Federal University of Amazonas (UFAM)
Computing Institute (IComp)
Graduate Program in Informatics (PPGI)

Automated Verification of Stand-alone Solar Photovoltaic Systems

Optimal Sizing and Project Validation

Alessandro Bezerra Trindade

Manaus
February 2020

Federal University of Amazonas (UFAM)
Computing Institute (IComp)
Graduate Program in Informatics (PPGI)

ALESSANDRO BEZERRA TRINDADE

Automated Verification of Stand-alone Solar Photovoltaic Systems

Optimal Sizing and Project Validation

A thesis submitted in partial fulfillment
of the requirements for the Degree of Doctor
of Philosophy in Informatics
Graduate Program in Informatics
Federal University of Amazonas

Supervisor: Prof. Lucas Carvalho Cordeiro, Ph.D.

MANAUS
2020

Ficha Catalográfica elaborada por Suely Oliveira Moraes – CRB 11/365

Trindade, Alessandro Bezerra.
T833a Automated verification of stand-alone solar photovoltaic systems: optimal sizing and project validation / Alessandro Bezerra Trindade. Manaus: UFAM, 2020.

130 fl.: il.: 21 cm

Orientador: Prof. PhD. Lucas Carvalho Cordeiro

Tese (Doutorado em Informática) - Universidade Federal do Amazonas, Programa de Pós-Graduação em Informática.

1. Verificação formal. 2. Model checking. 3. Sistemas solares fotovoltaicos I. Cordeiro, Lucas Carvalho (Orient.) II. Universidade Federal do Amazonas. III. Título.

CDU 004.02:621.31(043.2)

FOLHA DE APROVAÇÃO

"Automated Verification of Stand-alone Solar Photovoltaic Systems: Optimal Sizing and Project Validation"

ALESSANDRO BEZERRA TRINDADE

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Lucas Carvalho Cordeiro - PRESIDENTE

Prof. Edjard de Souza Mota - MEMBRO INTERNO

Prof. Raimundo da Silva Barreto - MEMBRO INTERNO

Prof. Sidelmo Magalhães Silva - MEMBRO EXTERNO

Prof. Wilson Negrão Macêdo - MEMBRO EXTERNO

Manaus, 31 de Janeiro de 2020

Abstract

With decreasing costs and increasing performance, the deployment of renewable energy systems is now growing faster than in the past decade. In 2017, for the first time, the number of people without access to electricity dipped below 1 billion, but trends in energy access still fall short of global goals. Particular attention is given to stand-alone solar photovoltaic systems in rural areas or where grid extension is unfeasible. Tools to evaluate or to size electrification projects are available, but they are based on simulations that do not cover all aspects of the design space. However, the use of formal methods to model and validate any system has grown with time, mainly to find bugs in sophisticated hardware and software systems: they aim to establish system correctness with mathematical rigor. The use of formal methods in electrical systems is a new subject, with published research spanning only the last four years. Moreover, the use of automated synthesis in order to obtain optimal sizing of solar photovoltaic systems has never been done before. This thesis marks the achievement of two major goals: first, the application of software model checking to verify formally the design of a stand-alone solar photovoltaic system, including solar panel, charge controller, battery, inverter, and electric load; second, a sound, automated approach to obtaining optimal sizing of stand-alone photovoltaic systems using program synthesis. For the formal verification, we used case studies from real photovoltaic systems deployed in five different sites, ranging from 975 W to $1,300$ W, in order to evaluate the proposed approach and to compare it with a specialized simulation tool. Different verification tools are evaluated also, in order to compare performance and soundness. Data from practical applications show the effectiveness of our proposed approach, where specific conditions that lead to failures in a photovoltaic solar system are detailed only by the automated verification method. In addition, for the use of program synthesis, we propose a variant of the counterexample guided inductive synthesis (CEGIS) approach. This variant has two phases linking the technical and the cost analysis. First, we synthesize a feasible candidate based on power reliability, but which may not attain the lowest cost. Second, the candidate is then verified iteratively with a lower bound cost via symbolic model checking. If the verification step succeeds, the lower bound is adjusted; if it fails, a counterexample provides the optimal solution. The proposed synthesis method is novel and unprecedented as it streamlines the design of photovoltaic systems. Experimental results using seven case studies demonstrate that our synthesis method can produce optimal system sizing within an acceptable run-time. We also present a comparison with a specialized simulation tool over real photovoltaic systems in order to show the effectiveness of our approach, which can provide a more detailed and accurate solution than the simulation tool.

Keywords: Formal verification; automated verification; model checking; program synthesis; electrical systems; solar photovoltaic systems.

Resumo

Com custos decrescentes e com melhoria de desempenho, a implantação de sistemas de energia renovável está crescendo cada vez mais rapidamente no mundo. Em 2017, pela primeira vez, o número de pessoas sem acesso a eletricidade ficou abaixo de 1 bilhão, mas os dados quanto à universalização do acesso a energia ficaram aquém das metas globais. Particular atenção é dada aos sistemas isolados solares fotovoltaicos em áreas rurais ou onde as elevadas extensões tornam a rede inviável. Ferramentas para avaliar ou dimensionar projetos de eletrificação estão disponíveis, mas elas são baseadas em simulações que não cobrem todos os aspectos do espaço de projeto. Por outro lado, o uso de métodos formais para modelar e validar qualquer tipo de sistema está crescendo com o tempo, principalmente para encontrar "bugs" em sistemas complexos de *hardware* e *software*: seu objetivo é estabelecer a corretude do sistema com rigor matemático. O uso de métodos formais em sistemas elétricos é um assunto recente, com pesquisas sendo publicadas apenas nos últimos quatro anos. Além disso, a síntese automatizada nunca foi usada antes para se obter um ótimo dimensionamento de sistemas solares fotovoltaicos. Esta tese marca duas conquistas principais: (1) a primeira aplicação de verificação de modelos de *software* para verificar o projeto de um sistema isolado solar fotovoltaico, incluindo painel solar, controlador de carga, bateria, inversor e carga elétrica; e (2) uma abordagem confiável e automatizada para obter o dimensionamento ótimo de sistemas fotovoltaicos usando a síntese de programas onde cada componente e função de um sistema solar fotovoltaico é descrito, incluindo suas propriedades, e o modelo comportamental que representa o dimensionamento ótimo é sintetizado automaticamente. Relacionado à verificação formal, estudos de caso de sistemas fotovoltaicos reais instalados em cinco localidades diferentes são usados para avaliar a abordagem proposta e para compará-la com ferramenta de simulação especializada. Diferentes ferramentas de verificação são avaliadas também, a fim de comparar o desempenho e a confiabilidade dos resultados. Dados de aplicações práticas mostram a eficácia da abordagem proposta, onde condições específicas que levam a falhas em um sistema solar fotovoltaico são detalhadas apenas pelo método de verificação automatizado. Além disso, em relação ao uso da síntese de programas, propõe-se uma variante do método de síntese indutiva guiada por contraexemplos (CEGIS), com duas fases bem definidas: primeiro, ele sintetiza o dimensionamento de sistemas fotovoltaicos baseados em confiabilidade de energia, mas que pode não alcançar o menor custo; segundo, a solução proposta é então verificada iterativamente com um limite inferior via verificação de modelo simbólico. Se a etapa de verificação não falhar, o limite inferior será ajustado; e se falhar, o contraexemplo é fornecido com o dimensionamento ótimo, vinculando assim a resposta técnica da primeira fase à análise de custo da segunda fase. Os dados de equipamentos comerciais de

diferentes fabricantes são fornecidos ao mecanismo de síntese e as soluções candidatas são derivadas da análise financeira do dimensionamento obtido. O método de síntese proposto é novo e sem precedentes para simplificar o projeto de sistemas fotovoltaicos. Resultados experimentais usando sete estudos de caso mostram que o nosso método de síntese é capaz de produzir em um tempo de execução aceitável o dimensionamento ótimo do sistema fotovoltaico, e um comparativo com uma ferramenta de simulação especializada e sistemas fotovoltaicos reais mostra a eficácia da abordagem adotada.

Palavras-chave: Verificação formal; verificação automatizada; verificação de modelos; síntese de programa; sistemas elétricos; sistema solar fotovoltaico

Acknowledgements

(It is not possible to arrive at this point without the help of several people. I hope I have not forgotten anyone and do wrong to none).

I would like to thank God (for the gift of life); Solange and Louise (without the patience and support of my family I could not be here); my parents and grandparents; Lucas (for having believed in me and for his patience); the Electricity Department at UFAM (for the support); Professors Feitosa, Marco Cristo and Barreto from IComp (support and advice); IT support team from IComp (Virtual Machine maintenance); the IComp administrative team (for their administrative support); the financial support from FAPEAM, the British Council/Newton Fund and Fundação Amazonas Sustentável; Davide Poggio from the University of Sheffield for the HOMER Pro license (Research England QR GCRF Institutional Allocation); and the Coventry University team (Elena, Kojo, and Ross for the opportunity to use field deployed photovoltaic solar systems).

Contents

List of Figures

List of Tables

List of Abbreviations, Units and Nomenclature

a Ideality or quality factor
AC Alternating current
ANEEL National Electricity Agency (from Brazil)
C_{10} 10h of rated battery capacity
C_{bank} Total capacity of the battery bank
CBMC C Bounded Model Checker
CEGIS Counterexample Guided Inductive Synthesis
CFA Control-Flow Automata
$cos\varphi$ Power factor of the inverter
CPA Configurable Program Analysis
CPAchecker Configurable Program Analysis Checker
CPES Cyber-Physical Energy Systems
CPS Cyber-Physical Systems
DC Direct current
DOD Maximum depth of discharge ($DOD = 1 - SOC$), considering the battery autonomy
DOD_{day} Depth of discharge for one day
$E_{consumption}$ Energy consumption
$E_{corrected}$ Energy corrected
E_p Energy produced by one panel
EQ Experimental Question
ESBMC Efficient SMT-based Bounded Model Checker
G Solar irradiance
G_{ref} Solar irradiance at STC ($1000W/m^2$)
GEF Global Environment Facility
HDI Human Development Index
IEA International Energy Agency
IRR Internal Rate of Return
I Current
$Insolation$ Also called solar irradiation or solar exposure (kWh/m^2 per day)
I_0 Reverse saturation current
I_{AC} Current that the inverter supply to the house
$I_{c,min}$ Charge controller minimum current
I_{DC} Current required by the inverter from the DC source
$I_{min,DCbus}$ Minimum DC bus current to be considered during sizing
I_{in} Input current
I_{load} Current that feeds the load

I_{out} Output current
I_{ph} Photocurrent
$I_{ph,ref}$ Photocurrent at STC
I_{pv} Current from the PV system
$I_{total,PVpanels}$ Total current supplied by the PV panels array
I_{sc} Short-circuit current
$I_{sc,amb}$ Ambient short-circuit current
$I_{sc,ref}$ Short-circuit current at STC
k_B Boltzmann constant ($1.3806503 \times 10^{-23} J/K$);
LCC Life Cycle Cost
LCOE Levelized Cost of Energy
LLP (or LPLP) Loss of Load probability
LPSP Loss of Power Supply Probability
$MAX_{AC,ref}$ Peak power that the inverter can support
MPPT Maximum Power Point Tracking
N Number of series-connected cells
N_Btotal Number of sized batteries
N_{BP} Number of sized parallel batteries
$N_{BP}min$ Total minimum number of parallel batteries
N_{BS} Number of sized series batteries
$N_{BS}min$ Total minimum number of series batteries
NOCT Nominal Operating Cell Temperature
NPC Net Present Cost
N_{PS} Number of sized panels in series
N_{PSmin} Total number of panels in series
N_{PP} Number of sized panels in parallel
N_{PPmin} Total umber of panels in parallel
N_{TP} Number of sized panels
N_{TPmin} Total minimum number of solar panels needed
PV Photovoltaic
P Power
$P_{min,panels}$ Minimum power that the PV panels must supply, used during PV sizing
P_{in} Input power
P_{out} Output power
P_{surge} Maximum peak power demanded by house for few seconds
q Absolute value of the electron's charge ($-1.60217646 \times 10^{-19} C$)
QF quantifier-free (formula)
RQ Research Question
REEEP Renewable Energy and Energy Efficiency Partnership

SAT Satisfiability or Boolean satisfiability problem
SAT When the verification tool returns a failure
SDGs Sustainable Development Goals
SMT Satisfiability module theories
SOC Battery state of charge
STC Standard Test Condition (reference state)
T Temperature
T_{air} Ambient temperature
T_{ref} Temperature at STC ($= 298.15K$ or 25^oC)
UNEP United Nations Environment Programme
UNSAT Condition when verification tool succeeds (could not reach an error)
V Voltage
V_{AC} Voltage that the inverter supplies to the house
V_c Battery charge voltage
VC Verification conditions
V_d Battery discharge voltage
V_{DC} Input voltage to the inverter delivered by the DC source
V_{in} Input voltage
V_{oc} Open-circuit voltage
$V_{oc,ref}$ Reference open-circuit voltage at STC
V_{out} Output voltage
V_{system} DC voltage of the bus
$V_{total,PVpanels}$ Total voltage supplied by the PV panels array
V_T Thermal voltage ($V_T = k_B T/q$)
ΔT Temperature variation ($\Delta T = T - T_{ref}$)
μ_I Short-circuit current temperature coefficient (A/K)
μ_V Open-circuit voltage temperature coefficient (V/K)
η_b Battery efficiency
η_c Charge controller efficiency
η_i Inverter efficiency
η_p PV panel efficiency

Chapter 1

Introduction

Energy production, distribution, and optimization are all Cyber-Physical Systems (CPS) problems (UC, 2016). CPS are engineered systems, which are built from, and depend on, the seamless integration of computational and physical components (NSF, 2015; CHAVES et al., 2019; CORDEIRO; FILHO; BESSA, 2019). During operation, these components must frequently adapt to the operating environment changes (dynamics of the physical processes) faced at run-time. They must be able to continue to behave in a controlled and safe way, thus posing novel technical challenges for the software engineering of services and applications for CPS (METZGER; POHL, 2014). Software pervasiveness in CPS places new challenges and demands new paradigms for software design, particularly in the face of highly dynamic environments, rapidly changing requirements, unpredictable and uncertain operating conditions demand new paradigms for software design (FILIERI et al., 2015).

