura 56. A estrutura escolhida para o modelo é baseada em um bloco *MATLAB Function* (Figura 56 (a)), que requer a criação de um *script* específico (Figura 56 (b)).

Figura 56 – Elemento principal da implementação do bloco em Simulink do MGNL: (a) apresentação em Simulink e (b) algoritmo da função de cálculo do bloco.



Fonte: O Autor (2024).

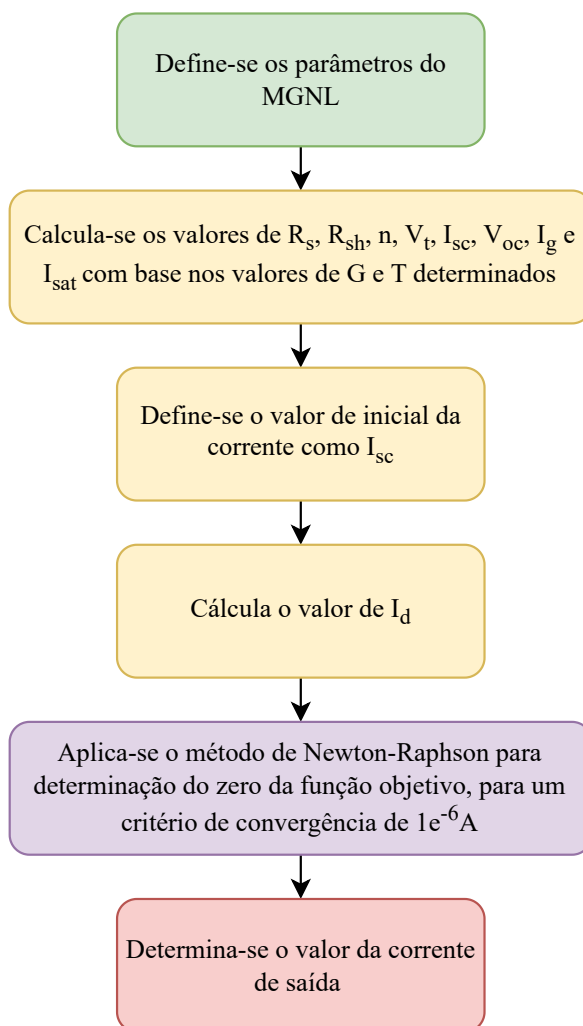
Desta forma, visando deixar mais claro o *script* apresentado, pode-se apresentar o fluxograma da Figura 57, que descreve de maneira objetiva as etapas do algoritmo de formação do bloco do MGNL.

Assim, esta metodologia possibilita a incorporação eficiente dos parâmetros e a execução adequada do modelo. A utilização do MATLAB/Simulink proporciona uma comparação justa com o bloco *PV Array*, além de um ambiente para simulação e análise dos resultados.

4.5 COMPARAÇÃO DOS RESULTADOS DO *PV ARRAY* E DO MGNL

Como uma etapa inicial de validação, é fundamental comparar os resultados das curvas P-V em diferentes níveis de irradiância e temperatura, utilizando tanto o MGNL quanto o bloco *PV Array*. Esse processo é essencial para verificar a capacidade de ambos os modelos em reproduzir as características reais dos módulos FV sob condições diversas. Além disso, a apresentação gráfica do EMAPN torna a comparação mais clara e objetiva, facilitando a identificação de eventuais disparidades entre os modelos.

Figura 57 – Fluxograma do algoritmo de função de cálculo do bloco MGNL.

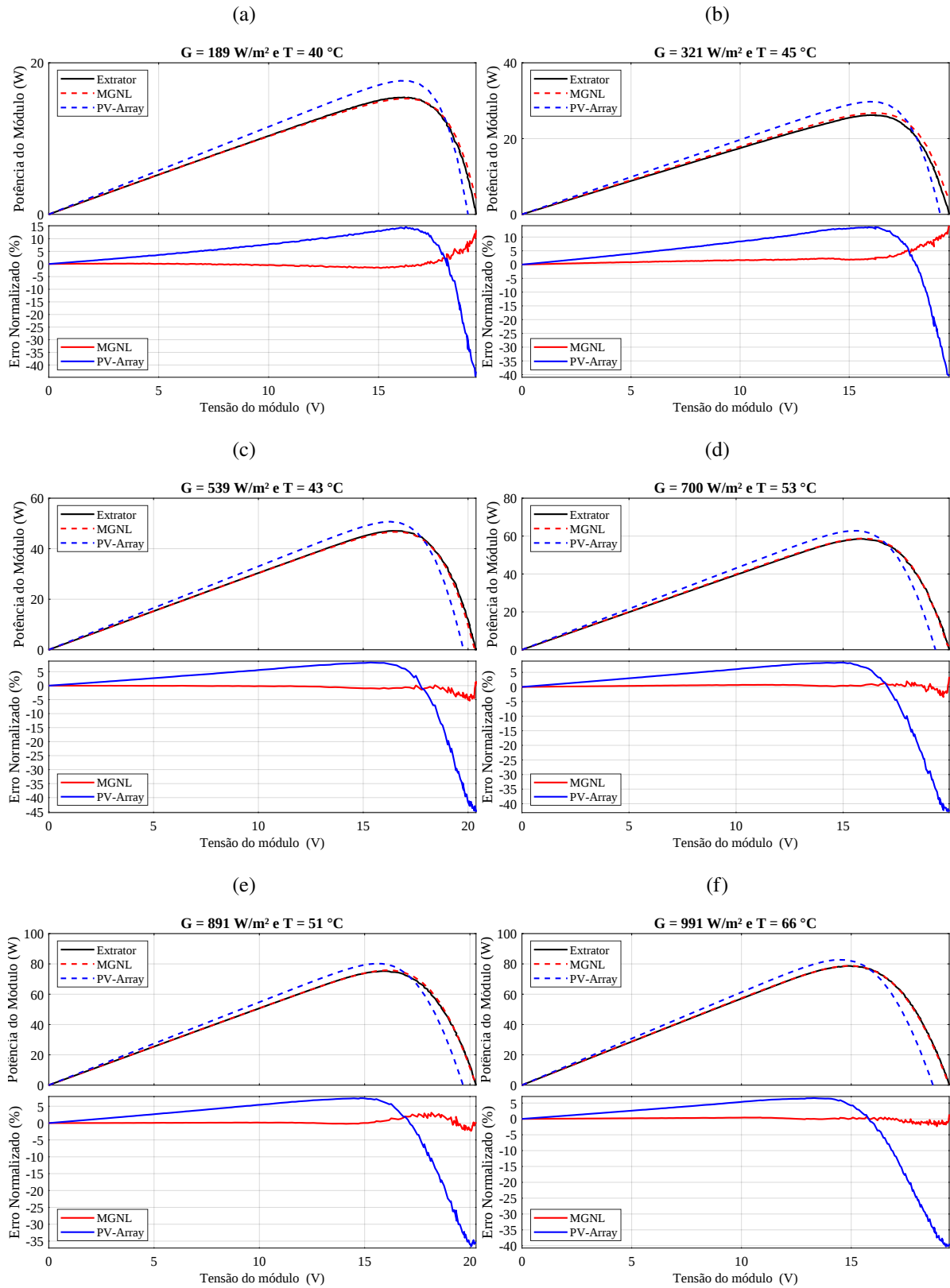


Fonte: O Autor (2024).

Nesse sentido, as Figuras 58 e 59 mostram os resultados para os módulos RSM-100P e YL265C-30b, respectivamente, optou-se por apresentar os resultados apresentados visando apresentar valores para irradiâncias e temperaturas distintas. Esses gráficos são essenciais para a análise comparativa, pois permitem visualizar não apenas as diferenças absolutas entre as curvas modeladas e as obtidas com o extrator de curvas, mas também como elas se comportam em relação à variação dos parâmetros ambientais.

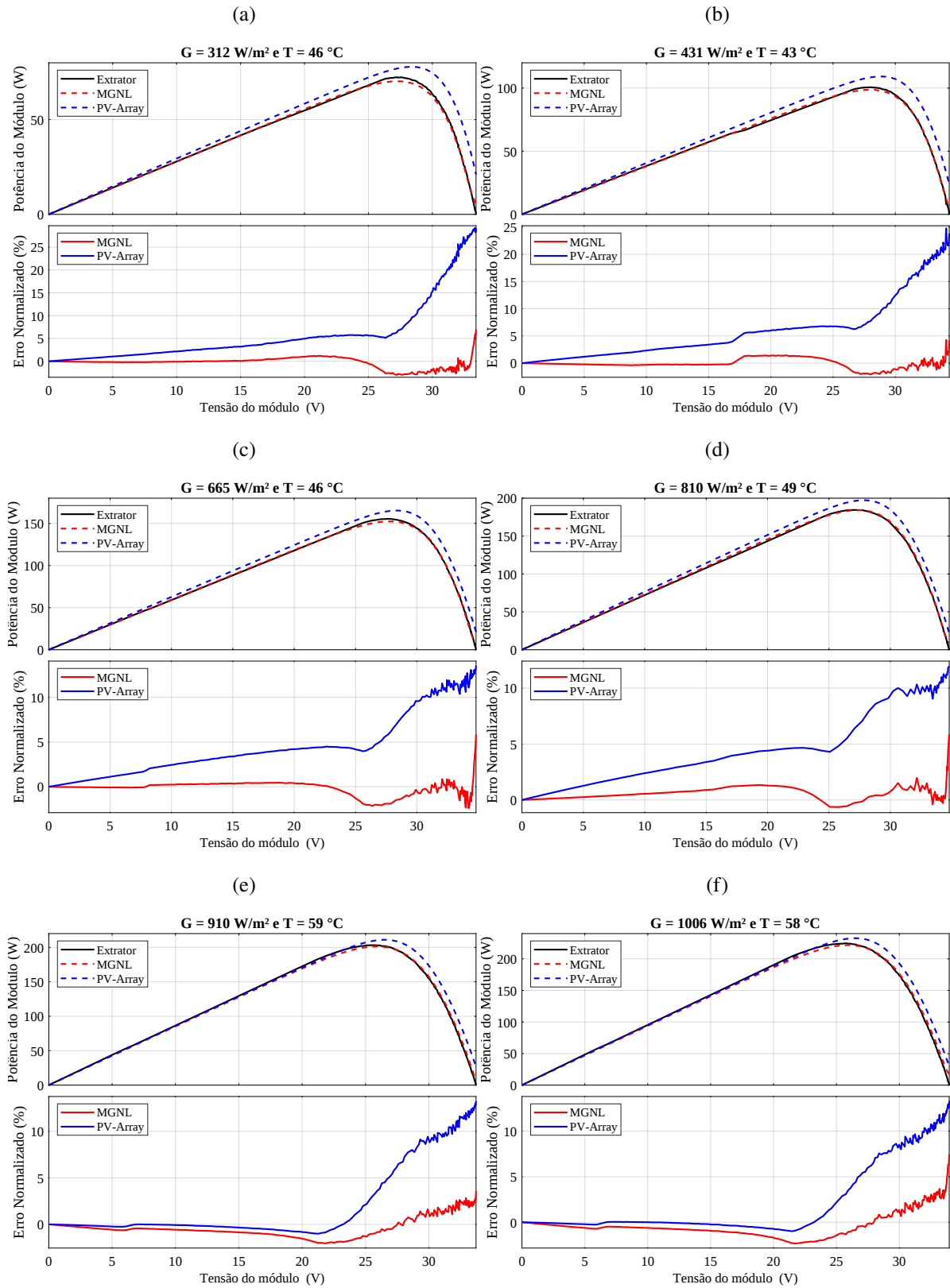
Assim, ao realizar essa comparação, é possível perceber a precisão do MGNL em diferentes cenários, bem como identificar possíveis melhorias ou ajustes necessários. Esse processo garante a confiabilidade dos resultados obtidos com o MGNL e valida sua aplicabilidade em detecção de falhas em sistemas FV.

Figura 58 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco *PV Array* com valores distintos de irradiância e temperatura para o módulo RSM-100P (P1).



Fonte: O Autor (2024).

Figura 59 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco *PV Array* com valores distintos de irradiância e temperatura para o módulo YL265C-30b.



Fonte: O Autor (2024).

Ao analisar os resultados obtidos para o módulo RSM-100P, Figura 58, fica evidente que o desempenho do MGNL foi significativamente superior aos resultados apresentados pelo bloco *PV Array*. Essa discrepância pode ser atribuída à forma como cada modelo é implementado: o MGNL utiliza resultados diretamente extraídos da curva específica do módulo, enquanto o bloco *PV Array* se baseia nos dados do *datasheet* fornecido pelo fabricante.

Essa diferença metodológica é essencial, pois qualquer discrepância entre os valores fornecidos pelo fabricante e os valores efetivamente medidos pode distorcer as curvas simuladas pelo bloco *PV Array*. É importante mencionar que o MGNL, ao utilizar dados diretamente extraídos das curvas I-V do módulo, tende a representar resultados mais precisos e condizentes com o comportamento real do módulo.

Logo, ressalta-se a importância de considerar a fonte dos dados ao utilizar modelos de simulação, como o MGNL e o bloco *PV Array*. Enquanto o MGNL se destaca pela sua precisão ao utilizar dados específicos do módulo, o bloco *PV Array* pode ser mais suscetível a erros devido a discrepâncias entre os dados do *datasheet* e a realidade do módulo.

Analisando os resultados para o módulo YL265C-30b, observa-se que os resultados obtidos estão aquém, assim como o RSM-100P, do esperado em comparação com o bloco *PV Array*. Uma possível explicação para esse fenômeno pode ser o simples desgaste do módulo, uma vez que foi fabricado em meados de 2014 e, apesar de armazenado em local protegido, pode ter sofrido deterioração de seus componentes ao longo do tempo.

Essa situação levanta uma reflexão importante sobre a avaliação das curvas I-V de módulos sujeitos à ação do tempo. À medida que um módulo é instalado e opera por vários anos, é natural esperar uma redução em sua capacidade produtiva. Portanto, é arriscado avaliar esse módulo utilizando apenas o bloco do MATLAB, que representa as condições de fábrica do módulo.

Uma abordagem alternativa seria avaliar o comportamento das curvas I-V do módulo ao longo dos anos e, assim, estudar a variação dos parâmetros do MGNL. Isso permitiria compreender como esses parâmetros se alteram com o tempo e fazer uma projeção mais precisa do desempenho do módulo ao longo de sua vida útil.

4.6 *PV ARRAY* AJUSTADO

Visando mitigar os impactos das informações distorcidas oriundas dos *datasheets* dos fabricantes, entende-se como uma alternativa plausível a adaptação dos parâmetros inseridos no bloco *PV Array*. Com base nos valores dos parâmetros obtidos com o MGNL, ajustam-se as propriedades do bloco, visando assim melhorar o funcionamento do mesmo e torná-lo mais aderente ao caso concreto.

Inicialmente, optou-se por substituir todos os parâmetros modificáveis pelos valores obtidos através do MGNL. A este ajuste foi atribuída a nomenclatura de opção 1. Porém, ao avaliar os resultados obtidos, notou-se que os métodos aplicados pelo bloco *PV Array* para translação de curvas podem ser incompatíveis com os valores de α e β obtidos através do MGNL.

Acredita-se que isso ocorra devido às diferenças nas equações utilizadas para a translação do V_{oc} e I_{sc} no bloco *PV Array* e no MGNL, conforme pode ser observado em (32) a (35), que já foram apresentadas em capítulos precedentes:

$$I_{sc} = I_{sc,ref} [1 + \alpha_{I_{sc}}(T - T_{ref})], \quad (32)$$

$$V_{oc} = V_{oc,ref} [1 + \beta_{V_{oc}}(T - T_{ref})], \quad (33)$$

$$I_{sc} = [I_{sc,ref} + \alpha_i(T - T_{ref})] \left(\frac{S}{S_{ref}} \right), \quad (34)$$

$$V_{oc} = V_{oc,ref} + \beta_T(T - T_{ref}) + \beta_S V_t \ln \left(\frac{S}{S_{ref}} \right), \quad (35)$$

destacando que (32) e (33) são originárias do MATLAB, enquanto (34) e (35) são as equações utilizadas por (SILVA, 2019).

Esse fato corrobora a ideia de não ajustar os valores de α e β no bloco *PV Array*. A este ajuste foi dado o nome de opção 2. Os resultados obtidos para ambos os ajustes podem ser vistos na Tabela 12. Para este caso, pode-se acarear os valores da média dos EMAPN e média dos NRMSD com os resultados obtidos com o MGNL, 0,78 % e 1,26 %, respectivamente.

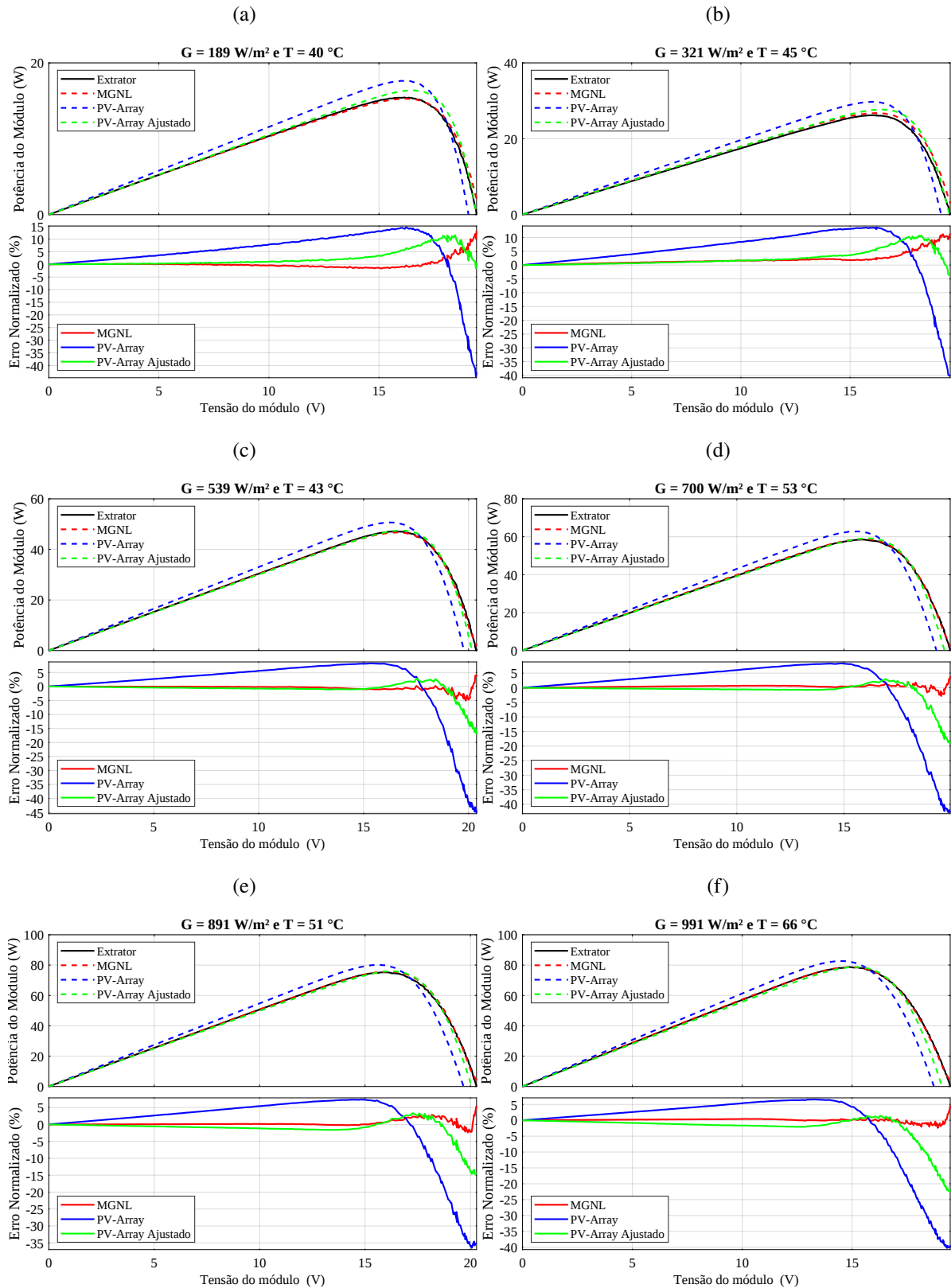
Tabela 12 – Avaliação dos parâmetros utilizados e resultados obtidos.