While some existing research efforts do aim to enhance and optimize the software development processes for CPS, further investigation and discussion of better and more effective models are still needed in practice (AL-JAROODI et al., 2016). Among the opportunities for enhancements in the development processes for CPS software, there exists the need to develop new techniques and tools to support CPS requirement gathering and analysis to synthesize *correct-by-construction* implementations of CPS. These techniques have to deal with predefined requirements enforced by nature and inherited constraints of the target CPS. Besides, they should be able to provide verification and validation mechanisms for completeness, correctness, and consistency (AL-JAROODI et al., 2016). Uncertainty and variability, at the same time, can be dealt with by formal verification (SOFTWARE; NESSI, 2014).

Among the many CPS systems, in this thesis, we will focus specifically on energy generation by stand-alone solar PV systems. The lack of access to clean and affordable energy is considered a core dimension of poverty (HUSSEIN; LEAL FILHO, 2012).

Nevertheless, the world is progressing; in particular, the number of people without electricity access fell below the 1 billion thresholds for the first time in 2017 (IEA, 2018). The share of people without access to electricity from Africa is 58%, while 19% of the share comes from developing Asia, and 31% from Latin America (IEA, 2018). Numbers from Brazil show the aim to electrify 270 isolated areas and 2.7 million people by 2023 (EPE, 2018). Furthermore, there is a close relationship between the lack of energy and the low HDI (Human Development Index) of those localities (COELHO et al., 2015). It follows that increased access to energy allows economic growth and poverty alleviation (KAREKESI; LATA; COELHO, 2006).

The use of power generation technology with renewable energy sources is developing rapidly due to industrial development (YATIMI; AROUDAM, 2015). Renewable energy leads to advances all over the world by protecting the environment: it is clean (low greenhouse gas emissions), operates silently, long-lasting, low maintenance costs, zero fuel costs and an inexhaustible supply (NOROOZIAN; GHAREHPETIAN, 2013). Renewable energy, and particularly power generated from solar energy using photovoltaic (PV) panels, has emerged as an alternative to fossil or nuclear fuel generation.

Renewable sources of energy include hydro, wind, and solar PV. According to (SEIA, 2016) and (CHAUHAN; SAINI, 2015), the sun is the most abundant source of renewable energy on earth. In solar PV systems, solar radiation is captured from the sun and turned into electricity using solar PV cells made of silicon and other materials.

Nothing more than a niche market only a few years ago, solar PV systems have become a mainstream electricity provider, with an approximate 50% increase (or 100.9 GW)in new PV installations from 2016 to 2017 (EPIA, 2017), although this growth is not driven by stand-alone systems. However, in order to provide universal electricity, decentralized systems led by solar photovoltaic (PV) in off-grid and mini-grid systems will be the lowest-cost solution for three-quarters of the additional connections needed. Grid extension will be the standard, especially in urban areas (IEA, 2018).

The PV cell in a solar PV system, as defined in (RAWAT; KAUSHIK; LAMBA, 2016), is a semiconductor device, which directly converts solar radiation directly into electrical energy. Apart from the PV modules, the PV systems consist of a battery bank, controller, and inverter, plus the load.

The increase in the number of PV systems installed all over the world underscores the need for proper modeling of the equipment and simulation tools for researchers and practitioners involved in their application.

Moreover, the optimum sizing of these devices is essential for reliable operation. Therefore, these systems need to be designed under the site, the land area available, load requirement, load pattern, environmental conditions, and economics in order to utilize available resources efficiently and economically (RAWAT; KAUSHIK; LAMBA, 2016).

1.1 The Problem

Energy access is an essential issue because it can put together economic growth, human development, and environmental sustainability. Energy has long been recognized as essential for humanity to develop and thrive, but the adoption in 2015 by 193 countries of a goal to ensure access to affordable, reliable, sustainable and modern energy for all by 2030, as part of the new United Nations Sustainable Development Goals (SDGs), marked a new level of political recognition (IEA, 2018).

The International Energy Agency (IEA) revealed that from 2000 to 2016, nearly all of those who gained access to electricity worldwide did so through new grid connections, mostly with power generation from fossil fuels. Over the last five years, however, renewable sources have started to gain ground, as have off-grid and mini-grid systems. By 2030, renewable energy sources power over 60% of new access, and off-grid and mini-grid systems provide the means for almost half of new access (IEA, 2018). Moreover, if we focus on the Amazonas State, Brazil, where around 5% of the population in 2018 was isolated in 2,261 communities (or 41,167 houses), without electrical energy or even road access, using just boats by rivers, therefore energy is a fundamental issue.

Under this scenario, it is crucial to have very well designed PV systems that will not fail when installing in the field and at the lowest acquisition cost possible.

In order to address different aspects of a solar PV project, there is a variety of public domain and commercial software solutions available, such as RETScreen, HOMER, PVWatts, SAM, and Hybrid2 (PRADHAN et al., 2015; SWARNKAR; GIDWANI; SHARMA, 2016; DOBOS, 2014; BLAIR et al., 2014; MILLS; AL-HALLAJ, 2004); and even general-purpose simulation tools such as PSpice, and the MATLAB/Simulink package (GOW; MANNING, 1999; BENATIALLAH et al., 2017). According to (BROOKS; DUNLOP, 2013), the capabilities of these tools range from simple solar resource and energy production estimation to site survey and system design tools, and sophisticated financial analysis software (with optimization). Some tools also provide support to rebate program applications and tax incentives (specific to each country or region). In contrast, other programs and worksheets focus on the technical aspects of system sizing and design. Manufacturers and integrators

also have their proprietary software to perform system sizing (ZHOU et al., 2010), with the drawback of indicating only their products among the possibilities of choice, which restricts their solution. However, public domain and commercial software and even proprietary tools are based on running experiments in simulation models. Simulation has the advantage of being cheap (if compared to testing in real systems) and can be employed before the system design is concluded. However, it has the drawback of incomplete coverage since the verification of all possible combinations, and potential system failures are unfeasible (CLARKE; HENZINGER; VEITH, 2018).

Formal methods based on model checking offer a great potential to obtain a faster and more effective verification in the design process (CLARKE; HENZINGER; VEITH, 2018), including, for example, applications to CPS (ABATE et al., 2017a; ABATE et al., 2017b; BESSA et al., 2017; CHAVES et al., 2017; ABATE et al., 2017), with behavior that is, in principle, well determined. Any system type can be specified as computer programs using mathematical logic, which constitutes the intended (correct) behavior; one can then try to produce a formal proof or otherwise establish that the program meets its specification (GADELHA; ISMAIL; CORDEIRO, 2017). User or project requirements can be added during the creation of the formal model to be verified (TRINDADE; CORDEIRO, 2016; TRINDADE et al., 2017). This research area is referred to as formal methods (CLARKE; EMERSON; SIFAKIS, 2009), which aim to establish system correctness with mathematical rigor. In recent decades, research in formal methods has led to the development of up-and-coming verification techniques that facilitate the early detection of flaws in order to ensure the correctness of the system, including programming languages such as C/C++ and Java (GADELHA et al., 2018; RAMALHO et al., 2013; CORDEIRO et al., 2018).

Model-based verification techniques are based on models that describe possible system behavior in a mathematically precise, and unambiguous manner. Thus, problems such as incompleteness, ambiguities, and inconsistencies, which are generally discovered only in the later stages of the design, can be detected in advance, as described by (TRINDADE; CORDEIRO, 2016; TRINDADE et al., 2017). Model-checking algorithms can then verify the system model by systematically exploring all its states to check whether the given system meets the requirements.

Optimization of PV systems is not a recent topic; since the 1990s different techniques have been developed and evaluated using a wide variety of criteria to find the ultimate combinations for design parameters based on intuitive, numerical, and analytical methods (APPLASAMY, 2011). The ideal combination for any PV system is made by the best compromise between two considered objectives, which are *power reliability* and *system cost* (ALSADI S.; KHATIB, 2018).

However, formal methods based on *symbolic model checking* applied to synthesize PV systems have not been further explored in the literature, which could offer an excellent opportunity to obtain a more effective design process for PV systems (CLARKE; HENZINGER; VEITH, 2018).

1.2 Objectives

The main objective of this thesis is thus to prove that it is possible to use a formal verification method to validate the design and to use a formal synthesis method to optimize the sizing of stand-alone solar PV systems. Secondarily, we aim to develop two tools written in ANSI-C and platform-independent, based on the new proposed techniques to evaluate different state-of-the-art symbolic software verifiers (and solvers). We also aim to compare the experimental results with the simulation tool and with data from real PV systems installed in riverside communities, thereby evaluating correctness and performance.

The originality of this thesis lies in the creation of an automated verification technique for solar PV systems. The research led to the creation of two software solutions to carry out computational experiments and to compare with commercial tools and real data from systems installed in the field. One software solution to perform automated verification and validate the solar PV systems behavior, the other to produce the optimal sizing of solar PV systems through an automated synthesis program.

In this study, a mathematical model of each component of a stand-alone PV system: the solar panel, charge controller, batteries, inverter, and electrical load, is used for formal verification. The intended behavior of each system component can be verified and observed with the support of formal models, as a joint operation of the components, which in this case represents the operation of the solar PV system itself. The project requirements, such as battery autonomy and power demand, as well as the weather conditions, such as solar irradiance and ambient temperature, are provided for the proposed tool and automatically verified during the formal process. The model checking tool reports in which conditions a system does not meet the user requirements. A key benefit to this approach is that it helps in the detection of flaws in the design phase of system development, thereby considerably improving system reliability (AKRAM; NIAZI, 2018).

The proposed technique is applied through an algorithm implemented using the C programming language. We employed three state-of-the-art model checkers to verify PV designs formally, thus aiming to evaluate performance and correctness. The C Bounded Model Checker (CBMC) (KROENING; TAUTSCHNIG, 2014), the Efficient SMT-based Bounded Model Checker (ESBMC) (GADELHA et al., 2018),

and the Configurable Program Analysis Checker (CPAchecker).

Concerning the **automated synthesis** technique and optimal sizing of stand-alone PV systems, this thesis presents the development of a variant of counterexample guided inductive synthesis (CEGIS) that uses commercial equipment data, user requirements and weather conditions as input. Given a correctness specification σ, the proposed method uses σ as a starting point and then iteratively produces a sequence of candidate solutions that satisfy σ for power reliability. In particular, in each iteration, we synthesize the sizing of stand-alone PV systems, which may not attain the lowest cost. The candidate solution is then verified via symbolic model checking with a lower bound that serves as the minimum cost of reference; if the verification step does not fail, the lower bound is adjusted. If it fails, then a counterexample is provided with an optimal sizing that meets power reliability and system cost requirements. Note that in this study, the focus is not on new criteria or even optimization objectives. Instead, the novelty lies in the practical approach to the pursuit of the optimal solution of PV systems using formal methods, which outperforms existing state-of-the-art simulation tools.

We aim to present new techniques that will guarantee a good design of stand-alone solar PV systems, that do not cause a drop in energy supply after deployment, and that these systems are designed at the lowest possible cost.

1.3 The Solution: an Outline

Our approach deals with the theoretical and pragmatic aspects of using model checking in stand-alone solar PV systems to tackle two issues: validation of a designed system and optimal sizing.

Validation is a important issue when a sized system must be evaluated in order to decide if it meets the design requirements and user needs. In other words, it is a way to guarantee that a sized PV system will not fail (stopping to supply energy to a house, for example) after deployment at the field.

Optimization is important when there exist (just) the requirements of the house to be electrified. In our particular case, we need a optimal PV system solution, in technical and economical terms, that can show us a list of equipment to be bought and how it will be connected together (configuration).

We thus develop algorithms and the corresponding tools; we then evaluate them through several case studies, in comparison with specialized simulation tools, supported by real data from installed PV systems.

Fig. 1.1 outlines the proposed solution to perform the validation of stand-alone solar PV systems via automated verification. Fig.1.2 illustrates the proposed solution to obtain the optimal sizing of stand-alone solar PV systems via automated synthesis. We pictured here only a summary of each phase of our proposed approach, highlighting the input parameters, the outputs for the validation, or the optimal size of the solar PV system; more details will be presented in the next chapters. Moreover, the illustrations show the methodological approach concerning comparative result analysis, employing our approaches, the commercial software tool, and data from real PV systems deployed in the field.

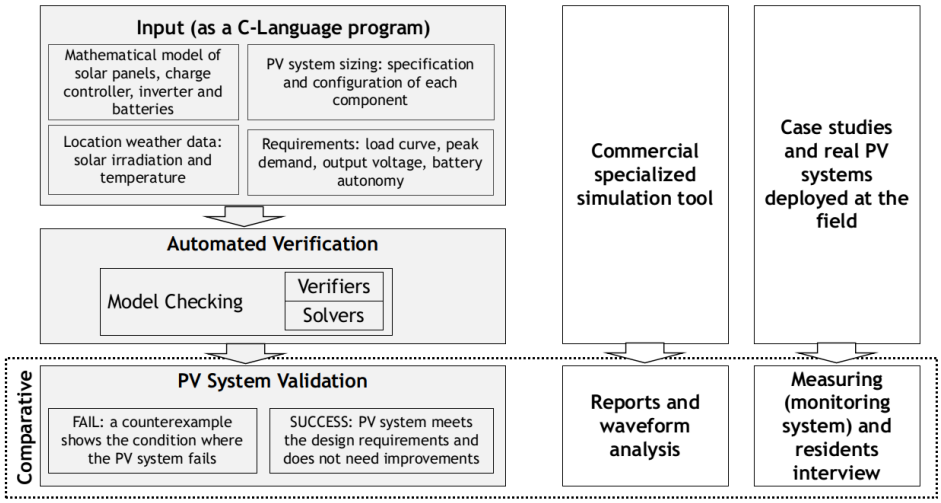


Figure 1.1: Proposed validation of PV systems via automated verification.

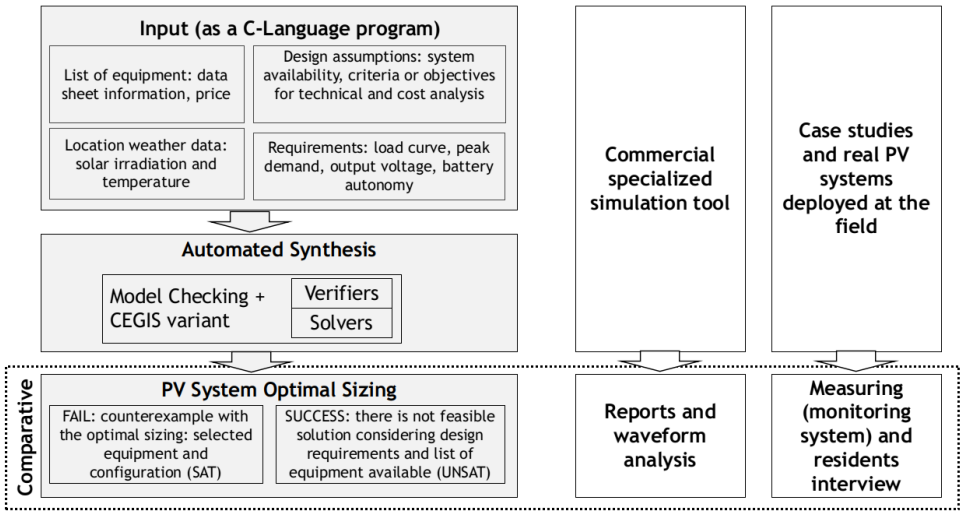


Figure 1.2: Proposed optimal sizing of PV systems via automated synthesis.

Note that it is out of our scope to perform code modification on the verifiers or the solvers used during the automated verification or automated synthesis steps of our approach. Here we intend to create front-end applications for the verifiers and

solvers (which are the back-end). That decision is based on the fact that we can be verifier-independent, and evaluate different back-ends, now and in the future, always searching for the best performance and soundness.

1.4 Contributions

Concerning the proposed **automated verification** technique, this thesis makes three main contributions:

- **First**, we propose an algorithm written in the C programming language which implements the automated verification method to formally check the sizing and the operation of a given stand-alone PV system;
- **Second**, we evaluate the verification technique by comparing three state-of-the-art model checkers in five real case studies; and
- **Third**, experimental results show that this approach can find subtle design errors in stand-alone PV systems not easily detected by other approaches based on simulation.

In the area of **automated synthesis**, our study makes a further three original contributions:

- **First**, it is the first application of a sound and automated formal synthesis approach which can provide accurate results for optimal sizing of stand-alone PV systems;
- **Second**, we propose a variant CEGIS method of synthesis with striking differences of how the Synthesize and Verify phases from the original CEGIS work (without solution candidate vector and using an incremental and iterative loop to reach the optimal cost solution); and
- **Third**, experimental results in seven case studies show that our approach qualitatively outperforms state-of-the-art simulation tools: our solution is far more detailed and closer to real PV systems than solutions presented by simulation.

1.5 Related Work

In this section, we discuss initially the use of formal methods in electrical systems in general since the optimization of PV sizing is currently not obtained by formal methods or even program synthesis. Further, at the end of this section, we do a

parallel about how software simulation tools address the issues related to solar PV systems.

The conversion of the traditional power grid into a smart grid, a fundamental example of a CPS, raises many issues that require novel methods and applications. In 2012, a Chinese smart grid implementation was considered as a case study to address the verification problem for performance and energy consumption (YÜKSEL et al., 2012). The authors employed a stochastic model checking approach and presented a modeling and analysis study using PRISM, which is a probabilistic model checker (KWIATKOWSKA; NORMAN; PARKER, 2011). The focus of this study was on how CPSs integrate information and communication technology functions to the physical elements of a system for monitoring and controlling purposes. The authors did not focus on power generation or even solar PV systems.

In 2015, an automated approach for applying Monte-Carlo simulation to power system protection schemes presented limitations of incomplete coverage of all possible operating conditions (SENGUPTA; MUKHOPADHYAY; SINHA, 2015). The authors proposed an automated simulation-based verification technique to verify the correctness of protection settings efficiently using a hybrid automata-temporal-logic framework. The initial focus was on relay operations and test-case generation to ensure the early detection of design errors. However, this study was limited to power system protection and did not deal with electricity generation or even solar PV systems.

Other related studies from 2015 include a framework named Modana to achieve an integrated process from modeling with SysML/MARTE to analysis using statistical model checking for CPS in terms of non-functional properties such as time and energy (CHENG et al., 2015). In order to demonstrate Modana's capability, the authors modeled energy-aware buildings as a case study and discussed the analysis of energy consumption in different scenarios. The focus here is on smart buildings and HVAC (heating, ventilation, and air conditioning) systems. This research, however, did not address the design and verification of solar PV systems.

In 2017, a researcher suggested the application of formal methods to verify and control the behavior of computational devices, interacting over a shared and smart infrastructure (ABATE, 2017). The author discussed the aggregation of large populations of thermostatically-controlled loads and PV panels, and the similar problems of energy management in smart buildings, of demand-response on smart grids, and respectively of frequency stabilization and grid robustness. The focus was on controlling the behavior of components, thereby verifying the given smart grid as a "system of systems" within the context of "internet of things". The author,

however, used an approximate model checking of stochastic and hybrid models.

In 2018, a verification methodology was proposed for the Cyber-Physical Energy Systems (CPES) with applications to PV panels and its distributed powerpoint tracking (DRIOUICH; PARENTE; TRONCI, 2018). This approach relied on representing the unpredictable behavior of the environment to cover all possible feasible scenarios. The simulation results obtained by JModelica covered the system's complete dynamic behavior; however, almost three days of computer effort to verify the design space of a single operational hour of the PV panels' behavior made it clear that time was an issue. This related study did not include any other components of a stand-alone solar PV system.

Another study from 2018 presented an approach to modeling smart grid components using a formal specification. The authors used a state-based formal specification language named Z; they demonstrated the application of Z to four smart grid components (AKRAM; NIAZI, 2018). The formal specification presented can be considered a first step towards modeling smart grids using formal methods. The starting point of this study was that a smart home could be considered an integrated system consisting of various objects and systems, which communicate and interact with each other. This approach is based on Petri nets and works under the assumption that modeling smart homes lead to a clear understanding of the overall behavior of the smart grids.

However, prior studies did not deal with formal verification of a complete stand-alone PV system (with batteries, charge controller, and power inverter) or even solar PV systems optimization. Formal methods based on *symbolic model checking* and its application to synthesize PV systems are still unexplored in the literature. Moreover, it is precisely on these gaps that this thesis is focused on.

Software simulation tools, when compared to the proposed automated verification techniques, has a common part of the approach that is the use of mathematical models in order to describe how the equipment or the PV system items behave itself over the time or when a specific input is applied to the system. However, simulation tools have a focus dependence with the choice of **input** whereas the proposed techniques focus on the mathematical rigor of the solution and the search of an **output** that falsifies the expected requirements of the system (CLARKE; HENZINGER; VEITH, 2018). Therefore, simulation software focus on the inputs whereas our techniques focus on the outputs.

1.6 Thesis Organization

This introduction has outlined the context, motivation, and problem addressed by this thesis and the objectives, solutions, and contributions of the research. The remaining chapters are organized as follows:

Chapter ‘ Background’ presents the theoretical basis of formal methods, formal verification, automated verification, solar PV systems, design and validation of PV systems, program synthesis, and the mathematical modeling.

Chapter ‘ Automated Formal Verification of Stand-Alone Solar PV Systems’ presents the automated formal verification technique, the experimentation details, and the results obtained with the tool created using this new method of PV systems validation.

Chapter ‘ Optimal Sizing of Stand-alone Solar PV Systems via Automated Formal Synthesis’ presents the automated synthesis technique from computer science and its application to obtain the optimal sizing of stand-alone solar PV systems, with details of the experimentation, and the results using the tools created to compare this new technique with a simulation tool and from data collected from fieldwork.

In the ‘ Conclusions’, we present the main contributions, future work directions and concluding remarks.

Appendix A presents a list of papers that we submitted to international journals and conferences on the topic of the automated verification method. It covers the years 2015-2019, restricted to the Ph.D. process.

Appendix B depicts the tools created during the Ph.D. process to implement and validate the two scientific methods of the thesis (and instructions for their use).

Appendix C presents detailed data from all the equipment used during the experimental part of the thesis, covering information from data-sheet, electrical features, brands, and models.

It is worth mentioning that we decided, with the consent of the coordination of the graduate program, that this document would be written in the active voice and entirely in English. The choice of the active voice comes from the choice of the English language, which in the thesis studied was always presented in the active voice. The decision to use the English language stems from the possibility that the research may lead to further developments and activities, in the possibility of reaching a more significant number of people and helping in the internationalization of the Federal University of Amazonas.

Chapter 2

Background

In this chapter we present some concepts related to the intersection of the two areas: computer science methods used to solve electrical engineering problems; in this case, the use of formal verification methodology to perform automated verification and optimal sizing of stand-alone solar PV systems.

First, we introduce the concept of formal verification as background to understanding how a methodology that performs (mainly) bug detection can be used to validate a design sizing or to obtain an optimal solution of PV systems.

And secondly, how a solar PV system can be modeled in order to be validated or optimized by formal verification methodology.

2.1 Formal Methods, Formal Design and Formal Verification

Formal methods are system design techniques that use rigorously specified mathematical models to validate systems, most notably (and known for) software and hardware systems (COLLINS, 1998). In contrast to other design systems, formal methods use mathematical proof as a complement to system testing in order to ensure correct behavior. With increasing scale and complexity, when safety becomes a more important issue, the formal approach to system design offers a better level of insurance.

Formal methods differ from other design systems through the use of formal verification schemes; the basic principles of the system must be proven correct before they are accepted (BOWEN; STAVRIDOU, 1993). Traditional system design has used extensive testing to verify behavior, but testing is capable of only finite conclusions. Dijkstra and others have demonstrated that tests can only show the situations where

a system will not fail, but can say nothing about the behavior of the system outside of the testing scenarios (BENTLEY, 2000). In contrast, a theorem once proven true remains true.

Prior to the 1980s, mainly deductive verification was used as the formal method, with the use of axioms and proving rules to demonstrate the correctness of the system. The original focus was to verify critical systems based on the premise that if the system is important, time must be spent to verify it (LOWRY; DVORAK, 1998). Originally, formal methods were performed by hand.

It is worth pointing out that formal verification does not avoid the need for testing (BOWEN; HINCHEY, 1995). Formal verification can not correct bad assumptions in the design, but it can help to identify errors in reasoning which would otherwise be left unverified. In several cases, engineers have reported finding flaws in systems following a formal review of their designs (KLING, 1996).

A formal design can be summarized as a three step process, as described below (COLLINS, 1998):

- Formal Specification. During the formal specification phase, the engineer rigorously defines a system using a modeling language. Modeling languages are fixed grammars which allow users to model complex structures from predefined types (that are rigorously defined). This process of formal specification is similar to the process of converting a word problem into algebraic notation and helps researchers and engineers to clearly define their problems, goals and solutions. Several engineers who have used formal specifications claim that the clarity that this stage produces is a benefit in itself (KLING, 1996).

- Verification. As stated above, formal methods differ from other specification systems by their heavy emphasis on provability and correctness. In building a system using formal specification, the designer is actually developing a set of theorems about his system. By proving these theorems correct, the formal verification ensures that the modeled system has an intended behavior. Verification is a difficult process, largely because even the simplest system has several dozen theorems, each of which has to be proven. Given the demands of complexity and Moore's law, almost all formal systems use an automated theorem proving tool of some form (COLLINS, 1998). That is the origin of 'automated verification' definition. These tools can prove simple theorems, verify the semantics of theorems, and provide assistance for verifying more complicated proofs, with feedback about the trace of the error (in order to correct the system and the specification of it).