Parâmetros	PV-Array			MGNL
	Sem ajuste	Opção 1	Opção 2	
V_{oc}	21,58 V	21,9995 V	21,9995 V	21,9995 V
I_{sc}	6,04 A	5,5664 A	5,5664 A	5,5664 A
V_{mpp}	17,40 V	18,1356 V	18,1356 V	18,1356 V
I_{mpp}	5,75 A	5,2176 A	5,2176 A	5,2176 A
β	-0,32 %/°C	-0,2563 %/°C	-0,32 %/°C	-0,2563 %/°C
α	0,05 %/°C	0,1210 %/°C	0,05 %/°C	0,1210 %/°C
Média dos EMAPN	7,07 %	3,46 %	1,82 %	0,78 %
Média dos NRMSD	10,61 %	5,35 %	2,96 %	1,26 %

Fonte: O Autor (2024).

Desta forma, entendeu-se mais adequado utilizar a opção 2 como padrão para a avaliação do *PV Array* ajustado. Para uma análise mais visual das informações aqui apresentadas, consulta-se a Figura 60.

Figura 60 – Comparativo entre curvas P-V obtidas através do MGNL versus obtidas utilizando o bloco *PV Array* e *PV Array* ajustado com valores distintos de irradiância e temperatura para o módulo RSM-100P (P1).



Fonte: O Autor (2024).

Desprende-se, assim, que é adequada a adaptação dos valores de V_{oc} , I_{sc} , V_{mpp} e I_{mpp} para os valores obtidos através do MGNL, acarretando um funcionamento mais apropriado do bloco *PV Array*.

No entanto, isso também evidencia duas adversidades distintas. A primeira é a falta de confiabilidade dos *datasheets*, que podem não representar de maneira precisa os resultados esperados para os módulos reais. A segunda é a possibilidade de um eventual defeito de fabricação daquele módulo ou mesmo do lote ao qual ele pertence.

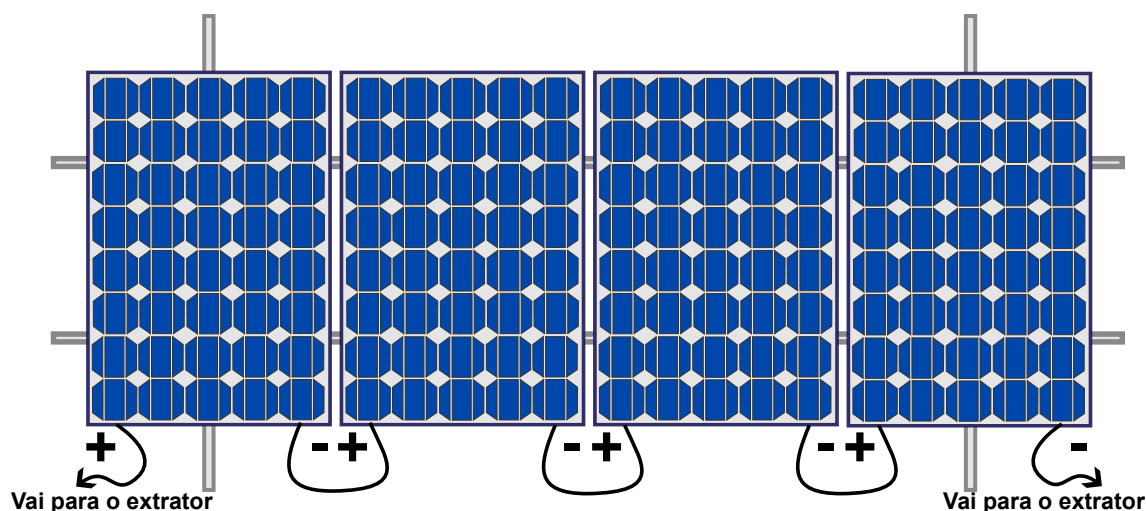
Neste contexto, entende-se que o ajuste dos parâmetros do bloco *PV Array* é desejável para uma maior aderência dos resultados esperados aos resultados medidos. Uma alternativa válida é a utilização de uma câmara solar, ajustada em STC, para parametrização inicial do módulo. No entanto, caso não se tenha à disposição uma câmara solar, pode-se seguir a metodologia apresentada nesta seção, que envolve a medição sob quaisquer condições, seguida da aplicação do MGNL e a translação para o STC.

Por fim, compreende-se que o *PV Array* ajustado, Figura 60, apresenta problemas ao observar o erro normalizado para diferentes valores de irradiância e temperatura. Isso sugere que os métodos de translação de curvas utilizados pelo bloco inserem incertezas no modelo.

4.7 AVALIAÇÃO COM ARRANJO DE QUATRO MÓDULOS

Para aprofundar a análise, é relevante avaliar um arranjo composto por quatro módulos RSM-100P em série (4x1), a fim de considerar eventuais problemas externos que são intrínsecos à avaliação de um único módulo isolado. Este tipo de análise pode fornecer uma compreensão mais abrangente do desempenho do sistema em condições reais. Assim, foi idealizada a montagem ilustrada na Figura 61, que mostra o esquemático da bancada experimental utilizada.

Figura 61 – Esquemático da bancada experimental para extração de curvas com arranjo de quatro módulos RSM-100P.



Fonte: O Autor (2024).

Para complementar a análise teórica, foi realizada uma montagem experimental *in loco*, conforme apresentado na Figura 62. Essa montagem permitiu a coleta de dados em condições reais de operação, considerando variáveis ambientais como irradiância e temperatura.

Figura 62 – Bancada experimental para extração de curvas com arranjo de quatro módulos RSM-100P.



Fonte: O Autor (2024).

A Figura 62 demonstra a montagem física do arranjo de quatro módulos. Uma das principais dificuldades encontradas foi a obtenção de medições precisas de temperatura e irradiância, devido à maior área de cobertura. Esse desafio reforça a necessidade de um tratamento minucioso dos dados coletados.

Com os dados obtidos pelo extrator de curva, foi possível realizar o tratamento necessário, conforme descrito nas seções anteriores deste trabalho. As curvas I-V obtidas foram selecionadas para constituir um banco de dados deste sistema. Na Tabela 13 são listadas as condições de irradiância e temperatura das curvas de referência e de treinamento utilizadas, enquanto a Figura 63 ilustra essas curvas, mostrando a variação da corrente em função da tensão para diferentes condições ambientais.

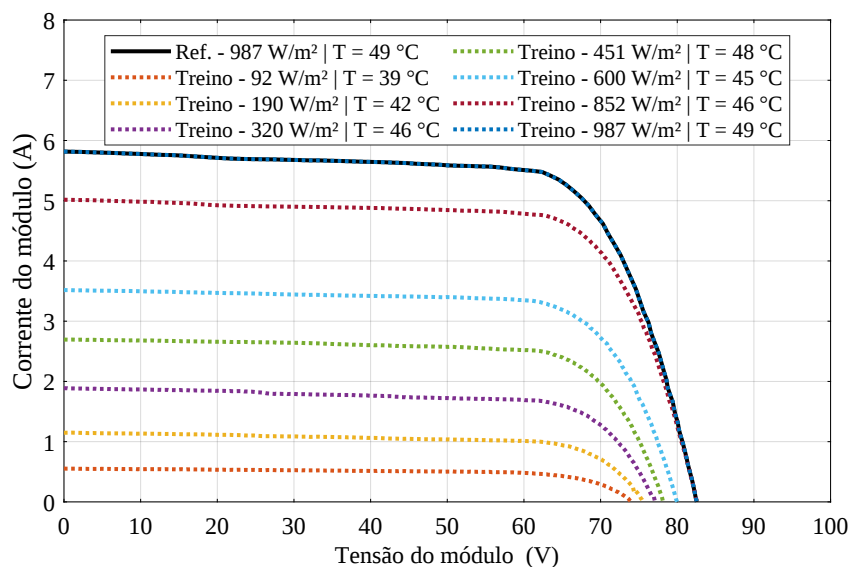
Tabela 13 – Curvas de referência e treinamento utilizadas no modelo de (SILVA, 2019) para o módulo RSM-100P (4x1).

Curvas de referência e treinamento para o MGNL	
Configuração 4x1	
$G(W/m^2)$	$T(^{\circ}C)$
987*	49*
92	39
190	42
320	46
451	48
600	45
852	46
987	49

* Valores de Referência

Fonte: O Autor (2024).

Figura 63 – Curvas I-V selecionadas como referência e como treinamento para configuração com quatro módulos em série (4x1).



Fonte: O Autor (2024).

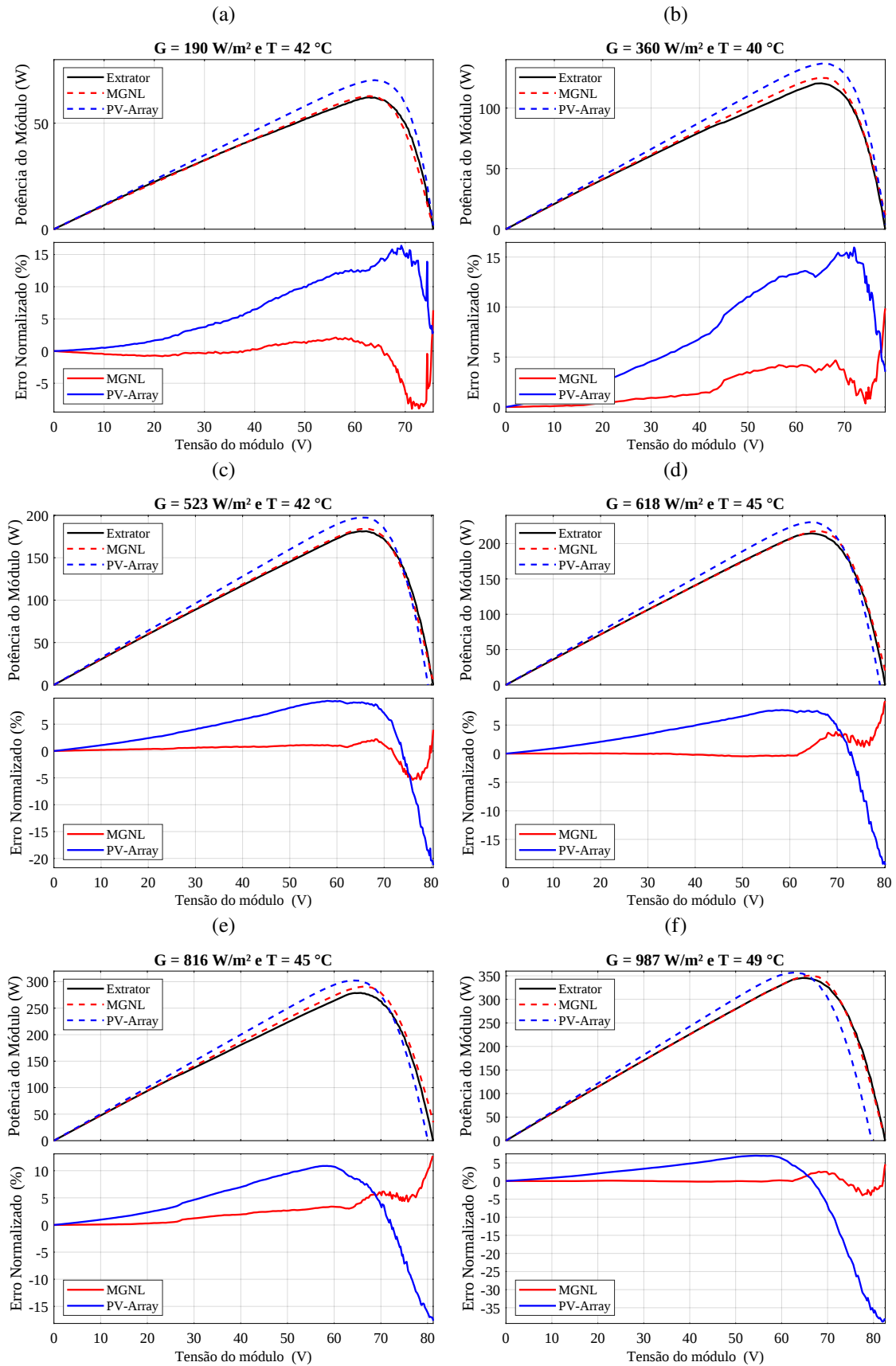
Utilizando essas curvas, foram obtidos os parâmetros do MGNL apresentados na Tabela 14. Esses parâmetros foram fundamentais para a obtenção do modelo e a subsequente comparação com o bloco *PV Array* do MATLAB/Simulink. Deve-se destacar que utilizando o bloco *PV Array* é possível utilizar uma comparação de arranjo de quatro módulos em série, enquanto no bloco do MGNL utilizou-se os parâmetros obtidos das medições dos quatro módulos juntos, ou seja, os parâmetros do modelo representam o conjunto de módulos.

Tabela 14 – Parâmetros obtidos a partir da configuração 4x1.

Parâmetros	Configuração 4x1
$R_{s,ref1}$	1,15 Ω
$R_{s,ref2}$	0,57 Ω
$R_{s,ref}$	0,57 Ω
$R_{sh,ref}$	234,13 Ω
k_{Rs}	14,31 $\%/^{\circ}C$
k_{Rsh}	-1,00 $\%/^{\circ}C$
$k_{v,T}$	-0,00 $V/^{\circ}C$
$k_{v,G}$	1,13
k_i	1,35 $mA/^{\circ}C$
γ_{Rs}	-1,07
γ_{Rsh}	-0,79
n_{ref}	0,96
$I_{g,ref}$	5,84 A
$I_{sat,ref}$	2,46 nA
Média dos EMAPNs	1,43 %
Média dos NRMSDs	2,44 %

Fonte: O Autor (2024).

Figura 64 – Comparativo entre curvas P-V obtidas através do MGNL e utilizando o bloco *PV Array* com valores distintos de irradiância e temperatura para quatro módulos RSM-100P (4x1).



Fonte: O Autor (2024).

Na Figura 64, são apresentados os comparativos entre as curvas P-V obtidas através do MGNL e do bloco *PV Array*, sob diferentes condições de irradiância e temperatura. Cada uma das Figuras de (a) a (f) corresponde a uma condição específica, demonstrando a precisão do modelo MGNL em reproduzir o comportamento real dos módulos. O erro normalizado (%) também é apresentado, evidenciando a acurácia do modelo em comparação com o bloco *PV Array*.

Outro aspecto analisado é o impacto da avaliação de uma montagem de múltiplos módulos, configurando um pequeno arranjo. Essa análise permite compreender melhor as variações de desempenho devido à interação entre os módulos e as condições ambientais variáveis.

Para tanto, pode-se inicialmente avaliar os valores de R_s e R_{sh} para cada um dos módulos individualmente e, posteriormente, para os quatro módulos em série, assim pode-se obter a Tabela 15. Destaca-se que há a necessidade de transpor os valores de R_s e R_{sh} para os valores em STC, ou seja, aplicar o MGNL e encontrar os valores de R_s e R_{sh} nas condições de STC, visando trazer a comparação para os mesmos critérios.

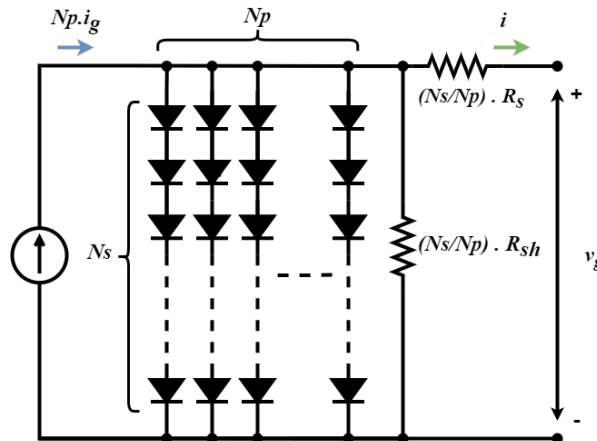
Tabela 15 – Avaliação de R_s e R_{sh} dos módulos isolados e em série.

Parâmetros	P1	P2	P3	P4	Arranjo	$\Sigma_1^4 R_n$
$R_{s,ref}(\Omega)$	0,29	0,23	0,29	0,27	0,57	-
$R_{sh,ref}(\Omega)$	247,13	115,04	98,73	168,43	234,13	-
$R_{s,stc}(\Omega)$	0,38	0,20	0,17	0,22	1,55	0,97
$R_{sh,stc}(\Omega)$	107,14	122,77	96,55	192,14	179,79	518,6
n_{ref}	1,12	1,33	1,07	1,17	0,96	-

Fonte: O Autor (2024).

Neste ponto, deve-se avaliar o comportamento esperado para a associação dos módulos. A Figura 65 apresenta o circuito equivalente de um arranjo. A conexão em série aumenta a tensão, enquanto a conexão em paralelo aumenta a corrente. O circuito mostra a combinação de R_s e R_{sh} , que afetam a performance do arranjo FV (NGUYEN; NGUYEN, 2015; TSAI et al., 2008; GOW; MANNING, 1996).

Figura 65 – Associação de módulos de acordo com o SDM.



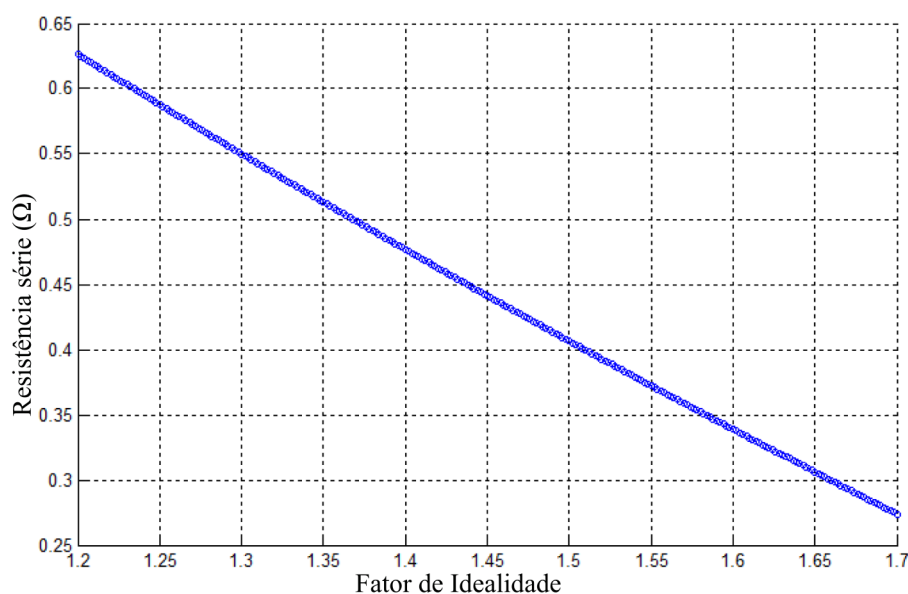
Fonte: Adaptado de (NGUYEN; NGUYEN, 2015).

No entanto, para o caso concreto, em que os valores de R_s e R_{sh} são distintos entre os módulos ensaiados e que a conexão dos módulos foi elaborada em série, espera-se que as resistências do arranjo sejam aproximadamente a combinação das resistências dos módulos individuais, fato este que não foi evidenciado no experimento prático, uma vez que a resistência série é $R_{s1,stc} + R_{s2,stc} + R_{s3,stc} + R_{s4,stc} = 0,97 \, \Omega$ e a resistência paralela é $R_{sh1,stc} + R_{sh2,stc} + R_{sh3,stc} + R_{sh4,stc} = 518,6 \, \Omega$.

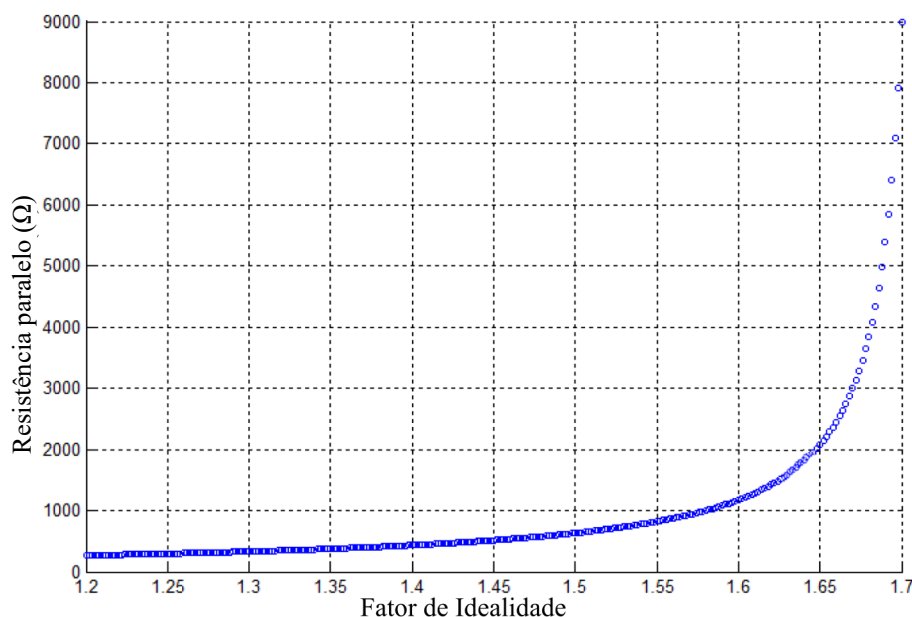
Neste ponto, deve-se mencionar as diferenças encontradas com relação ao n encontrado para as experimentações supracitadas. Assim, conforme apresentado em (HUSSEIN, 2017), a variação do n implica também na variação dos valores de R_s e R_{sh} . Em outras palavras, caso os valores do n sejam distintos, é impraticável fazer qualquer associação sem uma adaptação anterior.

Nas Figuras 66 e 67 são apresentadas as relações de R_s e R_{sh} para um módulo de silício multicristalino modelo BP SX-150 da BP Solar (HUSSEIN, 2017).

Figura 66 – Relação entre R_s e n .



Fonte: Adaptado de (HUSSEIN, 2017).

Figura 67 – Relação entre R_{sh} e n .

Fonte: Adaptado de (HUSSEIN, 2017).

Dito isto, entende-se que só seria coerente a associação de módulos com diferentes valores de n caso se normalizassem os valores e as resistências fossem transpostas para um ponto comum.

4.8 CONCLUSÃO

Desprende-se, deste capítulo que, para a utilização prática do MGNL, um banco de dados de curvas I-V robusto é fundamental, pois permite o acesso a um conjunto diversificado de dados experimentais, facilitando comparações, análises e validações de resultados. Além disso, a correta utilização do extrator de curvas é essencial para garantir a precisão e a confiabilidade das medições, evitando erros que poderiam comprometer a interpretação dos resultados.

Não obstante, o tratamento e a triagem adequados dos dados são etapas importantes, pois possibilitam a identificação e a correção de eventuais inconsistências nos dados experimentais, assegurando a qualidade e a integridade das informações obtidas. Além disso, a validação cuidadosa dos resultados obtidos pelo MGNL é imprescindível para garantir a confiabilidade das simulações realizadas, possibilitando a correta interpretação dos fenômenos físicos envolvidos.

Outro ponto de destaque é a possível adequação dos parâmetros do bloco *PV Array* como uma opção válida para a melhor adaptação do referido bloco ao caso concreto, trazendo assim resultados significativamente melhores.

Por fim, constata-se a dificuldade de associação de módulos em série, uma vez que esta associação, para ocorrer de maneira mais precisa, necessita de uma compensação pelos fatores de idealidade dos diodos, fato este que requer um estudo mais profundo e adequado.

5 CONCLUSÃO

Ao longo deste trabalho, foram exploradas em profundidade as complexidades e desafios associados à modelagem de sistemas fotovoltaicos utilizando o bloco *PV Array* do MATLAB/Simulink. A análise revelou que, embora essa ferramenta seja amplamente adotada devido à sua versatilidade e capacidade de simulação, apresenta limitações intrínsecas que podem comprometer a precisão dos resultados.

Além disso, a precisão depende da escolha correta dos parâmetros a serem utilizados, o que exige um conhecimento detalhado das características dos módulos fotovoltaicos. Deve-se destacar também a necessidade de confiabilidade nos dados do *datasheet*, uma vez que essas são as primeiras informações do módulo. Não obstante, a metodologia de translação de curvas aplicada neste bloco pode introduzir incertezas em diferentes condições de irradiância e temperatura, afetando a confiabilidade dos resultados obtidos.

Outra limitação identificada é a variação nas características dos módulos, mesmo quando do mesmo modelo e lote de produção, que apresentam parâmetros distintos. Essa variação pode resultar em discrepâncias significativas na modelagem, especialmente em sistemas de larga escala, onde pequenas diferenças podem se acumular e incorrer em erros substanciais.

Uma alternativa válida para reduzir esses problemas é a utilização de MGNL, que em contrapartida requer um banco de dados de curvas I-V e a aplicação de técnicas de tratamento e triagem de dados para garantir a qualidade das informações utilizadas.

A partir dos resultados obtidos pode-se perceber outra forma para mitigar as limitações do bloco *PV Array* com relação aos parâmetros iniciais aplicados a ele. Apresentando como uma alternativa viável a utilização dos valores obtidos pelo MGNL para a parametrização do bloco do MATLAB/Simulink, o que no texto foi chamado *PV Array* ajustado.

Não obstante, é válido apresentar algumas dificuldades encontradas ao longo do desenvolvimento deste trabalho. A imprecisão e falta de clareza das informações encontradas nos *datasheets*, em especial dos módulos RSM-100P e RS6E-155M, tornaram o desenvolvimento deste trabalho mais difícil; como exemplo, menciona-se a quantidade e a forma de ligação dos diodos de passagem no RSM-100P, que na prática eram destoantes do *datasheet*. Iniciou-se, assim, uma abordagem mais crítica com relação às informações do *datasheet*.

Além disso, pela natureza da proposta do trabalho, a prerrogativa de fazer as medições em ambiente não controlado dificultou a análise dos dados extraídos, uma vez que o ambiente externo pode induzir erros de medição, como, por exemplo, o atraso entre a medição da curva I-V e dos dados do piranômetro. Assim, para mitigar esses impactos, foi necessário estudar formas de tratar e selecionar as curvas I-V extraídas.

Ademais, entende-se que os resultados obtidos neste trabalho demonstram uma latente necessidade de melhorar o bloco *PV Array*, para torná-lo mais preciso, além de possibilitar outras

formas de ajuste dos parâmetros iniciais. Além disso, neste trabalho adotou-se uma abordagem mais prática para os MGNL, substituindo os bancos de dados de curvas I-V por curvas extraídas diretamente de um extrator de curvas.

Em suma, este trabalho sublinha a necessidade de uma abordagem crítica ao utilizar o bloco *PV Array* do MATLAB/Simulink para modelagem de sistemas fotovoltaicos, além de apresentar alternativas para o uso desta ferramenta, vislumbrando reconhecer e reduzir suas limitações.

Este capítulo, além de compreender uma breve conclusão deste trabalho, também apresenta-se aqui uma lista de sugestões de trabalhos futuros que podem ser desenvolvidos a partir das pesquisas e discussões aventadas no texto, bem como a lista de artigos publicados e em processo de publicação durante este mestrado.

5.1 SUGESTÃO DE TRABALHOS FUTUROS

- I. Avaliação estatística das curvas obtidas para diferentes pontos de operação do sistema. É interessante observar as limitações dos módulos e dos instrumentos de medição, a fim de investigar em quais condições há uma maior ocorrência de inconsistências;
- II. Aprimorar a metodologia de tratamento e triagem das curvas I-V, e avaliar a aplicabilidade com outras tecnologias de módulos;
- III. Desenvolver tecnologia dinâmica de avaliação dos parâmetros do módulo, visando assim a possibilidade de acompanhar o comportamento dos parâmetros ao longo do tempo;
- IV. Avaliar de maneira mais profunda os impactos da associação de módulos que possuem parâmetros exatamente iguais, como fator de idealidade, e investigar os impactos desta associação nos demais parâmetros do MGNL;
- V. Implementar os experimentos apresentados em ambiente controlado, utilizando um simulador solar, visando assim uma validação experimental mais precisa;
- VI. Desenvolver uma aplicação e um bloco no ambiente MATLAB/Simulink que possibilite a utilização do MGNL a partir dos dados obtidos por um extrator de curvas I-V, sem necessidade de *scripts* complexos e possibilitando o ajuste dos filtros e tratamentos de curvas apresentados neste trabalho. Disponibilizar este novo bloco em fóruns específicos, como fórum do *Mathworks* e *Github*;
- VII. Avaliar a diferença da eficácia na utilização de técnicas de detecção de degradações em potência distintas com o bloco *PV Array* e MGNL.

5.2 ARTIGOS PUBLICADOS NO TEMA DO MESTRADO

V. M. Cavalcante, T. A. Fernandes, **R. A. Freitas**, F. Bradaschia, M. C. Cavalcanti and L. R. Limongi, "A Global Nonlinear Model for Photovoltaic Modules Based on 3-D Surface Fitting,"in IEEE Journal of Photovoltaics, doi: 10.1109/JPHOTOV.2024.3414115.

V. M. Cavalcante, T. A. Fernandes, **R. A. Freitas**, N. A. de M. S. Amâncio, F. Bradaschia and M. C. Cavalcanti, "Impact of Optimization Algorithm Choice on Nonlinear Global Model for Energy Generation Forecasting,"SOBRAEP Open Journal of Power Electronics, doi: doi.org/10.18618/REP.2005.1.063070.

V. M. Cavalcante, T. A. Fernandes, **R. A. Freitas**, N. A. de M. S. Amâncio, F. Bradaschia and M. C. Cavalcanti, "Effect of Optimization Algorithm Choice on Photovoltaic Module Modeling Performance,"2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP), Florianopolis, Brazil, 2023, pp. 1-8, doi: 10.1109/SPEC56436.2023.10407366.

V. M. C. Junior, T. A. Fernandes, **R. A. Freitas**, N. A. De M. S. Amâncio, F. Bradaschia and M. C. Cavalcanti, "Impact of Post-Nrel Database Technologies for Photovoltaic Modules in NREL-Based Mathematical Models,"2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP), Florianopolis, Brazil, 2023, pp. 1-8, doi: 10.1109/SPEC56436.2023.10407484.

REFERÊNCIAS

- ALI, M. H. et al. Real time fault detection in photovoltaic systems. *Energy Procedia*, v. 111, p. 914–923, 2017. ISSN 1876-6102. 8th International Conference on Sustainability in Energy and Buildings, SEB-16, 11-13 September 2016, Turin, Italy. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1876610217302874>>. Citado na página 42.
- BANSAL, N.; JAISWAL, S. P.; SINGH, G. Comparative investigation of performance evaluation, degradation causes, impact and corrective measures for ground mount and rooftop solar pv plants – a review. *Sustainable Energy Technologies and Assessments*, v. 47, p. 101526, 2021. ISSN 2213-1388. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2213138821005373>>. Citado na página 38.
- BARBOSA, E. J. *TÉCNICAS DE SEGUIMENTO DO PONTO DE MÁXIMA POTÊNCIA PARA SISTEMAS FOTOVOLTAICOS BASEADAS NOS MODELOS GLOBAIS NÃO LINEARES DE UM E DOIS DIODOS*. Tese (Tese) — Universidade Federal de Pernambuco, junho 2023. Citado 2 vezes nas páginas 44 e 67.
- BLAABJERG, F. *Control of Power Electronic Converters and Systems*. 1. ed. [S.l.]: Academic Press, 2018. v. 1. (Volume 1, v. 1). ISBN 9780128052457,0128052457. Citado 3 vezes nas páginas 27, 28 e 29.
- BLAABJERG, F. *Control of power electronic converters and systems. Volume 2*. 1st ed.. ed. [S.l.]: Academic Press, 2018. ISBN 9780128161685,012816168X. Citado 2 vezes nas páginas 28 e 33.
- Canadian Solar. *Datasheet: KuDymond CS3U-P-AG*. 2020. Accessed: 2024-05-25. Disponível em: <https://www.canadiansolar.com/wp-content/uploads/2020/05/Canadian_Solar-Datasheet-KuDymond_CS3U-P-AG_EN.pdf>. Citado na página 30.
- CAVALCANTE, V. M. et al. Effect of optimization algorithm choice on photovoltaic module modeling performance. In: *2023 IEEE 8th Southern Power Electronics Conference and 17th Brazilian Power Electronics Conference (SPEC/COBEP)*. [S.l.: s.n.], 2023. p. 1–8. Citado na página 82.
- CHOUDER, A. et al. Modeling and simulation of a grid connected pv system based on the evaluation of main pv module parameters. *Simulation Modelling Practice and Theory*, v. 20, n. 1, p. 46–58, 2012. ISSN 1569-190X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1569190X11001456>>. Citado na página 49.
- de Blas, M. et al. Selecting a suitable model for characterizing photovoltaic devices. *Renewable Energy*, v. 25, n. 3, p. 371–380, 2002. ISSN 0960-1481. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0960148101000568>>. Citado 2 vezes nas páginas 27 e 38.
- de SOTO, W. et al. Improvement and validation of a model for photovoltaic array performance. *Solar Energy*, v. 80, n. 1, p. 78–88, 2006. ISSN 0038-092X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0038092X05002410>>. Citado 8 vezes nas páginas 9, 10, 49, 51, 53, 54, 55 e 57.

DHOUNDIYAL, P. et al. Fault detection and classification in solar photovoltaic array. In: *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)*. [S.l.: s.n.], 2022. p. 1–6. Citado na página 41.

DURAN, E. et al. Different methods to obtain the i–v curve of pv modules: A review. In: *2008 33rd IEEE Photovoltaic Specialists Conference*. [S.l.: s.n.], 2008. p. 1–6. Citado na página 69.

FERNANDES, T. A. *ANÁLISE DE MODELOS GLOBAIS NÃO LINEARES PARA MÓDULOS FOTOVOLTAICOS DE TECNOLOGIAS CIGS, HIT, ASI-MICRO E ASI-TANDEM*. Tese (Trabalho de Conclusão de Curso) — Universidade Federal de Pernambuco, fevereiro 2024. Citado 3 vezes nas páginas 44, 80 e 81.

First Solar. *First Solar - FS-4122A-3*. 2018. Disponível em: <<https://www.firstsolar.com/-/media/First-Solar/Technical-Documents/Series-4-Datasheets/Series-4V3-Module-Datasheet.ashx>>. Citado na página 56.

GHOLAMI, A. et al. Step-by-step guide to model photovoltaic panels: An up-to-date comparative review study. *IEEE Journal of Photovoltaics*, v. 12, n. 4, p. 915–928, 2022. Citado 2 vezes nas páginas 33 e 42.

GLOVER, J. D. et al. *Power System Analysis and Design*. [S.l.]: Cengage Learning, 2011. Citado na página 41.

GOW, J.; MANNING, C. Development of a model for photovoltaic arrays suitable for use in simulation studies of solar energy conversion systems. In: *1996 Sixth International Conference on Power Electronics and Variable Speed Drives (Conf. Publ. No. 429)*. [S.l.: s.n.], 1996. p. 69–74. Citado na página 99.

GREEN, M. *Solar Cells: Operating Principles, Technology, and System Applications*. [S.l.]: Prentice-Hall, 1982. (Prentice-Hall Contemporary Comparative Politics Series). ISBN 9780138222703. Citado 3 vezes nas páginas 27, 28 e 33.

HAJJI, M. et al. Fault detection and diagnosis in grid-connected pv systems under irradiance variations. *Energy Reports*, v. 9, p. 4005–4017, 2023. ISSN 2352-4847. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S235248472300272X>>. Citado na página 24.

HARROU, F. et al. Reliable fault detection and diagnosis of photovoltaic systems based on statistical monitoring approaches. *Renewable Energy*, v. 116, p. 22–37, 2018. ISSN 0960-1481. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0960148117309114>>. Citado na página 46.

HASSAN, A. et al. An improved genetic algorithm based fractional open circuit voltage mppt for solar pv systems. *Energy Reports*, v. 9, p. 1535–1548, 2023. ISSN 2352-4847. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352484722026762>>. Citado na página 24.

HUANG, Y.-C. et al. Optimization of module parameters for pv power estimation using a hybrid algorithm. *IEEE Transactions on Sustainable Energy*, v. 11, n. 4, p. 2210–2219, 2020. Citado na página 49.

HUMADA, A. M. et al. Solar cell parameters extraction based on single and double-diode models: A review. *Renewable and Sustainable Energy Reviews*, v. 56, p. 494–509, 2016. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032115013180>>. Citado na página 37.

HUSSEIN, A. A simple approach to extract the unknown parameters of pv modules. *TURKISH JOURNAL OF ELECTRICAL ENGINEERING COMPUTER SCIENCES*, v. 25, p. 4431–4444, 01 2017. Citado 2 vezes nas páginas 100 e 101.

IEC. *IEC 60891:2021 - Photovoltaic devices - Procedures for temperature and irradiance corrections to measured I-V characteristics*. International Electrotechnical Commission, 2021. Accessed: 2024-07-18. Disponível em: <<https://webstore.iec.ch/en/publication/61766>>. Citado na página 51.