- Implementation. Once the model has been specified and verified, it is implemented by converting the specification into code.

2.2 Project Validation, Automated Verification and Synthesis Using Model Checking

It is necessary to keep in mind that validation is the process of determining whether a design meets the needs of the user, whereas verification is the process of determining whether a design meets a set of requirements, specifications, and regulations.

If the requirements, specifications, and regulations are given in a formal language, then it may be possible to automate verification.

Simulation may also be used for validation, but it raises more problems for verification. In order to use simulation for verification, it is necessary to ensure adequate coverage of operating conditions, scenarios, and system inputs. Testing can also be used for validation, but for the same reasons, it too raises problems for verification.

According to (CLARKE, 2008), verification procedure is an intelligent exhaustive search of the state space of the design. In addition, according to (FOREJT et al., 2011), formal verification is a systematic approach that applies mathematical reasoning to obtain guarantees about the correctness of a system. One successful method in this domain is model checking.

2.2.1 Model Checking

Model checking is an automatic verification technique, as defined by (CLARKE, 2008). Model checking was originally developed for reasoning about finite state of concurrent systems, though nowadays it is mainly used for hardware and software verification, but can be applied to any kind of system.

The process of model checking can be divided in three components: modeling, specification, and verification method.

- In modeling, a model (usually mathematical) of the system is created;
- In specification, usually a list of properties to be satisfied by the system is established, i.e., the requirements, such as reliability to performance, for example;
- It is expressed usually in temporal logic form (CTL);
- The model checking is the verification method itself.

The model checking algorithm can be described as:

- Given the model $'M'$ and a CTL formula ϕ as input;
- The moodel checking algorithm provides all the states of model M which satisfy ϕ;
- It returns YES if ϕ is $TRUE$, or returns NO if ϕ is $FALSE$.

Specifically, in the case of ϕ being $FALSE$, the algorithm returns a counterexample that is useful as a system diagnostic, in order to discover in which situation the model has been violated. (CLARKE, 2008) considers this to be the most important advantage of model checking.

Model checking presents several other advantages: proofs are not needed (the algorithm is not a deductive procedure); there is no problem with partial specifications of the system, logics can easily express many concurrency properties; it is fast (compared to other rigorous methods such as interactive theorem proving). However, there is one major disadvantage in model checking: the state explosion problem.

The model checking problem can be defined as shown in (CLARKE, 2008):

- Let M be a Kripke structure (i.e., a state transition graph);
- f be the specification in temporal logic (a formula);
- Find all states s of M such that $M, s \models f$

Fig. 2.1 shows the structure of a typical model checking system. A preprocessor extracts a state transition graph from a system (program or circuit).

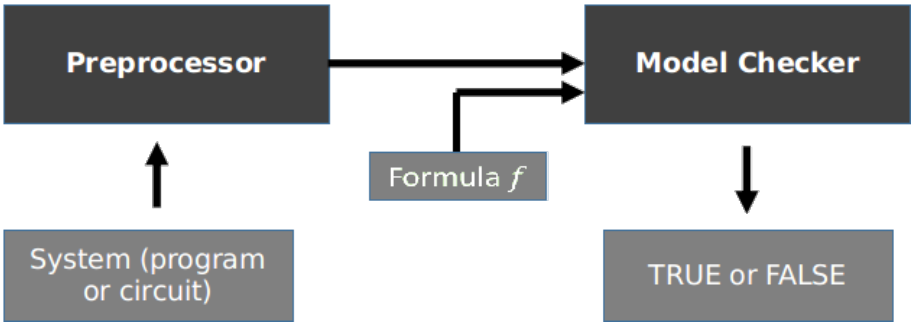


Figure 2.1: Model Checker structure. Source: (CLARKE, 2008).

Here it is worth mentioning that the term "model" is not used here with its dictionary definition. In other words, the problem is not dealing with an abstraction of the actual system under study. Therefore, model can be defined as a (usually finite-state) description of the system to be analyzed.

The Fig. 2.2 shows the process of converting a real system to a model in order to be verified by model checking.

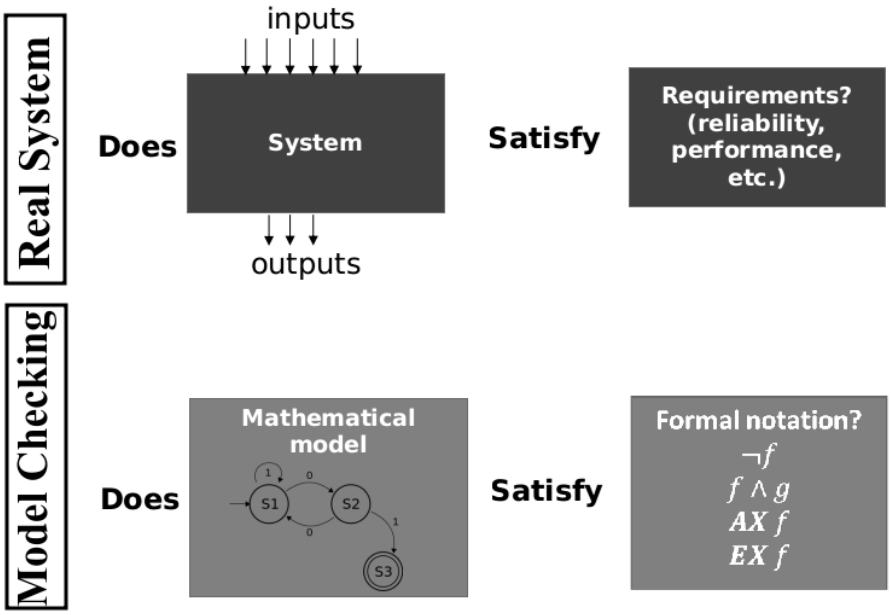


Figure 2.2: From real system verification to model checking. Source: adapted from (CLARKE, 2008).

In order to solve the problem of state explosion, many different techniques have been developed over the last decades. One of the most promising is Bounded Model Checking (BMC).

BMC is a method that checks the model up to a given point in the path length. The BMC algorithms traverse a finite state machine for a fixed number of steps, k, and check whether violation occurs within this bound. It uses fast SAT solvers, where SAT means satisfiability.

A SAT problem, as defined by (CLARKE, 2008), is a problem of determining if there are certain conditions or interpretations that satisfy a given Boolean expression. SAT solvers are used in BMC, such that if there is some Boolean function, the solver would search the model for conditions (value of variables) that would make the formula $TRUE$. If the SAT Solver finds a substitution for the formula/function then the substitute induces a counterexample.

The use of SMT, instead of Boolean Satisfiability SAT from the original BMC, comes as an alternative to overcome limitations of the system's modeling, especially considering that the complexity of these is increasing and the SMT method has a higher level and richer theories than the SAT to represent the models.

In this study we will evaluate three state-of-the-art model checkers to formally verifying and synthesize PV designs w.r.t. user requirements. In addition, other

solvers will be evaluated, according to the feature of each model checker.

2.2.2 CBMC

The C Bounded Model Checker (CBMC) falsifies assertions in C programs or proves that they are safe if a completeness threshold is given (KROENING; TAUTSCHNIG, 2014). CBMC implements a bit-precise translation of a C program, annotated with assertions and with loops unrolled up to a given depth, into a logical formula. If the formula is satisfiable, then a failing execution that leads to a violated assertion exists (KROENING; TAUTSCHNIG, 2014).

CBMC's verification flow can be summarized in three stages:

- Front-end: scans, parses and type-checks C code; it converts control flow elements, such as *if* or *switch* statements, loops and jumps, into equivalent guarded *goto* statements, thus aiming to reduce verification effort;
- Middle-end: performs symbolic execution by eagerly unwinding loops up to a fixed bound, which can be specified by the user on a per-loop basis or globally, for all loops and finally;
- Back-end: supports SAT and SMT solvers to discharge verification conditions (VCs).

Specifically, CBMC comes with a built-in solver for bit-vector formulas that is based on MiniSat. We used this solver during the experimentation in this Ph.D thesis (KROENING; TAUTSCHNIG, 2014).

2.2.3 ESBMC

The Efficient SMT-based Bounded Model Checker (ESBMC) is a bounded and unbounded model checker for C (GADELHA et al., 2018; GADELHA et al., 2019), C++ (RAMALHO et al., 2013), Qt (MONTEIRO et al., 2017), and CUDA (PEREIRA et al., 2017) programs, which supports the verification of LTL properties with bounded traces (MORSE et al., 2015).

ESBMC's verification flow can be summarized in three stages:

- A front-end that can read and compile C code, where the system's formal specification is first handled;
- Preprocessing steps deal with code representation, control flow and unwinding of loops, and model simplification, thereby aiming to reduce verification effort; and finally

- The SMT solving stage, where all constraints (C) and properties (P) of the system are encoded into SMT and checked for satisfiability ($C \wedge \neg P$).

ESBMC exploits the standardized input language of SMT solvers (SMT-LIB[1] logic format) to make use of a resource called *assertion stack* (MORSE, 2015). This enables ESBMC, and the respective solver, to learn from previous checks, thus optimizing the search procedure and potentially eliminating a large amount of formula state space to be searched, because it solves and disregards data during the process, incrementally. This technique is called 'incremental SMT' (SCHRAMMEL et al., 2017) and allows ESBMC to reduce the memory overhead, mainly when the verified system is complex and the computing platform does not have a large amount of memory to deal with the entire design space state. ESBMC in 'incremental SMT' uses only the Z3 solver (MOURA; BJØRNER, 2008).

2.2.4 CPAchecker

Automatic program verification requires a choice between precision and efficiency. The more precise a method, the fewer false positives it will produce, but also the more expensive it is, and thus applicable to fewer programs.

Historically, this trade-off was reflected in two major approaches to static verification: program analysis and model checking. In order to experiment with the trade-off, and in order to be able to set the dial between the two extreme points, Configurable Program Analysis (CPA) provides a conceptual basis for expressing different verification approaches in the same formal setting.

CPA formalism provides an interface for the definition of program analyses. Consequently, CPAchecker provides an implementation framework that allows the seamless integration of program analyses that are expressed in the CPA framework. The comparison among different approaches in the same experimental setting is intended to be easy and the experimental results are expected to be more meaningful (BEYER; KEREMOGLU, 2011). As to the architecture, the central data structure is a set of control-flow automata (CFA), which consists of control-flow locations and control-flow edges.

The CPA framework provides interfaces to SMT solvers and interpolation procedures (BEYER; KEREMOGLU, 2011). Currently, CPAchecker uses MathSAT as SMT solver; and CSIsat and MathSAT as interpolation procedures (BEYER; KEREMOGLU, 2011).

In this thesis, CPAchecker will be configured as a bounded model checker.

[1] http://smtlib.cs.uiowa.edu/

2.2.5 CEGIS and Program Synthesis

Program synthesis addresses an age-old problem in computer science: can a computer program itself? (BORNHOLT, 2019). Before the computer can automatically generate a program, it is necessary to give it a specification of what the program should do. The specification needs to describe the program's desired behavior to ensure that the program does what it is intended.

The basic idea of program synthesis is to automatically construct a program P that satisfies a correctness specification σ. In particular, program synthesis is automatically performed by engines that use a correctness specification σ as starting point, and then incrementally produce a sequence of candidate solutions that partially satisfy σ (ABATE et al., 2017a). As a result, a given candidate program p is iteratively refined, in order to match σ more closely.

CEGIS represents one of the most popular approaches to program synthesis that are currently in use for CPS (ABATE et al., 2017a), whose basic architecture is illustrated in Figure 2.3 and has close connections to algorithmic debugging using counterexamples and abstraction refinement (ALUR et al., 2013).

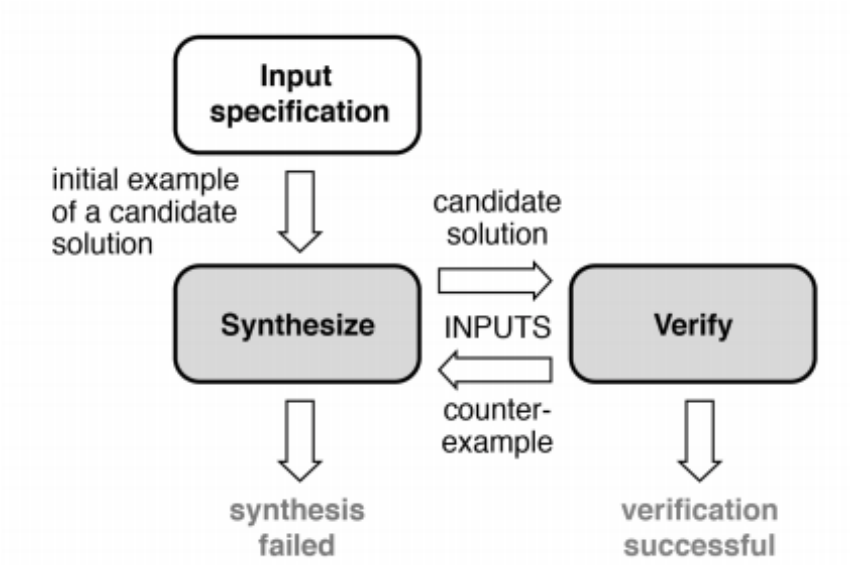


Figure 2.3: Counterexample Guided Inductive Synthesis (CEGIS).

The correctness specification σ provided to our program synthesizer is of the form $\exists\vec{F}.\forall\vec{x}.\sigma(\vec{x},\vec{F})$, where $\vec{F}$ ranges over functions, $\vec{x}$ ranges over ground terms, and σ is a quantifier-free (QF) formula typically supported by SMT solvers. The ground terms are interpreted over some finite domain $\mathcal{D}$, where $\mathcal{D}$ can be encoded using the SMT's bit-vectors part.

In Figure 2.3, the CEGIS method's SYNTHESIZE and VERIFY phases interact via a

finite set of test vector INPUTS, which is incrementally updated. Given the correctness specification σ, the SYNTHESIZE procedure tries to find an existential witness $\vec{F}$ satisfying the specification $\sigma(\vec{x}, \vec{F})$, for all $\vec{x}$ in INPUTS (as opposed to all $\vec{x} \in \mathcal{D}$). If SYNTHESIZE succeeds in finding a witness $\vec{F}$, the latter is a candidate solution (i.e., a feasible combination of equipment) for the full synthesis formula, which is passed to VERIFY in order to check whether it is a proper solution (*i.e.*, $\vec{F}$ satisfies the specification $\sigma(\vec{x}, \vec{F})$ for all $\vec{x} \in \mathcal{D}$). If this is the case, then the algorithm terminates, i.e., we have found a feasible solution; otherwise, in the CEGIS method, additional information is provided for the SYNTHESIZE phase, in the form of a new counterexample that is added to the INPUTS set and the loop iterates again.

One may notice that each iteration of the CEGIS loop adds a new input to the finite set INPUTS, which is then used for synthesis. Given that the full set of inputs $\mathcal{D}$ is finite because we use bit-vector expressions, this means that the refinement loop can only iterate over a finite number of times. However, SYNTHESIZE may conclude that no candidate solution obeying σ for the finite set INPUTS exists and our synthesis engine can then conclude that no feasible solution was found.

2.3 Solar Photovoltaic Systems

According to (ROY, 2013), a PV system is designed to supply electrical loads. These loads can be of the Alternating Current (AC) type or the Direct Current (DC) type. The electricity supply can be needed either in daytime or nighttime (most cases, in both). The most basic PV system can supply only in daytime. For the night hours or rainy days, batteries are needed, where power can be stored for later use (GULES et al., 2008).

PV systems are broadly classified into three distinct types, as described by (MOHANTY et al., 2016):

- Stand-alone systems, or off-grid systems, where the energy is generated and consumed in the same place and which does not interact with the main grid. Usually, the electricity consuming/utilizing device is part of the system, i.e., solar home systems, solar street lighting systems, solar lanterns, and solar power plants;
- Grid-connected systems, where the solar PV system is connected to the grid. The grid-connected system can be:
  - Grid-tied system, which can only feed power into the grid, with the result that this system cannot deliver power locally during blackouts and emergencies since these systems have to be completely disconnected from

the grid and have to be shut down as per national and international electrical safety standards;

  - Some grid-connected PV systems, with energy storage, can also provide power locally in an islanding mode.

- Solar PV hybrid systems. In a hybrid system, other source(s) of energy, such as wind, biomass or diesel, can work together with the solar PV system to provide the required demand. In this type of system, the main objective is to bring more reliability into the overall system at an affordable cost by adding distinct energy sources.

In the specific case of isolated communities, depending on the type of load, cost, resources availability, and the load requirements, stand-alone systems can be split into several categories, as described below. Since the goal of this thesis is to present solutions only to isolated/ off-grid applications, on-grid or hybrid configurations are not discussed here.

2.3.1 MPPT

There is a feature, called maximum power point tracking (MPPT), which is a control mechanism that maintains the PV panel operating at a voltage that corresponds to maximum power voltage, which maximizes the transfer of power while avoiding loss of PV cells (PINHO; GALDINO, 2014). This resource is found in modern PV systems and is strongly indicated due to its advantages.

2.3.2 Unregulated Stand-alone PV Systems With DC Load

Usually this type of system is for low power applications, as defined by (ROY, 2013). The PV system is directly connected to the load without any MPPT controller, as shown in Fig.2.4. At night, the system will not provide any power because of the absence of a battery.

2.3.3 Regulated Stand-alone PV Systems With DC Load

Similar to the unregulated stand-alone system with DC load, the main difference between this and the previous one is that this system requires a MPPT technique, as illustrated in Fig.2.5. Usually systems with MPPT should have a battery; otherwise, the extra power will be wasted. This is an inadequate use of PV systems, but it can be found in isolated communities in Brazil.

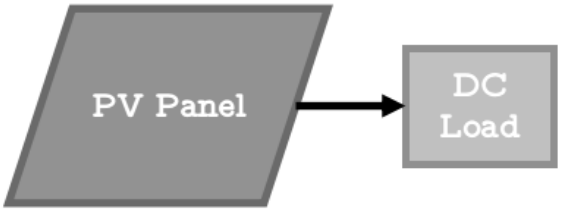


Figure 2.4: Unregulated stand-alone solar PV system with DC load. Source: (ROY, 2013).

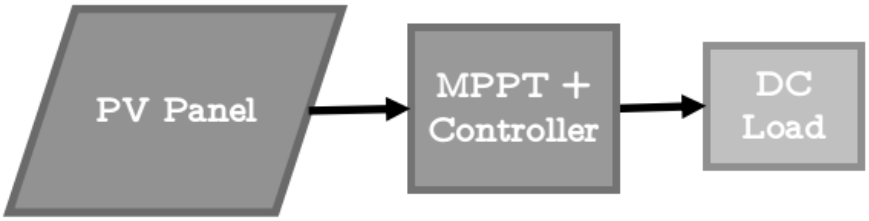


Figure 2.5: Regulated stand-alone solar PV system with DC load. Source: (ROY, 2013).

2.3.4 Regulated Stand-alone Systems With Battery and DC load

This configuration have PV array, battery, MPPT and DC load, as shown in Fig.2.6. The battery is used to store the extra power from the PV system. A charge controller is necessary for this type of system because it insures that the battery is charged with the correct voltage and current. Extra charging and deep discharging can reduce the battery life (KIM, 2006). The controller includes a DC-DC converter that is not shown in Fig.2.6, but it is inherent of a typical MPPT controller.

A PV systems that are used to feed loads with low variation of consumption can be sized to operate without the controller. This is known as a self-regulated stand-alone PV system with battery. However, the PV panel voltage must be compatible with the battery voltage. Usually, the bank of batteries is oversized in relation to the PV panel and to the load. The drawback is the operation of the batteries, usually overloaded or with excessive discharges (that can damage the batteries).

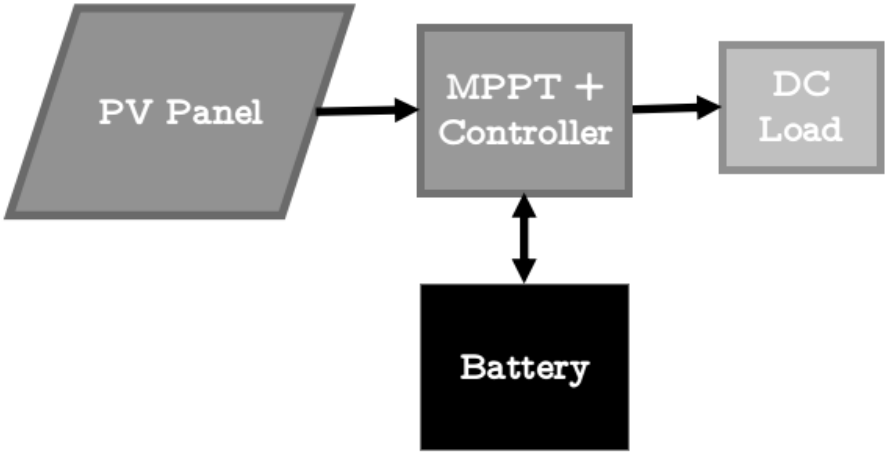


Figure 2.6: Regulated stand-alone solar PV system with battery and DC load. Source: (ROY, 2013).

2.3.5 Regulated Stand-alone Systems With Battery and AC load

This system is similar to the previous one, but the AC load draws the power from the PV system and, because of the AC load, an inverter (DC to AC converter) is required, as seen in Fig.2.7. This solution has a cost increase because it has more equipment. However, AC availability has the advantage of allowing the use of a higher number of AC appliances in homes or consumer units. This is the base configuration for this thesis since it is currently the most common system used specifically for remote rural areas of developing countries or areas where the grid extension is not feasible.

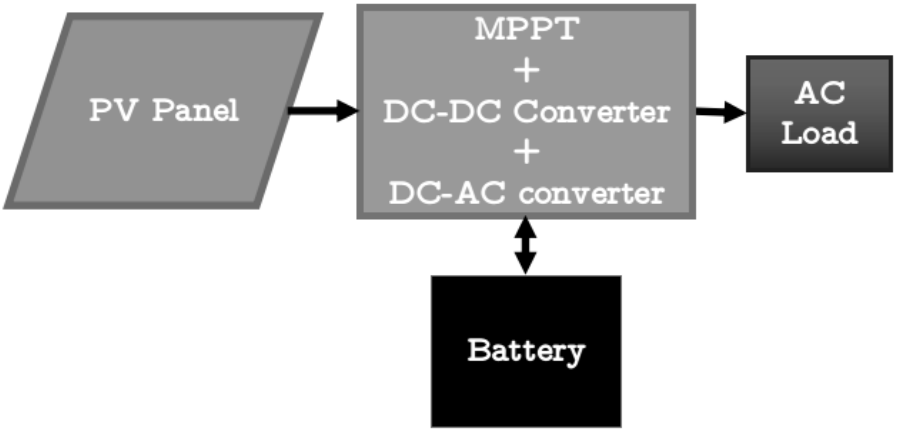


Figure 2.7: Regulated stand-alone PV system with battery, AC load. Source: (ROY, 2013).

2.3.6 Design and Validation of a Solar PV System

The design and validation of a solar PV system can be done by hand or with the aid of a software tool.

In order to address different aspects of a project system and system's design, many commercial tools are available for the solar PV market. According to (BROOKS; DUNLOP, 2013), the capabilities of these tools range from simple solar resource and energy production estimation to site survey and system design tools, to sophisticated financial analysis software (with optimization). Some tools also provide support to rebate program applications and tax incentives (specific to each country or region). In contrast, other programs and worksheets focus on the technical aspects of system sizing and design.

Manufacturers and integrators also have their proprietary software to perform inverter string sizing and various system sizing and design tools, with the drawback of just including their products among the possibilities of choice. In this study, the most widely used tools are presented:

- PVWatts
- SAM
- Homer
- RETScreen
- Hybrid2

2.3.6.1 PVWatts Calculator

According to (FREEMAN et al., 2014) and (DOBOS, 2014), this is a web application developed by the National Renewable Energy Laboratory (NREL), which estimates the electricity production of a grid-connected roof- or ground-mounted photovoltaic system based on a few inputs. According to (DOBOS, 2014), using the calculator requires information about the system's location, basic design parameters, and system economics. PVWatts calculates estimated values for the system's annual and monthly electricity production, and for the electricity's monetary value. This tool is suitable for very preliminary studies of a potential location for a photovoltaic system that uses crystalline silicon or thin-film photovoltaic modules. The production estimates that PVWatts computation does not account for many factors that are important in the design of a photovoltaic system, which makes it necessary to have the support of an energy expert. The calculator estimates the monthly and annual electricity production of a photovoltaic system using an hour-by-hour simulation over one year.

To represent the system's physical characteristics, PVWatts requires values for six inputs: system DC size, module type, array type, system losses, array tilt angle, and array azimuth angle.

2.3.6.2 SAM

SAM or System Advisor Model is a software solution produced by the U.S. Department of Energy and National Renewable Energy Laboratory. According to (BLAIR et al., 2014) and (CAMERON; BOYSON; RILEY, 2008), SAM is intended to help users to determine whether the model meets their project constraints/specifications. SAM is a performance and financial model designed to facilitate decision making for people involved in the renewable energy industry: project managers and engineers; financial and policy analysts; technology developers; and researchers. SAM makes performance predictions and energy cost estimates for grid-connected power projects based on installation and operating costs and system design parameters that the user specifies as inputs to the model. Projects can be either on the customer side of the utility meter, where they buy and sell electricity at retail rates or on the utility side of the meter, where they sell electricity at a price negotiated through a power purchase agreement. SAM is an electrical power generation model and assumes that the renewable energy system delivers power either to an electric grid or to a grid-connected building or facility to meet the electric load. It does not model thermal energy systems that meet a thermal process load. As mentioned in (BLAIR et al., 2014), SAM does not model isolated or off-grid power systems and does not model systems with electricity storage batteries.