International Energy Agency. *World Energy Outlook 2020*. 2021. Disponível em: <https://iea-pvps.org/wp-content/uploads/2021/10/Final-Report-IEA-PVPS-T13-19_2021_PV-Failure-Monitoring.pdf>. Citado na página 41.

International Energy Agency. *World Energy Outlook 2020*. 2023. Disponível em: <<https://iea.blob.core.windows.net/assets/86ede39e-4436-42d7-ba2a-edf61467e070/WorldEnergyOutlook2023.pdf>>. Citado 3 vezes nas páginas 21, 22 e 39.

International Renewable Energy Agency. *Renewable Energy and Jobs – Annual Review 2020*. 2020. Disponível em: <<https://www.irena.org/publications/2020/Sep/Renewable-Energy-and-Jobs-Annual-Review-2020>>. Citado na página 22.

JAN, A. A. et al. Solar photovoltaic technology and its impact on environmental, social and governance (esg) performance: A review. *KnE Social Sciences*, 12 2023. Citado na página 21.

JORDAN, D. C. et al. Compendium of photovoltaic degradation rates. *Progress in Photovoltaics: Research and Applications*, v. 24, p. 978–989, 7 2016. ISSN 1062-7995. Citado 2 vezes nas páginas 39 e 41.

JUNIOR, A. J. N. *CONTRIBUIÇÃO À MODELAGEM DE MÓDULOS FOTOVOLTAICOS E PROPOSTA DE UMA TÉCNICA HÍBRIDA DE SEGUIMENTO DO PONTO DE MÁXIMA POTÊNCIA COM ESTIMADOR DE IRRADIÂNCIA*. Tese (Tese) — Universidade Federal de Pernambuco, agosto 2018. Citado 4 vezes nas páginas 35, 36, 42 e 53.

JUNIOR, V. M. C. *ANÁLISE COMPARATIVA DE MODELOS GLOBAIS NÃO-LINEARES PARA MÓDULOS FOTOVOLTAICOS*. Tese (Dissertação) — Universidade Federal de Pernambuco, fevereiro 2023. Citado 6 vezes nas páginas 42, 43, 44, 52, 66 e 79.

JUNIOR, V. M. C. et al. Avaliação estática e dinâmica de fontes emuladoras solares comerciais e seus impactos em estudos de mppt. *Eletrônica de Potência*, v. 29, p. e202403–e202403, 2024. Citado na página 69.

KOESTER, L. et al. Review of photovoltaic module degradation, field inspection techniques and techno-economic assessment. *Renewable and Sustainable Energy Reviews*, v. 165, p. 112616, 2022. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S136403212200510X>>. Citado na página 39.

KÖNTGES, M. et al. Review of failures of photovoltaic modules. In: _____. *IEA PVPS Task13: Performance and Reliability of Photovoltaic Systems: Subtask 3.2*. [S.l.: s.n.], 2014. p. 1–132. Citado na página 39.

KURUKURU, V. S. B. et al. A novel fault classification approach for photovoltaic systems. *Energies*, v. 13, n. 2, 2020. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/13/2/308>>. Citado na página 41.

LANG, Z.; ZHANG, Y. Parameter identification and performance estimation for pv modules based on reduced forms model. *Journal of Renewable and Sustainable Energy*, v. 12, p. 053703, 09 2020. Citado na página 66.

LINDIG, S. et al. Review of statistical and analytical degradation models for photovoltaic modules and systems as well as related improvements. *IEEE Journal of Photovoltaics*, v. 8, n. 6, p. 1773–1786, 2018. Citado na página 39.

LIU, L. qun et al. Solar energy development in china—a review. *Renewable and Sustainable Energy Reviews*, v. 14, n. 1, p. 301–311, 2010. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032109002068>>. Citado na página 21.

LUQUE, A.; HEGEDUS, S. (Ed.). *Handbook of Photovoltaic Science and Engineering*. Wiley, 2011. Disponível em: <<https://onlinelibrary.wiley.com/doi/book/10.1002/9780470974704>>. Citado na página 27.

MAHMOUD, Y. A.; XIAO, W.; ZEINELDIN, H. H. A parameterization approach for enhancing pv model accuracy. *IEEE Transactions on Industrial Electronics*, v. 60, n. 12, p. 5708–5716, 2013. Citado na página 45.

MATAM, M. Pv array iv curves: Why we see them the way we do? In: *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*. [S.l.: s.n.], 2020. p. 1578–1580. Citado na página 29.

MATHWORKS. *MathWorks*. 2024. Disponível em: <<https://www.mathworks.com/>>. Citado 8 vezes nas páginas 46, 47, 48, 50, 51, 56, 61 e 63.

MEMON, S. A. et al. A machine-learning-based robust classification method for pv panel faults. *Sensors*, v. 22, n. 21, 2022. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/22/21/8515>>. Citado na página 41.

MERIBOUT, M. et al. Solar panel inspection techniques and prospects. *Measurement*, v. 209, p. 112466, 2023. ISSN 0263-2241. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0263224123000301>>. Citado na página 41.

NDIAYE, A. et al. Degradations of silicon photovoltaic modules: A literature review. *Solar Energy*, v. 96, p. 140–151, 2013. ISSN 0038-092X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0038092X13002703>>. Citado na página 41.

NELSON, J. *The Physics of Solar Cells: Photons In, Electrons Out*. [S.l.]: Imperial College Press, 2003. (Properties of Semiconductor Materials). ISBN 1860943403,9781860943409. Citado na página 27.

NGUYEN, X. H.; NGUYEN, M. P. Mathematical modeling of photovoltaic cell/module/arrays with tags in matlab/simulink. *Environmental Systems Research*, Springer, v. 4, n. 24, p. 1–13, 2015. Disponível em: <<http://dx.doi.org/10.1186/s40068-015-0047-9>>. Citado na página 99.

PARIDA, B.; INIYAN, S.; GOIC, R. A review of solar photovoltaic technologies. *Renewable and Sustainable Energy Reviews*, v. 15, n. 3, p. 1625–1636, 2011. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032110004016>>. Citado na página 33.

PAVAN, A. M. et al. A study on the mismatch effect due to the use of different photovoltaic modules classes in large-scale solar parks. *Progress in Photovoltaics: Research and Applications*, v. 22, 03 2014. Citado na página 62.

PILIOUGINE et al. Comparative of iec 60891 and other procedures for temperature and irradiance corrections to measured i–v characteristics of photovoltaic devices. *Energies*, v. 17, p. 566, 01 2024. Citado na página 51.

PILLAI, D. S. et al. A comprehensive review on protection challenges and fault diagnosis in pv systems. *Renewable and Sustainable Energy Reviews*, v. 91, p. 18–40, 2018. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032118301758>>. Citado 5 vezes nas páginas 9, 24, 39, 40 e 41.

RESUN. *Resun Solar Module RSM-100P POLY 100W*. 2024. Disponível em: <<https://www.resunsolar.com/wp-content/uploads/download/RSM-100P%20POLY%20100W.pdf>>. Citado 3 vezes nas páginas 58, 59 e 86.

SCHRODER, D. K. *Semiconductor Material and Device Characterization*. John Wiley Sons, 1998. Disponível em: <<https://books.google.com/books?id=ARxRAAAAMAAJ>>. Citado na página 28.

SILVA, E. *Proposta de um modelo global não-linear para caracterização do comportamento elétrico de células fotovoltaicas*. Tese (Tese) — Universidade Federal de Pernambuco, abril 2019. Citado 16 vezes nas páginas 13, 19, 24, 28, 30, 33, 34, 35, 42, 43, 44, 79, 80, 87, 93 e 96.

SILVA, E. A. et al. An eight-parameter adaptive model for the single diode equivalent circuit based on the photovoltaic module's physics. *IEEE Journal of Photovoltaics*, v. 7, n. 4, p. 1115–1123, July 2017. ISSN 2156-3403. Citado 4 vezes nas páginas 13, 66, 80 e 81.

SINGH, P. et al. Temperature dependence of solar cell performance—an analysis. *Solar Energy Materials and Solar Cells*, v. 101, p. 36–45, 2012. ISSN 0927-0248. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0927024812000931>>. Citado na página 53.

SOLMETRIC. *Solmetric PVA Analyzer Datasheet*. 2016. Disponível em: <https://live-solmetric.pantheonsite.io/wp-content/uploads/2023/06/PVA-1500HE_Datasheet_6_web.pdf>. Citado na página 68.

TAHIRI, F. E. et al. Modeling and performance analysis of a solar pv power system under irradiation and load variations. In: *2017 14th International Multi-Conference on Systems, Signals Devices (SSD)*. [S.l.: s.n.], 2017. p. 234–238. Citado na página 22.

TRETER, M. E. *MONITORAMENTO E AVALIAÇÃO DE DESEMPENHO INTEGRADOS À USINAS FOTOVOLTAICAS*. Tese (Tese) — Universidade do Estado de Santa Catarina, março 2023. Citado 2 vezes nas páginas 67 e 68.

TRETER, M. E. et al. Métodos de aquisição experimental de curvas i-v de arranjos fotovoltaicos: Uma revisão. *Proceedings of the 11th Seminar on Power Electronics and Control (SEPOC 2018)*, 2018. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/30442/submission-91601-700301-field_submission_abstract_file2.pdf?sequence=1&isAllowed=y>. Citado na página 68.

TSAI, H.-L. et al. Development of generalized photovoltaic model using matlab/simulink. In: SAN FRANCISCO, USA. *Proceedings of the world congress on Engineering and computer science*. [S.l.], 2008. v. 2008, p. 1–6. Citado na página 99.

VARSHNI, Y. Temperature dependence of the energy gap in semiconductors. *Physica*, v. 34, n. 1, p. 149–154, 1967. ISSN 0031-8914. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0031891467900626>>. Citado 2 vezes nas páginas 53 e 54.

WADIE, F. Analysis of the integrated effect of temporary overvoltages, pv transformer connection and overcurrent protection in hybrid pv-wind energy system. *Electric Power Systems Research*, v. 216, p. 109084, 2023. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378779622011336>>. Citado na página 24.

WANG, M. et al. An online optimization method for extracting parameters of multi-parameter pv module model based on adaptive levenberg-marquardt algorithm. *Energy Conversion and Management*, v. 245, p. 114611, 2021. ISSN 0196-8904. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0196890421007871>>. Citado 4 vezes nas páginas 42, 49, 52 e 66.

WOLF, M.; RAUSCHENBACH, H. Series resistance effects on solar cell measurements. *Advanced Energy Conversion*, v. 3, n. 2, p. 455–479, 1963. ISSN 0365-1789. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0365178963900638>>. Citado na página 37.

WURFEL, P. *Physics of Solar Cells - From Principles to New Concepts*. Springer, 2005. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=6ad57c3ebefcfbc5fddb0f638671af4e>>. Citado na página 28.

YINGLI. *YINGLI - YL265D-30b*. 2016. Disponível em: <http://d9no22y7yqre8.cloudfront.net/assets/uploads/products/downloads/DS_YLM60Cell-30b_40mm_US_EN_20160229_V04.pdf>. Citado na página 60.

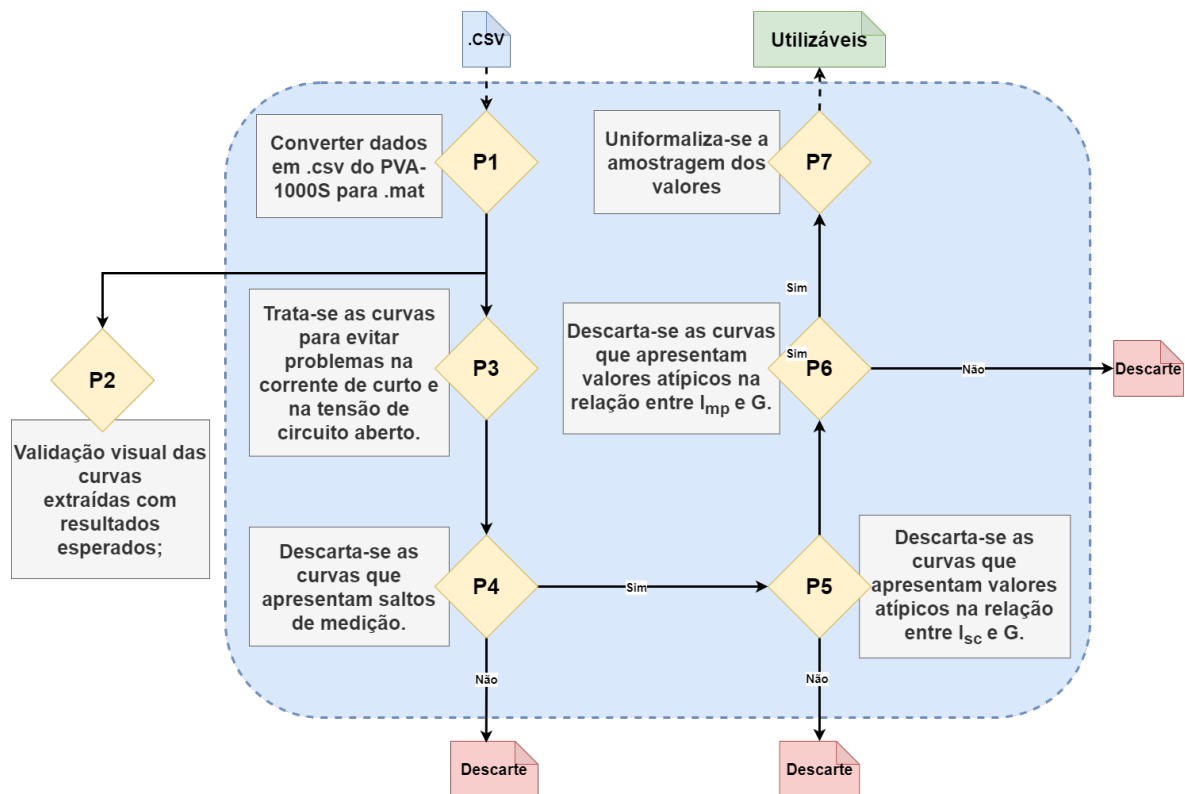
ZHANG, Y. et al. Modelling and estimating performance for pv module under varying operating conditions independent of reference condition. *Applied Energy*, v. 310, p. 118527, 2022. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306261922000162>>. Citado 3 vezes nas páginas 42, 49 e 66.

ZHANG, Y. et al. A novel method for performance estimation of photovoltaic module without setting reference condition. *International Journal of Electrical Power Energy Systems*, v. 134, p. 107439, 2022. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061521006785>>. Citado 2 vezes nas páginas 42 e 52.

APÊNDICE

Com o intuito de facilitar a compreensão dos algoritmos apresentados neste apêndice pode-se apresentar o fluxograma da Figura 68. Em que é apresentada de maneira simplificada a funcionalidade de cada uma das etapas, de P1 a P7, destacando que o P0 é apenas a função que executa os demais algoritmos.

Figura 68 – Fluxograma proposto para tratamento e filtro de curvas I-V a serem utilizadas como banco de dados do MGNL conforme organização dos algoritmos.



Fonte: O Autor (2024).

Apêndice 1 - P0 - Algoritmo Raiz

```

1  clc
2  clear all
3
4  % Define o modelo do painel solar
5  modelo = 'YL265C30b'
6
7  %% Cria as pastas
8
9  % Cria a pasta para armazenar os dados brutos
10 mkdir('01) Bruta');
11
12 % Cria a pasta para armazenar os dados de validacao
13 mkdir('02) Validacao');

```

```

14
15 % Cria a pasta para armazenar os dados tratados
16 mkdir('03) Tratada');
17
18 % Cria a subpasta para armazenar os PDFs dos dados tratados
19 mkdir('03) Tratada/PDF');
20
21 % Cria a pasta para armazenar os dados filtrados pelo Filtro 1
22 mkdir('04) Filtrada - Filtro 1');
23
24 % Cria a subpasta para armazenar os PDFs dos dados filtrados pelo Filtro 1
25 mkdir('04) Filtrada - Filtro 1/PDF');
26
27 % Cria a pasta para armazenar os dados descartados pelo Filtro 1
28 mkdir('05) Descartada - Filtro 1');
29
30 % Cria a subpasta para armazenar os PDFs dos dados descartados pelo Filtro 1
31 mkdir('05) Descartada - Filtro 1/PDF');
32
33 % Cria a pasta para armazenar os dados filtrados pelo Filtro 2
34 mkdir('06) Filtrada - Filtro 2');
35
36 % Cria a subpasta para armazenar os PDFs dos dados filtrados pelo Filtro 2
37 mkdir('06) Filtrada - Filtro 2/PDF');
38
39 % Cria a pasta para armazenar os dados descartados pelo Filtro 2
40 mkdir('07) Descartada - Filtro 2');
41
42 % Cria a subpasta para armazenar os PDFs dos dados descartados pelo Filtro 2
43 mkdir('07) Descartada - Filtro 2/PDF');
44
45 % Cria a pasta para armazenar os dados filtrados pelo Filtro 3
46 mkdir('08) Filtrada - Filtro 3');
47
48 % Cria a subpasta para armazenar os PDFs dos dados filtrados pelo Filtro 3
49 mkdir('08) Filtrada - Filtro 3/PDF');
50
51 % Cria a pasta para armazenar os dados descartados pelo Filtro 3
52 mkdir('09) Descartada - Filtro 3');
53
54 % Cria a subpasta para armazenar os PDFs dos dados descartados pelo Filtro 3
55 mkdir('09) Descartada - Filtro 3/PDF');
56
57 % Cria a pasta para armazenar os dados tratados finais
58 mkdir('10) Tratada');
59
60 % Cria a subpasta para armazenar os PDFs dos dados tratados finais
61 mkdir('10) Tratada/PDF');
62
63 %% Executa as Etapas;
64
65 % Executa o script que converte arquivos CSV para formato MAT
66 run("P1_csv2mat_YL265C30b.m")
67
68 % Comentado: Executa o script que valida as curvas I-V
69 run("P2_val_IV_YL265C30b.m")
70
71 % Executa o script que realiza o tratamento inicial dos dados
72 run("P3_tratamento_YL265C30b.m")
73

```

```

74 % Executa o script que aplica o primeiro filtro aos dados
75 run("P4_filtro_1_YL265C30b.m")
76
77 % Executa o script que aplica o segundo filtro aos dados
78 run("P5_filtro_2_YL265C30b.m")
79
80 % Executa o script que aplica o terceiro filtro aos dados
81 run("P6_filtro_3_YL265C30b.m")
82
83 % Executa o script que realiza o tratamento final dos dados
84 run("P7_tratamento_YL265C30b.m")