2.3.6.3 HOMER

As defined in (HOMER, 2017), this is a set of two tools: HOMER Legacy and HOMER Pro. HOMER is an acronym for Hybrid Optimization Model for Multiple Energy Resources. HOMER Legacy is the original HOMER software version created at the National Renewable Energy Laboratory (NREL). HOMER Legacy is a free computer model that simplifies the task of evaluating design options for both off-grid and grid-connected power systems for remote, stand-alone, and distributed generation applications. HOMER's optimization and sensitivity analysis algorithms allow the user to evaluate the economic and technical feasibility of a large number of technology options. Since 2016 Homer Legacy can be found at the HOMER web site, but it is only available for students and nonprofit organizations, as defined in (HOMER, 2017), and has no support available. For a short time, only the commercial version will remain.

The commercial version (paid), known as HOMER Pro, as defined in (SWARNKAR;

GIDWANI; SHARMA, 2016), is a tool for optimizing the microgrid design in all sectors, from village power and island utilities to grid-connected campuses and military bases. HOMER Pro puts together three tools in one product: optimization, simulation, and sensitivity analysis. It provides the full rigor of historical simulation and optimization in a model that is intended to be easy to use and is adaptable to a wide variety of projects. For a village or community-scale power system, HOMER can model both the technical and economic factors involved in the project. For larger systems, HOMER can provide an overview that compares the cost and feasibility of different configurations. Historical simulation is essential for modeling variable resources, such as solar and wind power, and for combined heat and power applications, where the thermal load is variable. HOMER's sensitivity analysis helps determine the potential impact of uncertain factors such as fuel prices or wind speed on a given system.

2.3.6.4 RETScreen

As mentioned in (PRADHAN et al., 2015), RETScreen is a decision-support tool designed to help decision-makers and energy professionals to evaluate the financial viability of renewable energy, energy efficiency, and/or co-generation projects.

RETScreen models various types of renewable energy technologies (RETs), allowing for comparisons between technology options. The software can be used to evaluate benefits from both clean energy production from power generation projects and savings through energy efficiency projects, accounting for project costs, greenhouse gas emission reductions, and financial risk. The software is freely distributed (but with restrictions to save work or print), and had three different versions:

- RETScreen 4 (discontinued, requires Microsoft Excel to run);
- RETScreen Software Suite, which includes the RETScreen 4 and a Windows-based graphical software that allows project owners to verify the ongoing energy performance of their facilities (discontinued in 2013);
- And the current (2016) RETScreen Expert, which allows users to evaluate energy investments over an entire project life-cycle (including benchmarking, feasibility, and performance analysis) in a fully integrated way, and within one software platform. This version is only Windows-based and has a complete paid version via an annual subscription way.

As described by (PRADHAN et al., 2015), RETScreen performs a standard five-step analysis: setting and site conditions, energy model, cost analysis, emission analysis, financial analysis, sensitivity, and risk analysis. It is developed and maintained by the Government of Canada through the CanmetENERGY Varennes Research

Centre of Natural Resources, in collaboration with NASA; Renewable Energy and Energy Efficiency Partnership (REEEP); United Nations Environment Programme (UNEP), and the Global Environment Facility (GEF). RETScreen is available in 36 languages; it is a multi-awarded tool and includes equipment databases for components manufactured and available worldwide.

2.3.6.5 Hybrid2

The Hybrid2 software package, as described in (MILLS; AL-HALLAJ, 2004), is a user-friendly tool that executes detailed long-term performance and economic analysis on a wide variety of hybrid power systems. Hybrid2 is a probabilistic/time series computer model, using time series data for loads, wind speed, insolation, and temperature. The power system is designed or selected by the user to predict the hybrid power system performance. Variations in wind speed and in load within each time step are factored into the performance predictions. The code does not consider short-term system fluctuations caused by system dynamics or component transients. This program is not supported anymore and according to (UMASS, 2016), probably after the user performs the free download of the tool, it will not work on Windows platforms later than Windows XP, which is a limitation.

Table 2.1 summarizes the tools described in this thesis, where just Hybrid2 is mentioned, no technical support is available. Only HOMER, RETScreen, and Hybrid2 perform off-grid system or battery backup analysis; all the tools perform solar photovoltaic analysis. Only HOMER and RETScreen are complete, including economic analysis and even optimization-sensitive analysis. However, only the paid version of these software packages have all the features, and they run only on Windows-based operating systems.

2.3.6.6 Thesis Proposal x Reference Tools

Considering that the focus of this research is on off-grid solutions and supported tools, only HOMER remains for comparison. All tools need some parameters inherently from the manufacturer's catalog, so the project starts with the manufacturer's and integrator's tool to define the essential items of the project: panels, inverters, controllers, and batteries. The potential solution is then analyzed by another tool to validate or even optimize the solution. The challenge of this study, therefore, is to prove that it is possible to use automated verification to validate an off-grid PV solution.

Table 2.1: Comparative coverage of reference softwares

Characteristic	PVWatts	SAM	HOMER	RETScreen	Hybrid2
Support	X	X	X	X	
Off-grid systems			X	X	X
Hybrid systems			X	X	X
Photovoltaics	X	X	X	X	X
Batteries			X		X
Main technical (T) or economical (E)	T	T	E	E	T
Optimization			X	X	
Sensitive analysis			X	X	

2.3.7 Component Models for Stand-alone PV Systems

The primary purpose of this section is to describe the models for the elements of a stand-alone PV system: the PV generator, battery, controller, inverter, and load. The modeling of the PV system is based on modular blocks, as illustrated in Fig.2.8, from (HANSEN et al., 2001). The modular structure facilitates the modeling of the other system structures and the replacement of elements such as a DC load instead of an AC load.

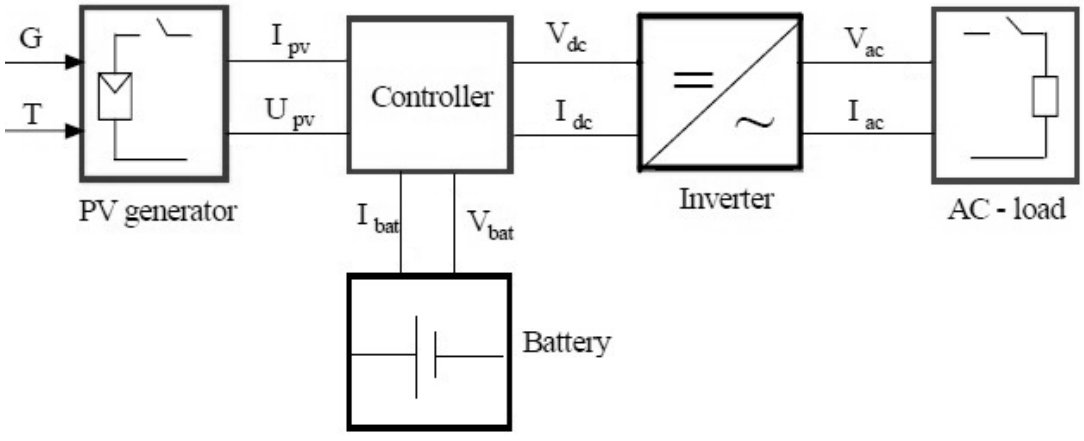


Figure 2.8: Block diagram for the stand-alone PV system. Source: (HANSEN et al., 2001).

In the literature, there are several mathematical models available for each component of stand-alone PV systems. In this section, the mathematical model for each component of PV system is presented.

2.3.7.1 PV Generator (Cell, Module, String, and Array)

A photovoltaic PV generator is the whole assembly of solar cells, connections, protective parts, and supports. In the present modeling, the focus is only on the cell/module/array.

The fundamental element of a PV System is the PV cell, also called a Solar Cell. A PV / Solar Cell is a semiconductor device that can convert solar energy into DC electricity. The semiconductor materials (usually silicon), which are specially treated to form an electric field, positive on one side (backside) and negative on the other (towards the sun). When solar energy (photons) hits the solar cell, electrons are knocked loose from the atoms in the semiconductor material, creating electron-hole pairs (LORENZO, 1994). If electrical conductors are attached to the positive and negative sides, forming an electrical circuit, the electrons are captured in the form of electric current I_{ph} (photocurrent).

In order to increase their application usage, many individual PV cells are interconnected together in a sealed, weatherproof package called Panel (or Module). For instance, a 12 V Panel will have 36 cells connected in series, and a 24 V Panel will have 72 PV cells connected in series. Besides, to achieve the desired voltage and current, modules are wired in series (strings) and parallel into what is called a PV array, as shown in Fig.2.9. The flexibility of the modular PV system allows designers to create PV systems that can meet a wide variety of electrical demands.

The PV modules are generally rated under standard test conditions (STC), which leads to the following specification by the manufacturers: solar radiation of 1000 W/m^2, cell temperature of 25^{o}C, and solar spectrum of 1.5. The parameters required for the input of the PV modules are dependent on the meteorological conditions of the area to be serviced by the photovoltaic solution. However, the climatic conditions are unpredictable due to the random nature of their occurrence (JAKHRANI et al., 2014).

These uncertainties lead to either over- or underestimation of energy yield from PV modules. An overestimation of up to 40% was reported as compared to the rated power output of PV modules (DURISCH et al., 2000).

The growing demand for photovoltaics technologies has led to research into the various aspects of its components from cell technology to the modeling, size optimization, and system performance (RAJANNA; SAINI, 2016), (BADEJANI; MASOUM; KALANTA, 2007); (YATIMI; AROUDAM, 2015), (FERRARI et al., 2016), (SALOUX; TEYSSEDOU; SORIN, 2011), (HASAN; PARIDA, 2016), (KING; BOYSON; KRATOCHVILL, 2004), (MELLIT; BENGHANEM; KALOGIROU, 2007). Model-

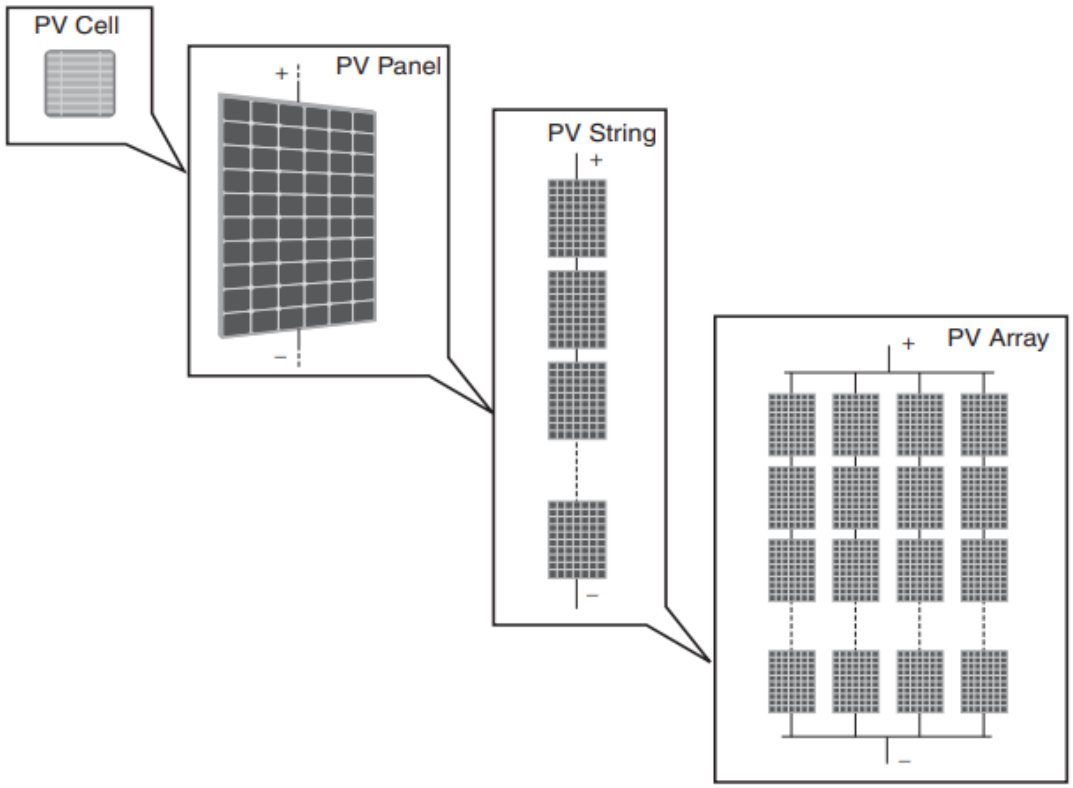


Figure 2.9: PV cell, module, string and array. Source: (SAMLEXSOLAR.COM, 2017).

ing PV modules is one of the major components responsible for the proper functioning of PV systems. However, the estimation of models is affected by various intrinsic and extrinsic factors, which ultimately influence the behavior of current and voltage. Therefore, the choice of the model is essential to estimate the performance of PV modules in different environmental conditions (JAKHRANI et al., 2014).

Modeling provides the means to understand the current, voltage, and power relationships of PV systems.

The performance of photovoltaic systems (solar cells/panels), that is, the output current/voltage curve ($I - V$ curve), is usually studied using an equivalent circuit model. This equivalent circuit consists of a current source with one or two diodes connected in parallel, and up to two resistors, one connected in parallel and the other one in series, to take into account energy losses in this model (CUBAS; PINDADO; SORRIBES-PALMER, 2017). Based on these electronic components, four basic configurations are usually used when studying PV systems, as shown in Fig. 2.10.

The 1-diode model, whose equation to relate the output current, I, to the output

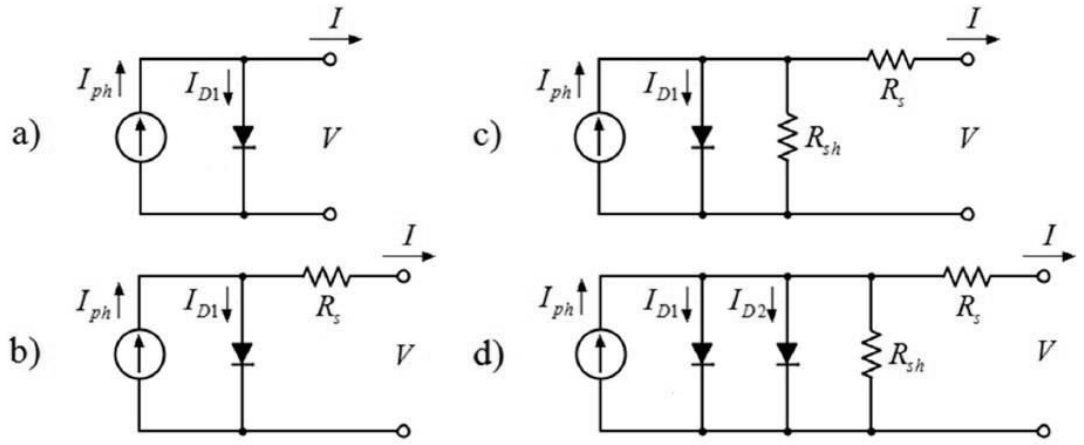


Figure 2.10: Four different equivalent circuit models: (a) 1-diode; (b) 1-diode/1-resistor; (c) 1-diode/2-resistor; (d) 2-diode/2-resistor. Source: (CUBAS; PINDADO; SORRIBES-PALMER, 2017).

voltage, V, is described in Equation 2.1.

$$I = I_{ph} - I_{D1} = I_{ph} - I_0 \left[exp\left(\frac{V}{NaV_T}\right) - 1 \right], \tag{2.1}$$

where:

- I_{ph} is the photocurrent delivered by the constant current source. I_{ph} is usually approximated to the reference short-circuit current of the PV panel (I_{sc});
- I_0 is the reverse saturation current corresponding to the diode;
- N is the number of series-connected cells in the photovoltaic system to be analyzed;
  - $N = 1$ in a single cell configuration.
- a is the ideality factor (or quality factor) that takes into account the deviation of the diodes from the Shockley diffusion theory;
  - $a = 1$ for ideal diodes and between 1 and 2 for real diodes.
- V_T is the thermal voltage ($V_T = k_B T/q$);
  - k_B is the Boltzmann constant ($1.3806503 \times 10^{-23} J/K$);
  - T is the temperature of the p-n junction (or cell temperature) expressed in Kelvin;
  - q is the absolute value of the electron's charge ($-1.60217646 \times 10^{-19} C$).

This model has only three unknown parameters (I_{ph}, I_0, and a), and it assumes that

the series resistance is $zero$ and shunt resistance is $infinite$ and, thus, both of these parameters are ignored.

The 1-diode/1-resistor model, is described in Equation 2.2.

$$I = I_{ph} - I_0 \left[exp\left(\frac{V + IR_s}{NaV_T}\right) - 1 \right], \tag{2.2}$$

where R_s is the series resistor.

In this model, there are four unknown parameters (I_{ph}, I_0, R_s, and a), and it assumes shunt resistance as $infinite$.

The 1-diode/2-resistor model, is described in Equation 2.3.

$$I = I_{ph} - I_0 \left[exp\left(\frac{V + IR_s}{NaV_T}\right) - 1 \right] - \frac{V + IR_s}{R_{sh}}, \tag{2.3}$$

where R_{sh} is the shunt resistor.

In this model, there are five unknown parameters (I_{ph}, I_0, R_s, R_{sh}, and a).

And the 2-diode/2-resistor model, is described in Equation 2.4.

$$I = I_{ph} - I_{01} \left[exp\left(\frac{V + IR_s}{Na_1V_T}\right) - 1 \right] - I_{02} \left[exp\left(\frac{V + IR_s}{Na_2V_T}\right) - 1 \right] - \frac{V + IR_s}{R_{sh}} \tag{2.4}$$

This model has six unknown parameters with two exponential terms. Briefly, both single and double diode models require the knowledge of all unknown parameters, which is usually not provided by manufacturers. Nevertheless, the current-voltage equation is a transcendental expression (JAKHRANI et al., 2014).

However, regardless of the adopted model, the parameters of the equations must be estimated to adapt the corresponding model to the real behavior of the solar cell/panel.

For this reason, researchers gradually focused on searching out the approximate methods for the calculation of unknown parameters, proceeding along three different paths. The analytical methods give exact solutions through algebraic equations, as done by (CUBAS; PINDADO; SORRIBES-PALMER, 2017) and (BRANO et al., 2010). However, due to the inherent nature and nonlinearity of PV cell or module characteristics, it is hard to discover the analytical solution of all unknown parameters, as described in (HASAN; PARIDA, 2016). Thus, numerical methods such as the Newton-Raphson method or the Levenberg-Marquardt algorithm were preferred,

as described by (MELLIT; BENGHANEM; KALOGIROU, 2007). This happens because numerical methods give approximate solutions to nonlinear problems without searching for exact solutions. However, numerical methods are time-consuming and need long term time series data, which may not be available in developing countries. Reference (JAKHRANI et al., 2014) used mixed methodology, bringing analytical and numerical steps together. (SHENAWY et al., 2015) et al. create a method to discover the unknown parameters of the PV panels through experimentation. Furthermore, (TIAN et al., 2012) used a mix of analytical and experimental methodology to establish the unknown parameters. However, samples of the PV modules are necessary to perform some tests when we use the experimental technique.

Therefore, a wide variety of models exists for estimation of the power output of PV modules (and $I-V$ or $P-V$ curves). However, this study will rely on the simplified 1-diode model, which was shown by (SALOUX; TEYSSEDOU; SORIN, 2011) that has a small error rate, between 0.03% and 4.68% for the selected PV panels tested. Besides, this mathematical modeling has the advantage of being an explicit model, which does not use iterative numerical calculation.

2.3.7.2 The Proposed PV Panel Model

With the proposed model, an explicit set of equations is derived from the ideal PV model given by Equation 2.1.

A single-diode without series and shunt resistances is considered. Equation 2.1 is used to write down expressions for currents and voltages at each key point shown in Fig. 2.11. Note that the MPPT point from the PV panel is illustrated, highlighting the maximum voltage V_m and maximum current I_m that a panel can produce.

Hence, the short-circuit current, the open-circuit voltage, the maximum power voltage and current are written as defined by (SALOUX; TEYSSEDOU; SORIN, 2011) and shown in Equations 2.5 to 2.8.

$$I_{sc} = I_{ph}|_{V=0} \tag{2.5}$$

$$V_{oc} = \frac{aNk_BT}{q} ln\left(1 + \frac{I_{sc}}{I_0}\right) \tag{2.6}$$

$$exp\left(\frac{qV_{oc}}{aNk_BT}\right) = \left(1 + \frac{qV_m}{aNk_BT}\right) exp\left(\frac{qV_m}{aNk_BT}\right) \tag{2.7}$$

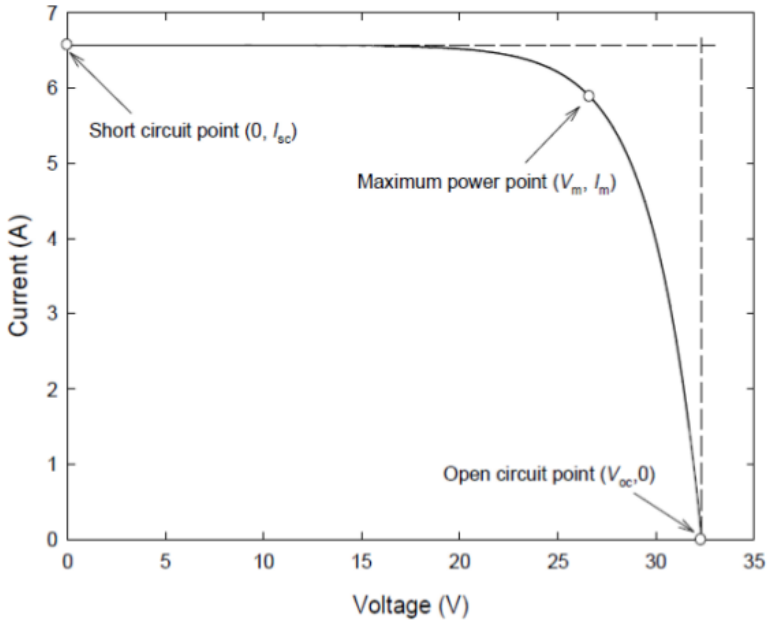


Figure 2.11: $I - V$ characteristic curve of an ideal PV cell. Source: (SALOUX; TEYSSEDOU; SORIN, 2011).

$$I_m = I_{ph} - I_0 \left[exp\left(\frac{qV_m}{aNk_BT}\right) - 1 \right] \tag{2.8}$$

Equations 2.7 and 2.8 are not explicit with regard to the key PV parameters, it therefore needs to be rewritten in a different way. A PV cell has a hybrid behavior, i.e., a current-source at the short-circuit point and a voltage-source at the open-circuit point (SALOUX; TEYSSEDOU; SORIN, 2011). These two regions are characterized by two asymptotes of the $I - V$ curve in Fig. 2.11, where the transition is a compromise between both behaviors. It is interesting to observe that the maximum power point corresponds to a trade-off condition, where the current is still high enough before it starts decreasing with the increase in the output voltage (Fig. 2.11).

Based on this, the tangent of the I-V curve can be used to evaluate the transition between current- to voltage-source controlled regions; this operation yields Equation 2.9.

$$\frac{dI}{dV} = -\frac{qI_0}{aNk_BT} exp\left(\frac{qV}{aNk_BT}\right) \tag{2.9}$$

Equation 2.9 is used to calculate the output voltage that corresponds to the maximum power operation condition of the cell, thus generating the Equation 2.10.

$$V_m = \frac{aNk_BT}{q} ln\left(-\frac{aNk_BT}{qI_0}\left(\frac{dI}{dV}\right)_{V_m}\right) \tag{2.10}$$

It is evident that Equation 2.10 requires an expression of the derivative of the current with voltage evaluated at the maximum power point. The fact that the maximum power corresponds to an extreme, the variation of the maximum output power with voltage is relatively small, i.e., a change in V_m has a relatively small effect on the maximum power of the cell (SALOUX; TEYSSEDOU; SORIN, 2011). Consequently, considering the asymptotic behavior of the $I - V$ curve in short- and open-circuit conditions, the derivative required by Equation 10 can be calculated as shown in Equation 2.11.

$$\frac{dI}{dV}|_{V_m} \cong -\frac{0 - I_{sc}}{V_{oc} - 0} = \frac{I_{sc}}{V_{oc}} \tag{2.11}$$

Replacing the Equation 2.11 into Equations 2.10 and 2.8, the voltage and the current at the maximum power point and consequently the maximum output power, are expressed by Equations 2.12, 2.13, and 2.14 ($P_m = V_m I_m$). These equations are used to calculate the key cell parameters at the maximum power point (MPPT) as a function of cell temperature and parameters from the manufacturer's data-sheet.

$$V_m = \frac{aNk_BT}{q} ln\left(\frac{aNk_BT}{qI_0}\frac{I_{sc}}{V_{oc}}\right) \tag{2.12}$$

$$I_m = I_{ph} + I_0 - \frac{aNk_BT}{q}\left(\frac{I_{sc}}{V_{oc}}\right) \tag{2.13}$$

$$P_m = \left[\frac{aNk_BT}{q} ln\left(\frac{aNk_BT}{qI_0}\frac{I_{sc}}{V_{oc}}\right)\right] \times \left[I_{ph} + I_0 - \frac{aNk_BT}{q}\left(\frac{I_{sc}}{V_{oc}}\right)\right] \tag{2.14}$$

However, the photocurrent delivered by the constant current source (I_{ph}) or even the reverse saturation current (I_0) is not given by PV manufacturers. Therefore, Equation 2.15 is used to calculate the photocurrent as a function of irradiance and temperature (VILLALVA; GAZOLI; FILHO, 2009).

$$I_{ph} = \frac{G}{G_{ref}}\left[I_{ph,ref} + \mu_I\left(T - T_{ref}\right)\right] \tag{2.15}$$

where the reference state (STC) of the cell is given by the solar irradiance $G_{ref} = 1000W/m^2$, and the temperature $T_{ref} = 298.15K (= 25^oC)$.

In Equation 2.15, μ_I is the short-circuit current temperature coefficient (A/K) and corresponds to the photocurrent obtained from a given PV cell working at (STC or standard test conditions) reference conditions (provided by PV manufacturers).

$I_{ph,ref}$ can also be approximated to the reference short-circuit current that is provided by PV manufacturers ($I_{sc,ref}$) as shown by (JAKHRANI et al., 2014).

The cell temperature (T) can be obtained from (ROSS, 1980) and is shown in Equation 2.16.

$$T = T_{air} + \frac{NOCT - 20}{800} G \tag{2.16}$$

where T_{air} is the ambient temperature, $NOCT$ is the nominal operating cell temperature (in ^{o}C) that is found on the PV manufacturer's data-sheet, and G is the solar irradiance (W/m^2) at the location of the PV system. In this thesis is not considered shading, which can impact in the cell temperature as well.