```

Apendice 2 - P1

```

1  clear all
2  clc
3
4  %% Pegar o nome da curva
5
6  % Define o local dos arquivos CSV
7  local = fullfile('.', '00) CSV', '*.csv');
8  % Obtem a lista de arquivos no diretorio especificado
9  files = dir(local);
10 % Conta o numero de arquivos encontrados
11 filesN = length(files);
12 % Define o modelo do painel solar
13 modelo = 'RSM100P';
14
15 %% Loop para processar cada arquivo
16
17 % Loop sobre cada arquivo encontrado
18 for k = 1:1:filesN
19     file = files(k);
20     % Obtem o nome do arquivo
21     nomearq = file.name;
22     % Obtem o caminho do diretorio do arquivo
23     caminho = file.folder;
24     % Constroi o caminho completo do arquivo
25     curva_caminho = fullfile(caminho,nomearq);
26     % Importa os dados do arquivo CSV
27     Curva_Crua = importfile(curva_caminho);
28
29     % Extrai as medicoes de temperatura, irradiancia, corrente de curto-circuito e tensao de
        circuito aberto
30     T_med = Curva_Crua(22,2);
31     Ir_med = Curva_Crua(21,2);
32     Isc_med = Curva_Crua(18,2);
33     Voc_med = Curva_Crua(17,2);
34
35     % Inicializa variaveis para armazenar as correntes e tensoes
36     size = length(Curva_Crua);
37     offset = 49;
38     ii = 0;
39     for i = offset:1:size
40         ii = ii+1;
41         I(ii) = Curva_Crua(i,2);
42         V(ii) = Curva_Crua(i,1);

```

```

43     end
44
45     % Filtra os valores validos de corrente
46     Nfiltro = length(I);
47     for i = 1:1:Nfiltro
48         if ~isnan(I(i))
49             I_temp(i) = I(i);
50         else
51             break
52         end
53     end
54     I_curva = I_temp;
55
56     % Filtra os valores validos de tensao
57     Nfiltro = length(V);
58     for i = 1:1:Nfiltro
59         if ~isnan(V(i))
60             V_temp(i) = V(i);
61         else
62             break
63         end
64     end
65     V_curva = V_temp;
66
67     % Arredonda os valores de irradiancia e temperatura
68     Ir = round(Ir_med);
69     T = round(T_med);
70
71     % Define o status da tag
72     tag_status = 'bruta';
73
74     % Salva as variaveis em um arquivo MAT
75     nomeArquivo = fullfile('.', '01) Bruta', sprintf('IxV_%s_S%d_T%d.mat', modelo, Ir, T));
76     save(nomeArquivo, 'I_curva', 'V_curva', 'Ir_med', 'T_med',
77         'Isc_med', 'Voc_med', 'modelo', 'tag_status');
78     disp(['Variaveis salvas em ' nomeArquivo]);
79
80 end

```

Apêndice 3 - P2

```

1  clear all
2  clc
3
4  %% Simulacao
5
6  % Desabilita avisos
7  warning off
8
9  %% Dados de datasheet (RSM100P)
10
11 % Define o modelo do painel solar
12 %modelo = 'RSM100P';
13
14 %% Irradiancia Constante
15
16 % Define o local dos arquivos MAT

```

```

17 local = fullfile('.', '01) Bruta', '*.mat');
18 % Obtem a lista de arquivos no diretorio especificado
19 files = dir(local);
20 % Conta o numero de arquivos encontrados
21 filesN = length(files);
22 % Define o modelo do painel solar
23 modelo = 'RSM100P';
24
25 % Loop sobre cada arquivo encontrado
26 for k = 1:1:filesN
27     file = files(k);
28     % Obtem o nome do arquivo
29     nomearq = file.name;
30     % Obtem o caminho do diretorio do arquivo
31     caminho = file.folder;
32     % Constroi o caminho completo do arquivo
33     curva_caminho = fullfile(caminho,nomearq);
34
35     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
36     expressionS = '_S(\d+)_';
37
38     % Encontrar correspondencia na string
39     match = regexp(nomearq, expressionS, 'match');
40
41     % Verificar se houve correspondencia
42     if ~isempty(match)
43         % Extrair o numero da correspondencia
44         Ir = str2double(match{1}(3:end-1));
45     end
46
47     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
48     expressionT = '_T(\d+)\.mat';
49
50     % Encontrar correspondencia na string
51     match = regexp(nomearq, expressionT, 'match');
52
53     % Verificar se houve correspondencia
54     if ~isempty(match)
55         % Extrair o numero da correspondencia
56         T = str2double(match{1}(3:end-4));
57     end
58
59     % Define a irradiancia
60     S = Ir;
61
62     % Executa a simulacao do modelo Simulink
63     sim('curvasIV_RSM100P.slx')
64
65     % Extrai os dados de corrente e tensao simulados
66     I_sim = transpose(I.Data);
67     V_sim = transpose(V.Data);
68     P_sim = I_sim.*V_sim;
69
70     % Configura a legenda do grafico
71     nome_leg = 'G = %d W/m2 | T = %d C';
72     leg = sprintf(nome_leg,Ir,T);
73
74     % Carrega os dados da curva medida
75     load(curva_caminho);
76     P_curva = I_curva.*V_curva;

```

```

77     P_max = max(P_curva);
78
79     % Define os limites dos eixos dos graficos
80     lim_I = ceil(Isc_med*1.1);
81     lim_V = ceil(Voc_med/10)*10;
82     lim_P = ceil(P_max/100)*100;
83
84     %% PLOTAGEM (Sconst)
85     % Configura tamanho e nome da figura
86
87     c = get(gca,'ColorOrder'); close all;
88     ImageDPI = 800; salvar = 1;
89     FontName = 'Times New Roman';
90     Size = 10;
91     ImageSizeX = 8.7*1.5; %tamanho em cm
92     ImageSizeY = 8.6*1.4; %tamanho em cm
93     FileLabel = nomearq;
94     figure('Units','centimeters','Position',[14 8 ImageSizeX
95         ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
96     set(gcf, 'Renderer', 'painters');
97
98     % Dimensoes do plot
99     width = 7.3*1.5/ImageSizeX;
100    height = 3.5*1.4/ImageSizeY;
101    x = 1.6/ImageSizeX;
102    y = 1.2/ImageSizeY;
103    y1 = 1.1/ImageSizeY + 1.1*height;
104
105    %% Configura Eixo 1
106    ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
107
108    hold on
109    % Plota a curva I-V simulada
110    plot(V_sim,I_sim,'linewidth',1.5,'Color','#0000FF')
111    % Plota a curva I-V medida
112    plot(V_curva,I_curva,'linewidth',1.5,'Color','#FF0000')
113
114    legend("Sim","Med",'Location','Southwest')
115
116    ly1 = ylabel(ax1,'Corrente do modulo
117        (A)','FontName',FontName,'FontSize',Size,'Interpreter',
118        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
119
120    set(ax1, ...
121        'Ylim' , [0,lim_I] , ...
122        'YTick' , 0:1:lim_I , ...
123        'Xlim' , [0,lim_V] , ...
124        'XTick' , 0:5:lim_V , ...
125        'TickDir' , 'in' , ...
126        'XMinorTick' , 'off' , ...
127        'YMinorTick' , 'off' , ...
128        'XGrid' , 'on' , ...
129        'YGrid' , 'on' , ...
130        'YMinorGrid' , 'off' , ...
131        'LineWidth' , 0.25 , ...
132        'XColor' , [0 0 0] , ...
133        'YColor' , [0 0 0] , ...
134        'FontSize' , Size , ...
135        'FontName' , FontName );

```

```

135     set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
136     set(ax1, 'XTickLabel', {''} );
137
138     %% Configura Eixo 2
139     ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
140
141     hold on
142     % Plota a curva P-V simulada
143     plot(V_sim,P_sim,'linewidth',1.5,'Color','#0000FF')
144     % Plota a curva P-V medida
145     plot(V_curva,P_curva,'linewidth',1.5,'Color','#FF0000')
146
147     legend("Sim","Med",'Location',"Northwest")
148
149     ly2 = ylabel(ax2,'Potencia do modulo
150         (W)','FontName',FontName,'FontSize',Size,'Interpreter',
151         'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
152
153     set(ax2, ...
154         'Ylim'      , [0,lim_P] , ...
155         'YTick'     , 0:20:lim_P , ...
156         'Xlim'      , [0,lim_V] , ...
157         'XTick'     , 0:5:lim_V , ...
158         'TickDir'   , 'in'      , ...
159         'XMinorTick', 'off'     , ...
160         'YMinorTick', 'off'     , ...
161         'XGrid'     , 'on'      , ...
162         'YGrid'     , 'on'      , ...
163         'YMinorGrid', 'off'     , ...
164         'LineWidth' , 0.25      , ...
165         'XColor'    , [0 0 0]   , ...
166         'YColor'    , [0 0 0]   , ...
167         'FontSize'  , Size      , ...
168         'FontName'  , FontName  );
169
170     set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
171
172     % Configura Eixo X
173     lx0 = xlabel(ax2,'Tensao do modulo
174         (V)','FontName',FontName,'FontSize',Size,'Interpreter','latex');
175
176     % Exportar Figura
177     if salvar == 1
178         set(gcf,'PaperUnits','centimeters',...
179             'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
180             'PaperSize',[ImageSizeX ImageSizeY]);
181         location = fullfile('.', '02) Validacao');
182         location = strcat(location,'/');
183         tic
184         print('-dpdf','-painters',strcat([location
185             FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
186         el_time = toc;
187         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
188     else
189         fprintf('\nOkk! Save option disable!!!\n');
190     end
191 end

```

Apendice 4 - P3

```

1  clear all
2  clc
3
4  %% Simulacao
5
6  % Desabilita avisos
7  warning off
8
9  %% Dados de datasheet (RSM100P)
10
11 %% Irradiancia Constante
12
13 % Define o local dos arquivos MAT
14 local = fullfile('.', '01 Bruta', '*.mat');
15 % Obtem a lista de arquivos no diretorio especificado
16 files = dir(local);
17 % Conta o numero de arquivos encontrados
18 filesN = length(files);
19 % Define o modelo do painel solar
20 modelo = 'RSM100P';
21
22 % Loop sobre cada arquivo encontrado
23 k = 1;
24 for k = 1:filesN
25     file = files(k);
26     % Obtem o nome do arquivo
27     nomearq = file.name;
28     % Obtem o caminho do diretorio do arquivo
29     caminho = file.folder;
30     % Constroi o caminho completo do arquivo
31     curva_caminho = fullfile(caminho,nomearq);
32
33     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
34     expressionS = '_S(\d+)_';
35
36     % Encontrar correspondencia na string
37     match = regexp(nomearq, expressionS, 'match');
38
39     % Verificar se houve correspondencia
40     if ~isempty(match)
41         % Extrair o numero da correspondencia
42         Ir = str2double(match{1}(3:end-1));
43     end
44
45     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
46     expressionT = '_T(\d+)\.mat';
47
48     % Encontrar correspondencia na string
49     match = regexp(nomearq, expressionT, 'match');
50
51     % Verificar se houve correspondencia
52     if ~isempty(match)
53         % Extrair o numero da correspondencia
54         T = str2double(match{1}(3:end-4));
55     end
56
57     % Define a irradiancia

```

```

58     S = Ir;
59
60     % Configura a legenda do grafico
61     nome_leg = 'G = %d W/m | T = %d oC';
62     leg = sprintf(nome_leg,Ir,T);
63
64     % Carrega os dados da curva medida
65     load(curva_caminho);
66     P_curva = I_curva.*V_curva;
67     P_max = max(P_curva);
68
69     % Define os limites dos eixos dos graficos
70     lim_I = ceil(Isc_med*1.1);
71     lim_V = ceil(Voc_med/10)*10;
72     lim_P = ceil(P_max/100)*100;
73
74     %% Tratamento de Plato em Voc
75     kv = 1;
76     kvmax = length(V_curva);
77     for kv = 1:1:kvmax
78         if V_curva(kv) <= Voc_med
79             V_curva_trat(kv) = V_curva(kv);
80             I_curva_trat(kv) = I_curva(kv);
81         else
82             V_curva_trat(kv) = Voc_med;
83             I_curva_trat(kv) = 0;
84         end
85     end
86
87     I_curva_trat(1) = Isc_med;
88     V_curva_trat(1) = 0;
89
90     P_curva_trat = I_curva_trat.*V_curva_trat;
91
92     %% PLOTAGEM (Sconst)
93     % Configura tamanho e nome da figura
94
95     c = get(gca,'ColorOrder'); close all;
96     ImageDPI = 800; salvar = 1;
97     FontName = 'Times New Roman';
98     Size = 10;
99     ImageSizeX = 8.7*1.5; % tamanho em cm
100    ImageSizeY = 8.6*1.4; % tamanho em cm
101    FileLabel = nomearq;
102    figure('Units','centimeters','Position',[14 8 ImageSizeX
103        ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
103    set(gcf, 'Renderer', 'painters');
104
105    % Dimensoes do plot
106    width = 7.3*1.5/ImageSizeX;
107    height = 3.5*1.4/ImageSizeY;
108    x = 1.6/ImageSizeX;
109    y = 1.2/ImageSizeY;
110    y1 = 1.1/ImageSizeY + 1.1*height;
111
112    %% Configura Eixo 1
113    ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
114
115    hold on
116    box on

```

```

117 % Plota a curva I-V medida
118 plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
119 % Plota a curva I-V tratada
120 plot(V_curva_trat,I_curva_trat,'linewidth',
121 2.5,'Color','#0000FF','LineStyle',':')
122
123 ly1 = ylabel(ax1,'Corrente do modulo
      (A)','FontName',FontName,'FontSize',Size,'Interpreter','Latex','HorizontalAlignment',
124 'center','backgroundcolor','none'); % left y-axis
125
126 set(ax1, ...
127 'Ylim' , [0,lim_I] , ...
128 'YTick' , 0:1:lim_I , ...
129 'Xlim' , [0,lim_V] , ...
130 'XTick' , 0:5:lim_V , ...
131 'TickDir' , 'in' , ...
132 'XMinorTick' , 'off' , ...
133 'YMinorTick' , 'off' , ...
134 'XGrid' , 'on' , ...
135 'YGrid' , 'on' , ...
136 'YMinorGrid' , 'off' , ...
137 'LineWidth' , 0.25 , ...
138 'XColor' , [0 0 0] , ...
139 'YColor' , [0 0 0] , ...
140 'FontSize' , Size , ...
141 'FontName' , FontName );
142
143 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
144 set(ax1, 'XTickLabel', {''}) );
145
146 %% Configura Eixo 2
147 ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
148
149 hold on
150 box on
151 % Plota a curva P-V medida
152 plot(V_curva,P_curva,'linewidth',1.5,'Color','#F4998D')
153 % Plota a curva P-V tratada
154 plot(V_curva_trat,P_curva_trat,
155 'linewidth',2.5,'Color','#0000FF','LineStyle',':')
156
157 ly2 = ylabel(ax2,'Potencia do modulo
      (W)','FontName',FontName,'FontSize',Size,'Interpreter','Latex','HorizontalAlignment',
158 'center','backgroundcolor','none'); % left y-axis
159
160 set(ax2, ...
161 'Ylim' , [0,lim_P] , ...
162 'YTick' , 0:25:lim_P , ...
163 'Xlim' , [0,lim_V] , ...
164 'XTick' , 0:5:lim_V , ...
165 'TickDir' , 'in' , ...
166 'XMinorTick' , 'off' , ...
167 'YMinorTick' , 'off' , ...
168 'XGrid' , 'on' , ...
169 'YGrid' , 'on' , ...
170 'YMinorGrid' , 'off' , ...
171 'LineWidth' , 0.25 , ...
172 'XColor' , [0 0 0] , ...
173 'YColor' , [0 0 0] , ...
174 'FontSize' , Size , ...

```

```

175         'FontName' , FontName );
176
177     set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
178
179     % Configura Eixo X
180     lx0 = xlabel(ax2,'Tensao do modulo
181     (V)','FontName',FontName,'FontSize',Size,'Interpreter','latex');
182
183     % Exportar Figura
184     if salvar == 1
185         set(gcf,'PaperUnits','centimeters',...
186         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
187         'PaperSize',[ImageSizeX ImageSizeY]);
188         location = fullfile('.', '03) Tratada\PDF');
189         location = strcat(location,'/');
190         tic
191         print('-dpdf','-painters',strcat([location
192             FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
193         el_time = toc;
194         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
195     else
196         fprintf('\nOkk! Save option disable!!!\n');
197     end
198
199     % Atualiza o status da tag
200     tag_status = 'tratada';
201
202     % Atualiza as curvas para salvar
203     V_curva = V_curva_trat;
204     I_curva = I_curva_trat;
205
206     % Salva as variaveis tratadas em um novo arquivo MAT
207     nomeArquivo = fullfile('.', '03) Tratada', sprintf('IxV_%s_S%d_T%d.mat', modelo, Ir, T));
208     save(nomeArquivo, 'I_curva',
209         'V_curva','Ir_med','T_med','Isc_med','Voc_med','modelo','tag_status');
210     disp(['Variaveis salvas em ' nomeArquivo]);
211
212     % Limpa variaveis para a proxima iteracao
213     clearvars -except files filesN k
214 end

```

Apêndice 5 - P4

```

1 clear all
2 clc
3
4 %% Simulacao
5
6 % Desabilita avisos
7 warning off
8
9 %% Dados de datasheet (RSM100P)
10
11 %% Irradiancia Constante
12
13 % Define o local dos arquivos MAT

```

```

14 local = fullfile('.', '03) Tratada', '*.mat');
15 % Obtem a lista de arquivos no diretorio especificado
16 files = dir(local);
17 % Conta o numero de arquivos encontrados
18 filesN = length(files);
19 % Define o modelo do painel solar
20 modelo = 'RSM100P';
21
22 % Loop sobre cada arquivo encontrado
23 k = 1;
24 for k = 1:filesN
25     file = files(k);
26     % Obtem o nome do arquivo
27     nomearq = file.name;
28     % Obtem o caminho do diretorio do arquivo
29     caminho = file.folder;
30     % Constroi o caminho completo do arquivo
31     curva_caminho = fullfile(caminho,nomearq);
32
33     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
34     expressionS = '_S(\d+)_';
35
36     % Encontrar correspondencia na string
37     match = regexp(nomearq, expressionS, 'match');
38
39     % Verificar se houve correspondencia
40     if ~isempty(match)
41         % Extrair o numero da correspondencia
42         Ir = str2double(match{1}(3:end-1));
43     end
44
45     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
46     expressionT = '_T(\d+)\.mat';
47
48     % Encontrar correspondencia na string
49     match = regexp(nomearq, expressionT, 'match');
50
51     % Verificar se houve correspondencia
52     if ~isempty(match)
53         % Extrair o numero da correspondencia
54         T = str2double(match{1}(3:end-4));
55     end
56
57     % Define a irradiancia
58     S = Ir;
59
60     % Configura a legenda do grafico
61     nome_leg = 'G = %d W/m | T = %d oC';
62     leg = sprintf(nome_leg,Ir,T);
63
64     % Carrega os dados da curva medida
65     load(curva_caminho);
66     P_curva = I_curva.*V_curva;
67     P_max = max(P_curva);
68
69     % Define os limites dos eixos dos graficos
70     lim_I = ceil(Isc_med*1.1);
71     lim_V = ceil(Voc_med/10)*10;
72     lim_P = ceil(P_max/100)*100;
73

```

```

74     %% Filtro
75     kvmax = length(V_curva);
76     kvint = ceil(2*kvmax/3);
77     for kv = 1:1:(kvmax-kvint-1)
78         deltaV(kv) = V_curva(kv+kvint+1) - V_curva(kv+kvint);
79     end
80
81     deltaV_max = max(deltaV);
82     tol_V = Voc_med*0.01; % Tolerancia 1%
83
84     if deltaV_max > tol_V
85         flag = 1; %nok
86         Ir_f = Ir;
87         T_f = T;
88     else
89         flag = 0; %ok
90     end
91
92     if flag == 0
93         % Se a curva passar no filtro
94         tag_status = 'filtrada';
95
96         %% PLOTAGEM (Sconst)
97         % Configura tamanho e nome da figura
98
99         c = get(gca,'ColorOrder'); close all;
100        ImageDPI = 800; salvar = 1;
101        FontName = 'Times New Roman';
102        Size = 10;
103        ImageSizeX = 8.7*1.5; % tamanho em cm
104        ImageSizeY = 8.6*1.4; % tamanho em cm
105        FileLabel = nomearq;
106        figure('Units','centimeters','Position',[14 8 ImageSizeX
107            ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
108        set(gcf, 'Renderer', 'painters');
109
110        % Dimensoes do plot
111        width = 7.3*1.5/ImageSizeX;
112        height = 3.5*1.4/ImageSizeY;
113        x = 1.6/ImageSizeX;
114        y = 1.2/ImageSizeY;
115        y1 = 1.1/ImageSizeY + 1.1*height;
116
117        %% Configura Eixo 1
118        ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
119
120        hold on
121        box on
122
123        % Plota a curva I-V medida
124        plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
125
126
127        ly1 = ylabel(ax1,'Corrente do modulo
128            (A)','FontName',FontName,'FontSize',Size,'Interpreter',
129            'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
130
131        set(ax1, ...
132            'Ylim' , [0,lim_I] , ...

```

```

131     'YTick'      , 0:1:lim_I , ...
132     'Xlim'      , [0,lim_V] , ...
133     'XTick'      , 0:5:lim_V, ...
134     'TickDir'    , 'in'      , ...
135     'XMinorTick' , 'off'     , ...
136     'YMinorTick' , 'off'     , ...
137     'XGrid'      , 'on'      , ...
138     'YGrid'      , 'on'      , ...
139     'YMinorGrid' , 'off'     , ...
140     'LineWidth'  , 0.25      , ...
141     'XColor'     , [0 0 0]   , ...
142     'YColor'     , [0 0 0]   , ...
143     'FontSize'   , Size      , ...
144     'FontName'   , FontName );
145
146 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
147 set(ax1, 'XTickLabel', {''} );
148
149 %% Configura Eixo 2
150 ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
151
152 hold on
153 box on
154
155 % Plota a curva P-V medida
156 plot(V_curva,P_curva,'linewidth',1.5,'Color','#F4998D')
157
158 ly2 = ylabel(ax2,'Potencia do modulo
        (W)','FontName',FontName,'FontSize',Size,'Interpreter',
        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
159
160 set(ax2, ...
161     'Ylim'      , [0,lim_P] , ...
162     'YTick'      , 0:25:lim_P , ...
163     'Xlim'      , [0,lim_V] , ...
164     'XTick'      , 0:5:lim_V , ...
165     'TickDir'    , 'in'      , ...
166     'XMinorTick' , 'off'     , ...
167     'YMinorTick' , 'off'     , ...
168     'XGrid'      , 'on'      , ...
169     'YGrid'      , 'on'      , ...
170     'YMinorGrid' , 'off'     , ...
171     'LineWidth'  , 0.25      , ...
172     'XColor'     , [0 0 0]   , ...
173     'YColor'     , [0 0 0]   , ...
174     'FontSize'   , Size      , ...
175     'FontName'   , FontName );
176
177 set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
178
179 %% Configura Eixo X
180 lx0 = xlabel(ax2,'Tensao do modulo
        (V)','FontName',FontName,'FontSize',Size,'Interpreter','latex');
181
182 % Exportar Figura
183 if salvar == 1
184     set(gcf,'PaperUnits','centimeters',...
185         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
186         'PaperSize',[ImageSizeX ImageSizeY]);
187     location = fullfile('.', '04) Filtrada - Filtro 1\PDF');

```

```

188     location = strcat(location, '/');
189     tic
190     print('-dpdf', '-painters', strcat([location
191         FileLabel], '.pdf'), strcat('-r', num2str(ImageDPI)))
192     el_time = toc;
193     fprintf('\nOkk! (%.2f seconds to save)\n', el_time);
194 else
195     fprintf('\nOkk! Save option disable!!!\n');
196 end
197
198 % Salva as variaveis tratadas em um novo arquivo MAT
199 nomeArquivo = fullfile('.', '04) Filtrada - Filtro 1', sprintf('IxV_%s_S%d_T%d.mat',
200     modelo, Ir, T));
201 save(nomeArquivo, 'I_curva',
202     'V_curva', 'Ir_med', 'T_med', 'Isc_med', 'Voc_med', 'modelo', 'tag_status');
203 disp(['Variaveis salvas em ' nomeArquivo]);
204 end
205
206 if flag == 1
207     % Se a curva nao passar no filtro
208     tag_status = 'descartada';
209
210     %% PLOTAGEM (Sconst)
211     % Configura tamanho e nome da figura
212
213     c = get(gca, 'ColorOrder'); close all;
214     ImageDPI = 800; salvar = 1;
215     FontName = 'Times New Roman';
216     Size = 10;
217     ImageSizeX = 8.7*1.5; % tamanho em cm
218     ImageSizeY = 8.6*1.4; % tamanho em cm
219     FileLabel = nomearq;
220     figure('Units', 'centimeters', 'Position', [14 8 ImageSizeX
221         ImageSizeY], 'PaperSize', [ImageSizeX ImageSizeY]);
222     set(gcf, 'Renderer', 'painters');
223
224     % Dimensoes do plot
225     width = 7.3*1.5/ImageSizeX;
226     height = 3.5*1.4/ImageSizeY;
227     x = 1.6/ImageSizeX;
228     y = 1.2/ImageSizeY;
229     y1 = 1.1/ImageSizeY + 1.1*height;
230
231     %% Configura Eixo 1
232     ax1 = axes('position', [x y1 width height], 'FontSize', Size, 'FontName', FontName);
233
234     hold on
235     box on
236
237     % Plota a curva I-V medida
238     plot(V_curva, I_curva, 'linewidth', 2.5, 'Color', '#F4998D')
239
240     ly1 = ylabel(ax1, 'Corrente do modulo
241         (A)', 'FontName', FontName, 'FontSize', Size, 'Interpreter',
242         'Latex', 'HorizontalAlignment', 'center', 'backgroundcolor',
243         'none'); % left y-axis
244
245     set(ax1, ...
246         'Ylim', [0, lim_I] , ...

```

```

242     'YTick'      , 0:1:lim_I , ...
243     'Xlim'      , [0,lim_V] , ...
244     'XTick'     , 0:5:lim_V, ...
245     'TickDir'   , 'in'      , ...
246     'XMinorTick', 'off'     , ...
247     'YMinorTick', 'off'     , ...
248     'XGrid'     , 'on'      , ...
249     'YGrid'     , 'on'      , ...
250     'YMinorGrid', 'off'     , ...
251     'LineWidth' , 0.25      , ...
252     'XColor'    , [0 0 0]   , ...
253     'YColor'    , [0 0 0]   , ...
254     'FontSize'  , Size      , ...
255     'FontName'  , FontName );
256
257 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
258 set(ax1, 'XTickLabel', {''} );
259
260 %% Configura Eixo 2
261 ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
262
263 hold on
264 box on
265
266 % Plota a curva P-V medida
267 plot(V_curva,P_curva,'linewidth',1.5,'Color','#F4998D')
268
269 ly2 = ylabel(ax2,'Potencia do modulo
    (W)','FontName',FontName,'FontSize',Size,'Interpreter',
    'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
270
271 set(ax2, ...
272     'Ylim'      , [0,lim_P] , ...
273     'YTick'     , 0:25:lim_P , ...
274     'Xlim'      , [0,lim_V] , ...
275     'XTick'     , 0:5:lim_V , ...
276     'TickDir'   , 'in'      , ...
277     'XMinorTick', 'off'     , ...
278     'YMinorTick', 'off'     , ...
279     'XGrid'     , 'on'      , ...
280     'YGrid'     , 'on'      , ...
281     'YMinorGrid', 'off'     , ...
282     'LineWidth' , 0.25      , ...
283     'XColor'    , [0 0 0]   , ...
284     'YColor'    , [0 0 0]   , ...
285     'FontSize'  , Size      , ...
286     'FontName'  , FontName );
287
288 set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
289
290 %% Configura Eixo X
291 lx0 = xlabel(ax2,'Tensao do modulo
    (V)','FontName',FontName,'FontSize',Size,'Interpreter','latex');
292
293 % Exportar Figura
294 if salvar == 1
295     set(gcf,'PaperUnits','centimeters',...
296         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
297         'PaperSize',[ImageSizeX ImageSizeY]);
298     location = fullfile('.', '05) Descartada - Filtro 1\PDF');

```

```

299         location = strcat(location, '/');
300         tic
301         print('-dpdf', '-painters', strcat([location
302             FileLabel], '.pdf'), strcat('-r', num2str(ImageDPI)))
303         el_time = toc;
304         fprintf('\nOkk! (%.2f seconds to save)\n', el_time);
305     else
306         fprintf('\nOkk! Save option disable!!!\n');
307     end
308     % Salva as variaveis tratadas em um novo arquivo MAT
309     nomeArquivo = fullfile('.', '05) Descartada - Filtro 1', sprintf('IxV_%s_S%d_T%d.mat',
310         modelo, Ir, T));
311     save(nomeArquivo, 'I_curva', 'V_curva', 'Ir_med',
312         'T_med', 'Isc_med', 'Voc_med', 'modelo', 'tag_status');
313     disp(['Variaveis salvas em ' nomeArquivo]);
314 end

```

Apendice 6 - P5

```

1  clear all
2  clc
3
4  %% Simulacao
5
6  % Desabilita avisos
7  warning off
8
9  %% Dados de datasheet (RSM100P)
10
11 %% Irradiancia Constante
12
13 % Define o local dos arquivos MAT
14 local = fullfile('.', '04) Filtrada - Filtro 1', '*.mat');
15 % Obtem a lista de arquivos no diretorio especificado
16 files = dir(local);
17 % Conta o numero de arquivos encontrados
18 filesN = length(files);
19 % Define o modelo do painel solar
20 modelo = 'RSM100P';
21
22 % Loop sobre cada arquivo encontrado
23 k = 1;
24 for k = 1:filesN
25     file = files(k);
26     % Obtem o nome do arquivo
27     nomearq = file.name;
28     % Obtem o caminho do diretorio do arquivo
29     caminho = file.folder;
30     % Constroi o caminho completo do arquivo
31     curva_caminho = fullfile(caminho, nomearq);
32
33     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
34     expressionS = '_S(\d+)_';
35
36     % Encontrar correspondencia na string

```

```

37     match = regexp(nomearq, expressionS, 'match');
38
39     % Verificar se houve correspondencia
40     if ~isempty(match)
41         % Extrair o numero da correspondencia
42         Ir = str2double(match{1}(3:end-1));
43     end
44
45     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
46     expressionT = '_T(\d+)\.mat';
47
48     % Encontrar correspondencia na string
49     match = regexp(nomearq, expressionT, 'match');
50
51     % Verificar se houve correspondencia
52     if ~isempty(match)
53         % Extrair o numero da correspondencia
54         T = str2double(match{1}(3:end-4));
55     end
56
57     % Define a irradiancia
58     S = Ir;
59
60     % Configura a legenda do grafico
61     nome_leg = 'G = %d W/m | T = %d oC';
62     leg = sprintf(nome_leg,Ir,T);
63
64     % Carrega os dados da curva medida
65     load(curva_caminho);
66     P_curva = I_curva.*V_curva;
67     P_max = max(P_curva);
68
69     % Define os limites dos eixos dos graficos
70     lim_I = ceil(Isc_med*1.1);
71     lim_V = ceil(Voc_med/10)*10;
72     lim_P = ceil(P_max/100)*100;
73
74     Ir_eq(k) = Ir_med;
75     Isc_eq(k) = Isc_med;
76 end
77
78 % Ajuste polinomial para Isc em funcao da irradiancia
79 [p,S] = polyfit(Ir_eq,Isc_eq,1);
80 [y_fit,delta] = polyval(p,Ir_eq,S);
81
82 x_Ir = 0:1:1500;
83 y_Isc = (x_Ir)*p(1) + p(2);
84
85 A = y_Isc(1001)*1.05 - x_Ir(1001)*p(1);
86 B = y_Isc(1001)*0.95 - x_Ir(1001)*p(1);
87
88 y_Isc_max = (x_Ir)*p(1) + A;
89 y_Isc_min = (x_Ir)*p(1) + B;
90
91 % Melhorar esse plot
92
93 c = get(gca,'ColorOrder'); close all;
94 ImageDPI = 800; salvar = 1;
95 FontName = 'Times New Roman';
96 Size = 12;

```

```

97 ImageSizeX = 8.7*1.5; % tamanho em cm
98 ImageSizeY = 8.6*1.4; % tamanho em cm
99 FileLabel = ['filtro2'];
100 figure('Units','centimeters','Position',[14 8 ImageSizeX ImageSizeY],'PaperSize',[ImageSizeX
    ImageSizeY]);
101 set(gcf, 'Renderer', 'painters');
102
103 % Dimensoes do plot
104 width = 7.3*1.5/ImageSizeX;
105 height = 7.3*1.4/ImageSizeY;
106 x = 1.5/ImageSizeX;
107 y = 1.2/ImageSizeY;
108 y1 = 1.1/ImageSizeY + 1.1*height;
109
110 % Configura Eixo 1
111 ax1 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
112
113 hold on
114 box on
115 % Plota a curva de ajuste polinomial
116 plot(x_Ir,y_Isc,'k','LineWidth',2,'LineStyle',':','Color','#0000FF')
117 % Plota as curvas de limite superior e inferior
118 plot(x_Ir,y_Isc_max,'Color','#add8e6')
119 plot(x_Ir,y_Isc_min,'Color','#add8e6')
120 % Plota os pontos de Isc medidos
121 plot(Ir_eq,Isc_eq,'r+','MarkerSize',10)
122
123 ly1 = ylabel(ax1,'Corrente de Curto Circuito
    (A)','FontName',FontName,'FontSize',Size,'Interpreter',
    'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
124 set(ax1, ...
125     'Ylim'      , [0,10]      , ...
126     'YTick'     , 0:1:10      , ...
127     'Xlim'      , [0,1500]    , ...
128     'XTick'     , 0:250:1500  , ...
129     'TickDir'   , 'in'        , ...
130     'XMinorTick', 'off'        , ...
131     'YMinorTick', 'off'        , ...
132     'XGrid'     , 'on'         , ...
133     'YGrid'     , 'on'         , ...
134     'YMinorGrid', 'off'        , ...
135     'LineWidth' , 0.25         , ...
136     'XColor'    , [0 0 0]      , ...
137     'YColor'    , [0 0 0]      , ...
138     'FontSize'  , Size         , ...
139     'FontName'  , FontName     );
140
141 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
142 lx0 = xlabel(ax1,'Irradiancia (W/m2)','FontName',
143 FontName,'FontSize',Size,'Interpreter','latex');
144
145 % Exportar Figura
146 if salvar == 1
147     set(gcf,'PaperUnits','centimeters',...
148     'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
149     'PaperSize',[ImageSizeX ImageSizeY]);
150     location = fullfile('.', '06 Filtrada - Filtro 2\PDF');
151     location = strcat(location, '/');
152     tic
153     print('-dpdf','-painters',strcat([location

```

```

        FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
154     saveas(gcf , strcat(FileLabel, '.fig'))
155     el_time = toc;
156     fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
157 else
158     fprintf('\nOkk! Save option disable!!!\n');
159 end
160
161 clearvars -except x_Ir y_Isc y_Isc_min y_Isc_max Ir_eq Isc_eq p S y_fit delta A B
162
163 %% Irradiancia Constante
164
165 local = fullfile('.', '04 Filtrada - Filtro 1', '*.mat');
166 files = dir(local);
167 filesN = length(files);
168 modelo = 'RSM100P';
169 k = 1;
170 for k = 1:1:filesN
171     file = files(k);
172     nomearq = file.name;
173     caminho = file.folder;
174     curva_caminho = fullfile(caminho,nomearq);
175
176     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
177     expressionS = '_S(\d+)_';
178
179     % Encontrar correspondencia na string
180     match = regexp(nomearq, expressionS, 'match');
181
182     % Verificar se houve correspondencia
183     if ~isempty(match)
184         % Extrair o numero da correspondencia
185         Ir = str2double(match{1}(3:end-1));
186     end
187
188     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
189     expressionT = '_T(\d+)\.mat';
190
191     % Encontrar correspondencia na string
192     match = regexp(nomearq, expressionT, 'match');
193
194     % Verificar se houve correspondencia
195     if ~isempty(match)
196         % Extrair o numero da correspondencia
197         T = str2double(match{1}(3:end-4));
198     end
199
200     % Define a irradiancia
201     S = Ir;
202
203     % Configura a legenda do grafico
204     nome_leg = 'G = %d W/m2 | T = %d oC';
205     leg = sprintf(nome_leg,Ir,T);
206
207     % Carrega os dados da curva medida
208     load(curva_caminho);
209     P_curva = I_curva.*V_curva;
210     P_max = max(P_curva);
211
212     % Define os limites dos eixos dos graficos

```

```

213     lim_I = ceil(Isc_med*1.1);
214     lim_V = ceil(Voc_med/10)*10;
215     lim_P = ceil(P_max/100)*100;
216
217     %% Filtro
218
219     % Calcula os limites de Isc para o filtro
220     Isc_max(k) = (p(1) * Ir_med) + A;
221     Isc_min(k) = (p(1) * Ir_med) + B;
222
223     % Verifica se Isc_med esta dentro dos limites
224     if Isc_med > Isc_max(k)
225         flag = 1;
226     else
227         flag = 0;
228     end
229
230     if Isc_med < Isc_min(k)
231         flag = 1;
232     else
233         flag = 0;
234     end
235
236     if flag == 0
237         % Se a curva passar no filtro
238         tag_status = 'filtrada';
239
240         %% PLOTAGEM (Sconst)
241         % Configura tamanho e nome da figura
242
243         c = get(gca,'ColorOrder'); close all;
244         ImageDPI = 800; salvar = 1;
245         FontName = 'Times New Roman';
246         Size = 10;
247         ImageSizeX = 8.7*1.5; % tamanho em cm
248         ImageSizeY = 8.6*1.4; % tamanho em cm
249         FileLabel = nomearq;
250         figure('Units','centimeters','Position',[14 8 ImageSizeX
251             ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
252         set(gcf, 'Renderer', 'painters');
253
254         % Dimensoes do plot
255         width = 7.3*1.5/ImageSizeX;
256         height = 3.5*1.4/ImageSizeY;
257         x = 1.6/ImageSizeX;
258         y = 1.2/ImageSizeY;
259         y1 = 1.1/ImageSizeY + 1.1*height;
260
261         %% Configura Eixo 1
262         ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
263
264         hold on
265         box on
266
267         % Plota a curva I-V medida
268         plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
269
270
271         ly1 = ylabel(ax1,'Corrente do modulo

```

```

    (A)', 'FontName', FontName, 'FontSize', Size, 'Interpreter',
    'Latex', 'HorizontalAlignment', 'center', 'backgroundcolor', 'none'); % left y-axis
272
273 set(ax1, ...
274     'Ylim'      , [0,lim_I] , ...
275     'YTick'     , 0:1:lim_I , ...
276     'Xlim'      , [0,lim_V] , ...
277     'XTick'     , 0:5:lim_V, ...
278     'TickDir'   , 'in'      , ...
279     'XMinorTick', 'off'     , ...
280     'YMinorTick', 'off'     , ...
281     'XGrid'     , 'on'      , ...
282     'YGrid'     , 'on'      , ...
283     'YMinorGrid', 'off'     , ...
284     'LineWidth' , 0.25      , ...
285     'XColor'    , [0 0 0]   , ...
286     'YColor'    , [0 0 0]   , ...
287     'FontSize'  , Size      , ...
288     'FontName'  , FontName  );
289
290 set(ax1, 'ticklength', 1/3*get(ax1, 'ticklength'))
291 set(ax1, 'XTickLabel', {''}) ;
292
293 %% Configura Eixo 2
294 ax2 = axes('position', [x y width height], 'FontSize', Size, 'FontName', FontName);
295
296 hold on
297 box on
298
299 % Plota a curva P-V medida
300 plot(V_curva, P_curva, 'linewidth', 1.5, 'Color', '#F4998D')
301
302 ly2 = ylabel(ax2, 'Potencia do modulo
    (W)', 'FontName', FontName, 'FontSize', Size, 'Interpreter',
    'Latex', 'HorizontalAlignment', 'center', 'backgroundcolor', 'none'); % left y-axis
303
304 set(ax2, ...
305     'Ylim'      , [0,lim_P] , ...
306     'YTick'     , 0:25:lim_P , ...
307     'Xlim'      , [0,lim_V] , ...
308     'XTick'     , 0:5:lim_V , ...
309     'TickDir'   , 'in'      , ...
310     'XMinorTick', 'off'     , ...
311     'YMinorTick', 'off'     , ...
312     'XGrid'     , 'on'      , ...
313     'YGrid'     , 'on'      , ...
314     'YMinorGrid', 'off'     , ...
315     'LineWidth' , 0.25      , ...
316     'XColor'    , [0 0 0]   , ...
317     'YColor'    , [0 0 0]   , ...
318     'FontSize'  , Size      , ...
319     'FontName'  , FontName  );
320
321 set(ax2, 'ticklength', 1/3*get(ax1, 'ticklength'))
322
323 % Configura Eixo X
324 lx0 = xlabel(ax2, 'Tensao do modulo
    (V)', 'FontName', FontName, 'FontSize', Size, 'Interpreter', 'latex');
325
326 % Exportar Figura

```

```

327     if salvar == 1
328         set(gcf,'PaperUnits','centimeters',...
329             'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
330             'PaperSize',[ImageSizeX ImageSizeY]);
331         location = fullfile('.', '06) Filtrada - Filtro 2\PDF');
332         location = strcat(location,'/');
333         tic
334         print('-dpdf','-painters',strcat([location
335             FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
336         el_time = toc;
337         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
338     else
339         fprintf('\nOkk! Save option disable!!!\n');
340     end
341
342     % Salva as variaveis tratadas em um novo arquivo MAT
343     nomeArquivo = fullfile('.', '06) Filtrada - Filtro 2', sprintf('IxV_%s_S%d_T%d.mat',
344         modelo, Ir, T));
345     save(nomeArquivo, 'I_curva',
346         'V_curva','Ir_med','T_med','Isc_med','Voc_med','modelo','tag_status');
347     disp(['Variaveis salvas em ' nomeArquivo]);
348 end
349
350 if flag == 1
351     % Se a curva nao passar no filtro
352     tag_status = 'descartada';
353
354     %% PLOTAGEM (Sconst)
355     % Configura tamanho e nome da figura
356
357     c = get(gca,'ColorOrder'); close all;
358     ImageDPI = 800; salvar = 1;
359     FontName = 'Times New Roman';
360     Size = 10;
361     ImageSizeX = 8.7*1.5; % tamanho em cm
362     ImageSizeY = 8.6*1.4; % tamanho em cm
363     FileLabel = nomearq;
364     figure('Units','centimeters','Position',[14 8 ImageSizeX
365         ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
366     set(gcf, 'Renderer', 'painters');
367
368     % Dimensoes do plot
369     width = 7.3*1.5/ImageSizeX;
370     height = 3.5*1.4/ImageSizeY;
371     x = 1.6/ImageSizeX;
372     y = 1.2/ImageSizeY;
373     y1 = 1.1/ImageSizeY + 1.1*height;
374
375     %% Configura Eixo 1
376     ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
377
378     hold on
379     box on
380
381     % Plota a curva I-V medida
382     plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
383
384     ly1 = ylabel(ax1,'Corrente do modulo

```

```

    (A)', 'FontName', FontName, 'FontSize', Size, 'Interpreter',
    'Latex', 'HorizontalAlignment', 'center', 'backgroundcolor', 'none'); % left y-axis
383
384 set(ax1, ...
385     'Ylim'      , [0,lim_I] , ...
386     'YTick'     , 0:1:lim_I , ...
387     'Xlim'      , [0,lim_V] , ...
388     'XTick'     , 0:5:lim_V, ...
389     'TickDir'   , 'in'      , ...
390     'XMinorTick', 'off'     , ...
391     'YMinorTick', 'off'     , ...
392     'XGrid'     , 'on'      , ...
393     'YGrid'     , 'on'      , ...
394     'YMinorGrid', 'off'     , ...
395     'LineWidth' , 0.25      , ...
396     'XColor'    , [0 0 0]   , ...
397     'YColor'    , [0 0 0]   , ...
398     'FontSize'  , Size      , ...
399     'FontName'  , FontName  );
400
401 set(ax1, 'ticklength', 1/3*get(ax1, 'ticklength'))
402 set(ax1, 'XTickLabel', {''}) ;
403
404 %% Configura Eixo 2
405 ax2 = axes('position', [x y width height], 'FontSize', Size, 'FontName', FontName);
406
407 hold on
408 box on
409
410 % Plota a curva P-V medida
411 plot(V_curva, P_curva, 'linewidth', 1.5, 'Color', '#F4998D')
412
413 ly2 = ylabel(ax2, 'Potencia do modulo
    (W)', 'FontName', FontName, 'FontSize', Size, 'Interpreter',
    'Latex', 'HorizontalAlignment', 'center', 'backgroundcolor', 'none'); % left y-axis
414
415 set(ax2, ...
416     'Ylim'      , [0,lim_P] , ...
417     'YTick'     , 0:25:lim_P , ...
418     'Xlim'      , [0,lim_V] , ...
419     'XTick'     , 0:5:lim_V , ...
420     'TickDir'   , 'in'      , ...
421     'XMinorTick', 'off'     , ...
422     'YMinorTick', 'off'     , ...
423     'XGrid'     , 'on'      , ...
424     'YGrid'     , 'on'      , ...
425     'YMinorGrid', 'off'     , ...
426     'LineWidth' , 0.25      , ...
427     'XColor'    , [0 0 0]   , ...
428     'YColor'    , [0 0 0]   , ...
429     'FontSize'  , Size      , ...
430     'FontName'  , FontName  );
431
432 set(ax2, 'ticklength', 1/3*get(ax1, 'ticklength'))
433
434 % Configura Eixo X
435 lx0 = xlabel(ax2, 'Tensao do modulo
    (V)', 'FontName', FontName, 'FontSize', Size, 'Interpreter', 'latex');
436
437 % Exportar Figura

```

```

438     if salvar == 1
439         set(gcf,'PaperUnits','centimeters',...
440             'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
441             'PaperSize',[ImageSizeX ImageSizeY]);
442         location = fullfile('.', '07) Descartada - Filtro 2\PDF');
443         location = strcat(location,'/');
444         tic
445         print('-dpdf','-painters',strcat([location
446             FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
447         el_time = toc;
448         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
449     else
450         fprintf('\nOkk! Save option disable!!!\n');
451     end
452     % Salva as variaveis tratadas em um novo arquivo MAT
453     nomeArquivo = fullfile('.', '07) Descartada - Filtro 2', sprintf('IxV_%s_S%d_T%d.mat',
454         modelo, Ir, T));
455     save(nomeArquivo, 'I_curva',
456         'V_curva','Ir_med','T_med','Isc_med','Voc_med','modelo','tag_status');
457     disp(['Variaveis salvas em ' nomeArquivo]);
458 end
459 end

```

Apendice 7 - P6

```

1  clear all
2  clc
3
4  %% Simulacao
5
6  % Desabilita avisos
7  warning off
8
9  %% Dados de datasheet (RSM100P)
10
11 %% Irradiancia Constante
12
13 % Define o local dos arquivos MAT
14 local = fullfile('.', '06) Filtrada - Filtro 2', '*.mat');
15 % Obtem a lista de arquivos no diretorio especificado
16 files = dir(local);
17 % Conta o numero de arquivos encontrados
18 filesN = length(files);
19 % Define o modelo do painel solar
20 modelo = 'RSM100P';
21
22 % Loop sobre cada arquivo encontrado
23 k = 1;
24 for k = 1:1:filesN
25     file = files(k);
26     % Obtem o nome do arquivo
27     nomearq = file.name;
28     % Obtem o caminho do diretorio do arquivo
29     caminho = file.folder;
30     % Constroi o caminho completo do arquivo
31     curva_caminho = fullfile(caminho,nomearq);

```

```

32
33 % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
34 expressionS = '_S(\d+)_';
35
36 % Encontrar correspondencia na string
37 match = regexp(nomearq, expressionS, 'match');
38
39 % Verificar se houve correspondencia
40 if ~isempty(match)
41     % Extrair o numero da correspondencia
42     Ir = str2double(match{1}(3:end-1));
43 end
44
45 % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
46 expressionT = '_T(\d+)\.mat';
47
48 % Encontrar correspondencia na string
49 match = regexp(nomearq, expressionT, 'match');
50
51 % Verificar se houve correspondencia
52 if ~isempty(match)
53     % Extrair o numero da correspondencia
54     T = str2double(match{1}(3:end-4));
55 end
56
57 % Define a irradiancia
58 S = Ir;
59
60 % Configura a legenda do grafico
61 nome_leg = 'G = %d W/m2 | T = %d oC';
62 leg = sprintf(nome_leg, Ir, T);
63
64 % Carrega os dados da curva medida
65 load(curva_caminho);
66 P_curva = I_curva.*V_curva;
67 P_max = max(P_curva);
68 mpp = find(P_curva == P_max);
69
70 % Obtem a tensao e corrente no ponto de maxima potencia
71 Vmpp = V_curva(mpp);
72 Impp = I_curva(mpp);
73
74 % Define os limites dos eixos dos graficos
75 lim_I = ceil(Isc_med*1.1);
76 lim_V = ceil(Voc_med/10)*10;
77 lim_P = ceil(P_max/100)*100;
78
79 Ir_eq(k) = Ir_med;
80 Impp_eq(k) = Impp;
81 end
82
83 % Ajuste polinomial para Impp em funcao da irradiancia
84 [p,S] = polyfit(Ir_eq, Impp_eq, 1);
85 [y_fit, delta] = polyval(p, Ir_eq, S);
86
87 x_Ir = 0:1:1500;
88 y_Impp = (x_Ir)*p(1) + p(2);
89
90 A = y_Impp(1001)*1.05 - x_Ir(1001)*p(1);
91 B = y_Impp(1001)*0.95 - x_Ir(1001)*p(1);

```

```

92
93 y_Impp_max = (x_Ir)*p(1) + A;
94 y_Impp_min = (x_Ir)*p(1) + B;
95
96 % Melhorar esse plot
97
98 c = get(gca,'ColorOrder'); close all;
99 ImageDPI = 800; salvar = 1;
100 FontName = 'Times New Roman';
101 Size = 12;
102 ImageSizeX = 8.7*1.5; % tamanho em cm
103 ImageSizeY = 8.6*1.4; % tamanho em cm
104 FileLabel = ['filtro3'];
105 figure('Units','centimeters','Position',[14 8 ImageSizeX ImageSizeY],'PaperSize',[ImageSizeX
    ImageSizeY]);
106 set(gcf, 'Renderer', 'painters');
107
108 % Dimensoes do plot
109 width = 7.3*1.5/ImageSizeX;
110 height = 7.3*1.4/ImageSizeY;
111 x = 1.5/ImageSizeX;
112 y = 1.2/ImageSizeY;
113 y1 = 1.1/ImageSizeY + 1.1*height;
114
115 % Configura Eixo 1
116 ax1 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
117
118 hold on
119 box on
120 % Plota a curva de ajuste polinomial
121 plot(x_Ir,y_Impp,'k','LineWidth',2,'LineStyle',':', 'Color','#0000FF')
122 % Plota as curvas de limite superior e inferior
123 plot(x_Ir,y_Impp_max,'Color','#add8e6','LineWidth',2)
124 plot(x_Ir,y_Impp_min,'Color','#add8e6','LineWidth',2)
125 % Plota os pontos de Impp medidos
126 plot(Ir_eq,Impp_eq,'r+','MarkerSize',10)
127
128 ly1 = ylabel(ax1,'Corrente em MPP (A)','FontName',FontName,'FontSize',Size,'Interpreter',
    'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
129 set(ax1, ...
130     'Ylim'      , [0,10]      , ...
131     'YTick'     , 0:1:10      , ...
132     'Xlim'      , [0,1500]    , ...
133     'XTick'     , 0:250:1500  , ...
134     'TickDir'   , 'in'       , ...
135     'XMinorTick', 'off'       , ...
136     'YMinorTick', 'off'       , ...
137     'XGrid'     , 'on'       , ...
138     'YGrid'     , 'on'       , ...
139     'YMinorGrid', 'off'       , ...
140     'LineWidth' , 0.25       , ...
141     'XColor'    , [0 0 0]     , ...
142     'YColor'    , [0 0 0]     , ...
143     'FontSize'  , Size        , ...
144     'FontName'  , FontName    );
145
146 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
147 lx0 = xlabel(ax1,'Irradiancia
    (W/m2)','FontName',FontName,'FontSize',Size,'Interpreter','latex');
148

```

```

149 % Exportar Figura
150 if salvar == 1
151     set(gcf,'PaperUnits','centimeters',...
152         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
153         'PaperSize',[ImageSizeX ImageSizeY]);
154     location = fullfile('.', '08) Filtrada - Filtro 3\PDF');
155     location = strcat(location,'/');
156     tic
157     print('-dpdf','-painters',strcat([location
158         FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
159     saveas(gcf, strcat(FileLabel, '.fig'))
160     el_time = toc;
161     fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
162 else
163     fprintf('\nOkk! Save option disable!!!\n');
164 end
165 clearvars -except x_Ir y_Impp y_Impp_min y_Impp_max Ir_eq Impp_eq p S y_fit delta A B
166
167 %% Irradiancia Constante
168
169 local = fullfile('.', '06) Filtrada - Filtro 2', '*.mat');
170 files = dir(local);
171 filesN = length(files);
172 modelo = 'RSM100P';
173 k = 1;
174 for k = 1:1:filesN
175     file = files(k);
176     nomearq = file.name;
177     caminho = file.folder;
178     curva_caminho = fullfile(caminho,nomearq);
179
180     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
181     expressionS = '_S(\d+)_';
182
183     % Encontrar correspondencia na string
184     match = regexp(nomearq, expressionS, 'match');
185
186     % Verificar se houve correspondencia
187     if ~isempty(match)
188         % Extrair o numero da correspondencia
189         Ir = str2double(match{1}(3:end-1));
190     end
191
192     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
193     expressionT = '_T(\d+)\.mat';
194
195     % Encontrar correspondencia na string
196     match = regexp(nomearq, expressionT, 'match');
197
198     % Verificar se houve correspondencia
199     if ~isempty(match)
200         % Extrair o numero da correspondencia
201         T = str2double(match{1}(3:end-4));
202     end
203
204     % Define a irradiancia
205     S = Ir;
206
207     % Configura a legenda do grafico

```

```

208     nome_leg = 'G = %d W/m2 | T = %d oC';
209     leg = sprintf(nome_leg,Ir,T);
210
211     % Carrega os dados da curva medida
212     load(curva_caminho);
213     P_curva = I_curva.*V_curva;
214     P_max = max(P_curva);
215
216     mpp = find(P_curva == P_max);
217
218     Vmpp = V_curva(mpp);
219     Impp = I_curva(mpp);
220
221     lim_I = ceil(Isc_med*1.1);
222     lim_V = ceil(Voc_med/10)*10;
223     lim_P = ceil(P_max/100)*100;
224
225     %% Filtro
226
227     Impp_max(k) = (p(1) * Ir_med) + A;
228     Impp_min(k) = (p(1) * Ir_med) + B;
229
230     if Impp > Impp_max(k)
231         flag = 1;
232     else
233         flag = 0;
234     end
235
236     if Impp < Impp_min(k)
237         flag = 1;
238     else
239         flag = 0;
240     end
241
242     if flag == 0
243         % Se a curva passar no filtro
244         tag_status = 'filtrada';
245
246         %% PLOTAGEM (Sconst)
247         % Configura tamanho e nome da figura
248
249         c = get(gca,'ColorOrder'); close all;
250         ImageDPI = 800; salvar = 1;
251         FontName = 'Times New Roman';
252         Size = 10;
253         ImageSizeX = 8.7*1.5; % tamanho em cm
254         ImageSizeY = 8.6*1.4; % tamanho em cm
255         FileLabel = nomearq;
256         figure('Units','centimeters','Position',[14 8 ImageSizeX
257             ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
258         set(gcf, 'Renderer', 'painters');
259
260         % Dimensoes do plot
261         width = 7.3*1.5/ImageSizeX;
262         height = 3.5*1.4/ImageSizeY;
263         x = 1.6/ImageSizeX;
264         y = 1.2/ImageSizeY;
265         y1 = 1.1/ImageSizeY + 1.1*height;
266
267         %% Configura Eixo 1

```

```

267     ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
268
269     hold on
270     box on
271
272     % Plota a curva I-V medida
273     plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
274
275
276
277     ly1 = ylabel(ax1,'Corrente do modulo
        (A)','FontName',FontName,'FontSize',Size,'Interpreter',
        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
278
279     set(ax1, ...
280         'Ylim'      , [0,lim_I] , ...
281         'YTick'     , 0:1:lim_I , ...
282         'Xlim'      , [0,lim_V] , ...
283         'XTick'     , 0:5:lim_V , ...
284         'TickDir'   , 'in'      , ...
285         'XMinorTick', 'off'     , ...
286         'YMinorTick', 'off'     , ...
287         'XGrid'     , 'on'      , ...
288         'YGrid'     , 'on'      , ...
289         'YMinorGrid', 'off'     , ...
290         'LineWidth' , 0.25      , ...
291         'XColor'    , [0 0 0]   , ...
292         'YColor'    , [0 0 0]   , ...
293         'FontSize'  , Size       , ...
294         'FontName'  , FontName );
295
296     set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
297     set(ax1, 'XTickLabel', {''} );
298
299     %% Configura Eixo 2
300     ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
301
302     hold on
303     box on
304
305     % Plota a curva P-V medida
306     plot(V_curva,P_curva,'linewidth',1.5,'Color','#F4998D')
307
308     ly2 = ylabel(ax2,'Potencia do modulo
        (W)','FontName',FontName,'FontSize',Size,'Interpreter',
        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
309
310     set(ax2, ...
311         'Ylim'      , [0,lim_P] , ...
312         'YTick'     , 0:25:lim_P , ...
313         'Xlim'      , [0,lim_V] , ...
314         'XTick'     , 0:5:lim_V , ...
315         'TickDir'   , 'in'      , ...
316         'XMinorTick', 'off'     , ...
317         'YMinorTick', 'off'     , ...
318         'XGrid'     , 'on'      , ...
319         'YGrid'     , 'on'      , ...
320         'YMinorGrid', 'off'     , ...
321         'LineWidth' , 0.25      , ...
322         'XColor'    , [0 0 0]   , ...

```

```

323     'YColor' , [0 0 0] , ...
324     'FontSize' , Size , ...
325     'FontName' , FontName );
326
327 set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
328
329 % Configura Eixo X
330 lx0 = xlabel(ax2,'Tensao do modulo
      (V)', 'FontName',FontName,'FontSize',Size,'Interpreter','latex');
331
332 % Exportar Figura
333 if salvar == 1
334     set(gcf,'PaperUnits','centimeters',...
335         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
336         'PaperSize',[ImageSizeX ImageSizeY]);
337     location = fullfile('.', '08) Filtrada - Filtro 3\PDF');
338     location = strcat(location,'/');
339     tic
340     print('-dpdf','-painters',strcat([location
      FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
341     el_time = toc;
342     fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
343 else
344     fprintf('\nOkk! Save option disable!!!\n');
345 end
346
347 % Salva as variaveis tratadas em um novo arquivo MAT
348 nomeArquivo = fullfile('.', '08) Filtrada - Filtro 3', sprintf('IxV_%s_S%d_T%d.mat',
      modelo, Ir, T));
349 save(nomeArquivo, 'I_curva',
350     'V_curva','Ir_med','T_med','Isc_med','Voc_med','modelo','tag_status');
351 disp(['Variaveis salvas em ' nomeArquivo]);
352 end
353
354 if flag == 1
355     % Se a curva nao passar no filtro
356     tag_status = 'descartada';
357
358     %% PLOTAGEM (Sconst)
359     % Configura tamanho e nome da figura
360
361     c = get(gca,'ColorOrder'); close all;
362     ImageDPI = 800; salvar = 1;
363     FontName = 'Times New Roman';
364     Size = 10;
365     ImageSizeX = 8.7*1.5; % tamanho em cm
366     ImageSizeY = 8.6*1.4; % tamanho em cm
367     FileLabel = nomearg;
368     figure('Units','centimeters','Position',[14 8 ImageSizeX
      ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
369     set(gcf, 'Renderer', 'painters');
370
371     % Dimensoes do plot
372     width = 7.3*1.5/ImageSizeX;
373     height = 3.5*1.4/ImageSizeY;
374     x = 1.6/ImageSizeX;
375     y = 1.2/ImageSizeY;
376     y1 = 1.1*ImageSizeY + 1.1*height;
377
378     %% Configura Eixo 1

```

```

379     ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
380
381     hold on
382     box on
383
384     % Plota a curva I-V medida
385     plot(V_curva,I_curva,'linewidth',2.5,'Color','#F4998D')
386
387
388
389     ly1 = ylabel(ax1,'Corrente do modulo
        (A)','FontName',FontName,'FontSize',Size,'Interpreter',
        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
390
391     set(ax1, ...
392         'Ylim'      , [0,lim_I] , ...
393         'YTick'     , 0:1:lim_I , ...
394         'Xlim'      , [0,lim_V] , ...
395         'XTick'     , 0:5:lim_V, ...
396         'TickDir'   , 'in'      , ...
397         'XMinorTick', 'off'     , ...
398         'YMinorTick', 'off'     , ...
399         'XGrid'     , 'on'      , ...
400         'YGrid'     , 'on'      , ...
401         'YMinorGrid', 'off'     , ...
402         'LineWidth' , 0.25      , ...
403         'XColor'    , [0 0 0]   , ...
404         'YColor'    , [0 0 0]   , ...
405         'FontSize'  , Size      , ...
406         'FontName'  , FontName );
407
408     set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
409     set(ax1, 'XTickLabel', {''} );
410
411     %% Configura Eixo 2
412     ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
413
414     hold on
415     box on
416
417     % Plota a curva P-V medida
418     plot(V_curva,P_curva,'linewidth',1.5,'Color','#F4998D')
419
420     ly2 = ylabel(ax2,'Potencia do modulo
        (W)','FontName',FontName,'FontSize',Size,'Interpreter',
        'Latex','HorizontalAlignment','center','backgroundcolor','none'); % left y-axis
421
422
423     set(ax2, ...
424         'Ylim'      , [0,lim_P] , ...
425         'YTick'     , 0:25:lim_P , ...
426         'Xlim'      , [0,lim_V] , ...
427         'XTick'     , 0:5:lim_V , ...
428         'TickDir'   , 'in'      , ...
429         'XMinorTick', 'off'     , ...
430         'YMinorTick', 'off'     , ...
431         'XGrid'     , 'on'      , ...
432         'YGrid'     , 'on'      , ...
433         'YMinorGrid', 'off'     , ...
434         'LineWidth' , 0.25      , ...
435         'XColor'    , [0 0 0]   , ...

```

```

436         'YColor' , [0 0 0] , ...
437         'FontSize' , Size , ...
438         'FontName' , FontName );
439
440     set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
441
442     % Configura Eixo X
443     lx0 = xlabel(ax2,'Tensao do modulo
444     (V)', 'FontName',FontName, 'FontSize',Size, 'Interpreter', 'latex');
445
446     % Exportar Figura
447     if salvar == 1
448         set(gcf,'PaperUnits','centimeters',...
449         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
450         'PaperSize',[ImageSizeX ImageSizeY]);
451         location = fullfile('.', '09) Descartada - Filtro 3\PDF');
452         location = strcat(location,'/');
453         tic
454         print('-dpdf','-painters',strcat([location
455             FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
456         el_time = toc;
457         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
458     else
459         fprintf('\nOkk! Save option disable!!!\n');
460     end
461
462     % Salva as variaveis tratadas em um novo arquivo MAT
463     nomeArquivo = fullfile('.', '09) Descartada - Filtro 3', sprintf('IxV_%s_S%d_T%d.mat',
464         modelo, Ir, T));
465     save(nomeArquivo, 'I_curva',
466         'V_curva', 'Ir_med', 'T_med', 'Isc_med', 'Voc_med', 'modelo', 'tag_status');
467     disp(['Variaveis salvas em ' nomeArquivo]);
468 end
end

```

Apendice 8 - P7

```

1  clear all
2  clc
3
4  %% Simulacao
5
6  % Desabilita avisos
7  warning off
8
9  %% Dados de datasheet (RSM100P)
10
11  %% Irradiancia Constante
12
13  % Define o local dos arquivos MAT
14  local = fullfile('.', '08) Filtrada - Filtro 3', '*.mat');
15  % Obtem a lista de arquivos no diretorio especificado
16  files = dir(local);
17  % Conta o numero de arquivos encontrados
18  filesN = length(files);
19  % Define o modelo do painel solar
20  modelo = 'RSM100P';

```

```

21
22 % Loop sobre cada arquivo encontrado
23 k = 1;
24 for k = 1:1:filesN
25     file = files(k);
26     % Obtem o nome do arquivo
27     nomearq = file.name;
28     % Obtem o caminho do diretorio do arquivo
29     caminho = file.folder;
30     % Constroi o caminho completo do arquivo
31     curva_caminho = fullfile(caminho,nomearq);
32
33     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
34     expressionS = '_S(\d+)_';
35
36     % Encontrar correspondencia na string
37     match = regexp(nomearq, expressionS, 'match');
38
39     % Verificar se houve correspondencia
40     if ~isempty(match)
41         % Extrair o numero da correspondencia
42         Ir = str2double(match{1}(3:end-1));
43     end
44
45     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
46     expressionT = '_T(\d+)\.mat';
47
48     % Encontrar correspondencia na string
49     match = regexp(nomearq, expressionT, 'match');
50
51     % Verificar se houve correspondencia
52     if ~isempty(match)
53         % Extrair o numero da correspondencia
54         T = str2double(match{1}(3:end-4));
55     end
56
57     % Define a irradiancia
58     S = Ir;
59
60     % Configura a legenda do grafico
61     nome_leg = 'G = %d W/m2 | T = %d oC';
62     leg = sprintf(nome_leg,Ir,T);
63
64     % Carrega os dados da curva medida
65     load(curva_caminho);
66     P_curva = I_curva.*V_curva;
67     P_max = max(P_curva);
68     mpp = find(P_curva == P_max);
69     mpp = mpp(1);
70
71     % Obtem a tensao e corrente no ponto de maxima potencia
72     Vmpp = V_curva(mpp);
73     Impp = I_curva(mpp);
74
75     % Define os limites dos eixos dos graficos
76     lim_I = ceil(Isc_med*1.1);
77     lim_V = ceil(Voc_med/10)*10;
78     lim_P = ceil(P_max/100)*100;
79
80     Ir_eq(k) = Ir_med;

```

```

81     Impp_eq(k) = Impp;
82 end
83
84 %% Irradiancia Constante
85
86 % Define novamente o local dos arquivos MAT
87 local = fullfile('.', '08) Filtrada - Filtro 3', '*.mat');
88 files = dir(local);
89 filesN = length(files);
90 modelo = 'RSM100P';
91 k = 1;
92 for k = 1:filesN
93     file = files(k);
94     nomearq = file.name;
95     caminho = file.folder;
96     curva_caminho = fullfile(caminho,nomearq);
97
98     % Expressao regular para encontrar o numero entre os underscores (Irradiancia)
99     expressionS = '_S(\d+)_';
100
101     % Encontrar correspondencia na string
102     match = regexp(nomearq, expressionS, 'match');
103
104     % Verificar se houve correspondencia
105     if ~isempty(match)
106         % Extrair o numero da correspondencia
107         Ir = str2double(match{1}(3:end-1));
108     end
109
110     % Expressao regular para encontrar o numero entre os underscores apos 'T' (Temperatura)
111     expressionT = '_T(\d+)\.mat';
112
113     % Encontrar correspondencia na string
114     match = regexp(nomearq, expressionT, 'match');
115
116     % Verificar se houve correspondencia
117     if ~isempty(match)
118         % Extrair o numero da correspondencia
119         T = str2double(match{1}(3:end-4));
120     end
121
122     % Define a irradiancia
123     S = Ir;
124
125     % Configura a legenda do grafico
126     nome_leg = 'G = %d W/m2 | T = %d oC';
127     leg = sprintf(nome_leg,Ir,T);
128
129     % Carrega os dados da curva medida
130     load(curva_caminho);
131     P_curva = I_curva.*V_curva;
132     P_max = max(P_curva);
133
134     mpp = find(P_curva == P_max);
135
136     Vmpp = V_curva(mpp);
137     Impp = I_curva(mpp);
138
139     lim_I = ceil(Isc_med*1.1);
140     lim_V = ceil(Voc_med/10)*10;

```

```

141     lim_P = ceil(P_max/100)*100;
142
143     %% Filtro
144
145     % Calcular o numero de pontos desejados
146     num_pontos = 500;
147
148     // Calcular a distancia entre os pontos de V
149     distancia = max(V_curva) / (num_pontos - 1);
150
151     // Inicializar vetores para os novos valores de V e I
152     V_novo = linspace(0, max(V_curva), num_pontos);
153     I_novo = zeros(1, num_pontos);
154
155     // Primeira etapa: calcular os pontos de I_novo que possuem media definida
156     for n = 1:num_pontos
157         // Encontrar os indices dos valores de V_curva dentro da faixa de interpolacao para o
            ponto atual de V_novo
158         indices = find(V_curva >= V_novo(n) - distancia/2 & V_curva <= V_novo(n) + distancia/2);
159
160         if ~isempty(indices)
161             // Se houver pontos de V_curva dentro da faixa de interpolacao, calcular a media dos
                valores de I_curva
162             I_novo(n) = mean(I_curva(indices));
163         end
164     end
165
166     // Segunda etapa: realizar a interpolacao entre os pontos de I_novo que ainda nao possuem
        valores definidos
167     for n = 1:num_pontos
168         if I_novo(n) == 0
169             // Se o valor de I_novo para o ponto atual ainda nao estiver definido
170             idx_anterior = find(I_novo(1:n-1), 1, 'last');
171             idx_posterior = find(I_novo(n+1:end), 1, 'first') + n;
172
173             if ~isempty(idx_anterior) && ~isempty(idx_posterior)
174                 // Se ambos os indices forem encontrados, realizar a interpolacao linear
175                 V_anterior = V_novo(idx_anterior);
176                 V_posterior = V_novo(idx_posterior);
177                 I_anterior = I_novo(idx_anterior);
178                 I_posterior = I_novo(idx_posterior);
179
180                 // Calcular a interpolacao linear para o valor de I_novo
181                 I_novo(n) = ((V_posterior - V_novo(n)) * I_anterior + (V_novo(n) - V_anterior) *
                    I_posterior) / (V_posterior - V_anterior);
182             elseif ~isempty(idx_anterior)
183                 // Se houver apenas um ponto anterior, atribuir o valor de I_novo correspondente
184                 I_novo(n) = I_novo(idx_anterior);
185             elseif ~isempty(idx_posterior)
186                 // Se houver apenas um ponto posterior, atribuir o valor de I_novo correspondente
187                 I_novo(n) = I_novo(idx_posterior);
188             end
189         end
190     end
191
192     I_novo(500) = 0;
193
194     //I_curva = I_novo;
195     //V_curva = V_novo;
196     P_novo = I_novo.*V_novo;

```

```

197
198 %% PLOTAGEM (Sconst)
199 % Configura tamanho e nome da figura
200
201 c = get(gca,'ColorOrder'); close all;
202 ImageDPI = 800; salvar = 1;
203 FontName = 'Times New Roman';
204 Size = 10;
205 ImageSizeX = 8.7*1.5; %tamanho em cm
206 ImageSizeY = 8.6*1.4; %tamanho em cm
207 FileLabel = nomearq;
208 figure('Units','centimeters','Position',[14 8 ImageSizeX
      ImageSizeY],'PaperSize',[ImageSizeX ImageSizeY]);
209 set(gcf, 'Renderer', 'painters');
210
211 % Dimensoes do plot
212 width = 7.3*1.5/ImageSizeX;
213 height = 3.5*1.4/ImageSizeY;
214 x = 1.6/ImageSizeX;
215 y = 1.2/ImageSizeY;
216 y1 = 1.1/ImageSizeY + 1.1*height;
217
218 %% Configura Eixo 1
219 ax1 = axes('position',[x y1 width height],'FontSize',Size,'FontName',FontName);
220
221 hold on
222 box on
223
224 % Plota a nova curva I-V
225 plot(V_novo,I_novo,'linewidth',2.5,'Color','#0000FF')
226 % Plota a curva I-V medida
227 plot(V_curva,I_curva,'+','Color','#F4998D')
228
229
230
231 ly1 = ylabel(ax1,'Corrente do modulo (A)','FontName',FontName,'FontSize'
232 ,Size,'Interpreter','Latex','HorizontalAlignment'
233 'center','backgroundcolor','none'); % left y-axis
234
235 set(ax1, ...
236     'Ylim' , [0,lim_I] , ... %0.7
237     'YTick' , 0:1:lim_I , ...
238     'Xlim' , [0,lim_V] , ...
239     'XTick' , 0:5:lim_V, ...
240     'TickDir' , 'in' , ...
241     'XMinorTick' , 'off' , ...
242     'YMinorTick' , 'off' , ...
243     'XGrid' , 'on' , ...
244     'YGrid' , 'on' , ...
245     'YMinorGrid' , 'off' , ...
246     'LineWidth' , 0.25 , ...
247     'XColor' , [0 0 0] , ...
248     'YColor' , [0 0 0] , ...
249     'FontSize' , Size , ...
250     'FontName' , FontName );
251
252 set(ax1,'ticklength',1/3*get(ax1,'ticklength'))
253 set(ax1, 'XTickLabel', {''} );
254
255 %% Configura Eixo 2

```

```

256     ax2 = axes('position',[x y width height],'FontSize',Size,'FontName',FontName);
257
258     hold on
259     box on
260
261     % Plota a nova curva P-V
262     plot(V_novo,P_novo,'linewidth',1.5,'Color','#0000FF')
263     % Plota a curva P-V medida
264     plot(V_curva,P_curva,'+','Color','#F4998D')
265
266     ly2 = ylabel(ax2,'Potencia do modulo
           (W)','FontName',FontName,'FontSize',Size,'Interpreter','Latex','HorizontalAlignment',
267     'center','backgroundcolor','none'); % left y-axis
268
269     set(ax2, ...
270         'Ylim'      , [0,lim_P] , ... %0.7
271         'YTick'     , 0:25:lim_P , ...
272         'Xlim'      , [0,lim_V] , ...
273         'XTick'     , 0:5:lim_V , ...
274         'TickDir'   , 'in'      , ...
275         'XMinorTick', 'off'     , ...
276         'YMinorTick', 'off'     , ...
277         'XGrid'     , 'on'      , ...
278         'YGrid'     , 'on'      , ...
279         'YMinorGrid', 'off'     , ...
280         'LineWidth' , 0.25      , ...
281         'XColor'    , [0 0 0]   , ...
282         'YColor'    , [0 0 0]   , ...
283         'FontSize'  , Size      , ...
284         'FontName'  , FontName );
285
286     set(ax2,'ticklength',1/3*get(ax1,'ticklength'))
287
288     % Configura Eixo X
289     lx0 = xlabel(ax2,'Tensao do modulo (V)','FontName',FontName,
290     'FontSize',Size,'Interpreter','latex');
291
292     % Exportar Figura
293     if salvar == 1
294         set(gcf,'PaperUnits','centimeters',...
295         'PaperPosition',[0 0 ImageSizeX ImageSizeY],...
296         'PaperSize',[ImageSizeX ImageSizeY]);
297         location = fullfile('.', '10) Tratada\PDF');
298         location = strcat(location,'/');
299         tic
300         print('-dpdf','-painters',strcat([location
           FileLabel],'.pdf'),strcat('-r',num2str(ImageDPI)))
301         el_time = toc;
302         fprintf('\nOkk! (%.2f seconds to save)\n',el_time);
303     else
304         fprintf('\nOkk! Save option disable!!!\n');
305     end
306
307     V_curva = V_novo;
308     I_curva = I_novo;
309     tag_status = 'finalizada';
310
311     % Salva as variaveis tratadas em um novo arquivo MAT
312     nomeArquivo = fullfile('.', '10) Tratada', sprintf('IxV_%s_S%d_T%d.mat', modelo, Ir, T));
313     save(nomeArquivo, 'I_curva', 'V_curva','Ir_med','T_med',

```

```
314     'Isc_med','Voc_med','modelo','tag_status');  
315     disp(['Variaveis salvas em ' nomeArquivo]);  
316  
317 end
```