Furthermore, (VILLALVA; GAZOLI; FILHO, 2009) have proposed a relationship, which allows the saturation current (I_0) to be expressed as a function of the cell temperature. In this study, this relation is explicitly written based on cell open-circuit conditions using the short-circuit current temperature coefficient in addition to the open-circuit voltage temperature coefficient (Equation 2.17).

$$I_0 = \frac{I_{sc,ref} + \mu_I (T - T_{ref})}{exp\left[\frac{q(V_{oc,ref} + \mu_V (T - T_{ref}))}{aNk_B T}\right] - 1} \tag{2.17}$$

where $V_{oc,ref}$ is the reference open-circuit voltage, and μ_V is an open-circuit voltage temperature coefficient (V/K).

The ideality (or quality) factor of the diode a, which is usually considered as a constant (VILLALVA; GAZOLI; FILHO, 2009), is determined in the reference state. Using the maximum power point current equation (Equation 2.14) and the saturation current in the reference temperature given by Equation 2.17, the diode quality coefficient is determined by Equation 2.18.

$$a = \frac{q(V_{m,ref} - V_{oc,ref})}{Nk_B T} \frac{1}{ln\left(1 - \frac{I_{m,ref}}{I_{sc,ref}}\right)} \tag{2.18}$$

where V_{mref}, $V_{oc,ref}$, $I_{m,ref}$, and $I_{sc,ref}$ are key cell values obtained under both actual cell temperature and solar irradiance conditions, usually provided by the manufacturers.

The model is now completely determined, i.e., with all the variables defined. This model requires the actual cell temperature (or the air temperature), the actual solar

irradiance and common data provided by manufacturers.

If the PV cells are in parallel, then there is a parallel array. There will therefore be a change in the I_{ph} and I_0 and the resulting current is given by Equation 2.19, as demonstrated in (SALOUX; TEYSSEDOU; SORIN, 2011).

$$I_{array} = (N_{cellsinparallel})(I_{onecell}) \tag{2.19}$$

where $I_{onecell}$ is the current from Equation 2.13.

In addition, if the panels are in series, the current does not change but the total voltage is the sum of the voltage of each individual panel.

$$V_{array} = (N_{cellsinseries})(V_{onepanel}) \tag{2.20}$$

where $V_{onepanel}$ is the maximum voltage from Equation 2.12.

2.3.7.3 Battery Storage Model

Because of the fluctuating nature of the output delivered by the PV arrays, batteries are an essential part of a PV system. Thus, during the hours of sunshine, the PV system feeds the load directly and excess electrical energy is stored in the batteries. During the night, or during a period with low solar irradiation, energy is supplied to the load from the battery bank (MELLIT; BENGHANEM; KALOGIROU, 2007).

Several models have been presented in the literature. However, regardless of the model, the following parameters are usually required(MELLIT; BENGHANEM; KALOGIROU, 2007):

- Nominal capacity (q_m), is the number of Ampere-hours (Ah) that can maximally be extracted from the battery, under predetermined discharge conditions.
- State of charge (SOC), is the ratio between the present capacity and the nominal capacity, i.e., $SOC = q/q_{max}$. Obviously $0 < SOC < 1$. If $SOC = 1$, then the battery is totally charged; and if $SOC = 0$, the battery is fully discharged.
- Charge (or discharge) regime. This parameter reflects the relationship between the nominal capacity of a battery and the current at which it is charged (or discharged). It is expressed in hours.
- Efficiency (η_b), is the ratio of the charge extracted during discharge divided by the amount of the charge needed to restore the initial state of charging or

discharging current.

- Lifetime, is the number of charge/discharge cycles the battery can sustain before losing 20% of its nominal capacity.

The merit of a stand-alone PV system is evaluated in terms of the reliability of the electricity supply to the load and in terms of the long-term efficiency of the components. Battery efficiency was described in this section, and the reliability is quantified by the concept of loss of load probability (LLP). LLP is defined as the ratio between the Ampere-hour deficit and the Ampere-hour demand, both with respect to the load, over a long period of time (COPETTI; LORENZO; CHENLO, 1993).

In general, the battery models view the battery as a voltage source E in series with an internal resistance R_0, as shown in Fig. 2.12.

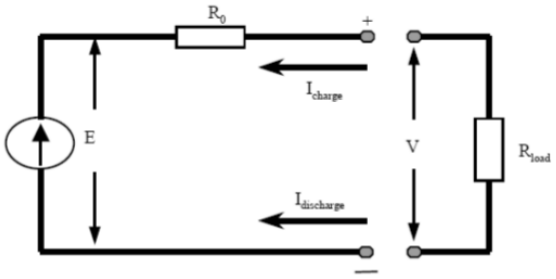


Figure 2.12: Schematic diagram of the battery. Source: (HANSEN et al., 2001).

The battery model, which describes the relationship between the voltage, the current and the state of charge, can be found in (COPETTI; LORENZO; CHENLO, 1993), (MANWELL; MCGOWAN, 1993), and (MANWELL; MCGOWAN, 1994).

Manwell and McGowen's Kinetic Battery Model (KiBaM) (MANWELL; MCGOWAN, 1993) was developed at the University of Massachusetts to predict the performance of the battery, based on manufacturer's data. However, it uses some data extracted from batteries tested in laboratory. It is therefore not suitable for this study.

Most of the models created were used to simulate and optimize PV storage systems based on lead-acid batteries, the most commonly used batteries in developing countries, owing to their relative low cost and wide availability. Batteries with modern technologies, as nickel–cadmium battery (NiCd), nickel metal hydrate (NiMH), lithium-ion (Li-ion), among others, although they have advantages (greater efficiency, longer life-time, greater depth of discharge), they are generally not economically viable in most PV systems. Table 2.2 shows the main features of some rechargeable batteries that are available at the market nowadays.

Here, the model adopted is the based on (COPETTI; LORENZO; CHENLO, 1993),

Table 2.2: Technical data for some rechargeable batteries. Adapted from (PINHO; GALDINO, 2014)

Technology	Efficiency %	Life Time years	Number of Cycles	Operational Temperature
Lead-acid (Pb-acid)	80-90	3-20	250-500	-15 to +50
Nickel–cadmium (NiCd)	60-70	3-25	300-700	-45 to +50
Nickel metal hydrate (NiMH)	80-90	2-5	300-600	-20 to +60
Lithium-ion (Li-ion)	90-95	-	500-1000	-20 to +60

who uses manufacturer's data and allows finding relations among voltage, current, state of charge and temperature.

The discharge voltage equation is shown in Equation 2.21. The first term represents the voltage variation with the state of charge (SOC), i.e., the electrolyte concentration, and the second is the variation due to internal resistance variation.

$$V_d = [2.085 - 0.12(1 - SOC)] - \frac{I}{C_{10}}\left(\frac{4}{1 + I^{1.3}} + \frac{0.27}{SOC^{1.5}} + 0.02\right)(1 - 0.007\Delta T) \tag{2.21}$$

where C_{10} means 10h of rated capacity, which is standard on the manufacturer's datasheet, ΔT is temperature variation ($\Delta T = T - T_{ref}$, $T_{ref} = 25^oC$, SOC indicates how much electric charge is stored in the cell at any given time, defined by Equation 2.22.

$$SOC = \left(1 - \frac{Q}{C}\right) \tag{2.22}$$

where Q is the charge delivered at the time of interest ($Q = It$), and C is the battery capacity.

The ratio between Q and C represents the depth of discharge (DOD) or the fraction of discharge, i.e., $DOD = 1 - SOC$. However it is worth to separate the DOD into two different definitions when the battery autonomy is bigger than one day (24 hours). Therefore, here in this thesis, the definition given here is for maximum DOD. When we deal with daily DOD we will call it of DOD_{day}, and obviously the sum of every DOD_{day} can not exceed the maximum DOD.

The efficiency of the battery discharge is assumed to be 100%, according to (COPETTI; LORENZO; CHENLO, 1993); however, the total amount of useful charge available during discharge is limited by the current rate and temperature given by Equation

2.23. This equation, known as capacity, is normalized with respect to discharge current corresponding to C_{10} rated capacity (I_{10}).

$$\frac{C}{C_{10}} = \frac{1.67}{1 + 0.67\left(\frac{I}{I_{10}}\right)^{0.9}}(1 + 0.005\Delta T) \tag{2.23}$$

Note that when the discharge current tends to zero at 25^{o}C, the maximum capacity that can be removed is about 67% over the C_{10} capacity.

For the charging process, however, the parameters are presented in Equation 2.24.

$$V_c = [2{+}0.16SOC]{+}\frac{I}{C_{10}}\left(\frac{6}{1 + I^{0.86}} + \frac{0.48}{(1 - SOC)^{1.2}} + 0.036\right)(1{-}0.025\Delta T) \tag{2.24}$$

SOC can be calculated easily at any point during the discharge period; however, during recharge it is much more difficult (COPETTI; LORENZO; CHENLO, 1993).

Generally, the efficient region is where SOC is below 0.7 and V_c is less than $2.3V$ per cell. The efficiency drops to zero at full charge and the function that represents the charge efficiency (η_c) variation with state of charge and current rate is given in Equation 2.25.

$$\eta_c = 1 - exp\left[\frac{20.73}{\frac{I}{I_{10}} + 0.55}(SOC - 1)\right] \tag{2.25}$$

(COPETTI; LORENZO; CHENLO, 1993) show that, during overcharge, gassing occurs and tests have demonstrated that the final charge voltage (V_{ec}) increases with the current intensity and with the decrease in temperature (Equation 2.26). A function was created for the gassing voltage also, as shown in Equation 2.27. In addition, the overcharge phenomenon can be represented by Equation 2.28.

$$V_{ec} = \left[2.45 + 2.011ln\left(1 + \frac{I}{C_{10}}\right)\right](1 - 0.002\Delta T) \tag{2.26}$$

$$V_g = \left[2.24 + 1.97ln\left(1 + \frac{I}{C_{10}}\right)\right](1 - 0.002\Delta T) \tag{2.27}$$

$$V_c = V_g + (V_{ec} - V_g)\left[1 - exp\left(\frac{Ah_{restored} - 0.95C}{I\tau}\right)\right] \tag{2.28}$$

where $Ah_{restored}$ represents the Ampere-hour stored in the battery with regard to the battery capacity (C) during this hour.

The function assumes that 95% of the capacity has already been restored at the start of overcharge.

The time constant of the phenomenon (τ) is inversely proportional to the charge intensity and can be written as Equation 2.29.

$$\tau = \frac{17.3}{1 + 852\left(\frac{I}{C_{10}}\right)^{1.67}} \tag{2.29}$$

Equation 2.24 can therefore be used to model the voltage (V_c) evolution of the battery, at the start of gassing ($V_c \leq V_g$). During overcharging ($V_c > V_g$), Equation 2.28 can be used until a constant final voltage (V_{ec}) is reached.

The battery's storage capacity can be calculated using Equation 2.30, as defined in (WENHAM; GREEN; WATT, 1994).

$$Storage\,capacity = \frac{N_C E_{load}}{DOD\eta_b} \tag{2.30}$$

where DOD is the maximum possible depth of battery discharge, E_{load} is the average energy consumed by the load corrected according PV equipment efficiency, N_C is the largest number of continuous cloudy days of the area, and η_b is the efficiency of the battery.

As an example of this formula's application, as shown in (ABDULATEEF, 2014), considering that an off-grid PV system is intended to supply $1.5kW/48V$ for 24 hours ($= 36kWh$); The largest number of continuous cloudy days in the selected site is about 1 day; For a maximum depth of battery discharge DOD of 0.8 and battery efficiency at 80%.

In this thesis, the adopted efficiency of the battery will be considered constant and equal to 86% as expected to lead-acid batteries (PINHO; GALDINO, 2014).

Using Equation 2.30, the storage capacity then becomes $56.3kWh$. Since the selected DC bus voltage is $48V$, then the required Ampere-hours of the battery is $1173Ah$ ($56.3kWh/48$). If a single battery is 12 V and 350 Ah, then four batteries are connected in series ($4 \times 350Ah = 1400Ah$) and a total of 16 batteries is defined (an array of 4 to reach the 48 V of the DC bus in four arrays to feed the Ah demanded).

In this study, a simplified model for battery charging (Equation 2.31) and discharging

(Equation 2.32) was considered, even recognizing that the process is not linear and is temperature-dependent. The equations are used to update the SOC of the batteries, and have the number of hours (Num_h) as a variable. There is a factor (1.15) which is present in the charging equation, and is necessary to express that during the charging process it is usual to reach 115% of the battery capacity.

$$SOC_{charge} = SOC_{previous} + \frac{100 * Pm * Num_h}{V_{system} * capacity * N_{BP} * 1.15} \tag{2.31}$$

$$SOC_{discharge} = SOC_{previous} - \frac{100 * I_{drained} * Num_h}{capacity} \tag{2.32}$$

It is worth mentioning that, specifically for PV systems used in Brazil, a Regulation (RN 493/2012), issued by the Brazilian Electricity Regulatory Agency (ANEEL) recommends a minimum of 48 hours of battery autonomy for stand-alone solar PV systems, among others related definitions.

2.3.7.4 Controller Model

The controller is named differently by different authors: controller (HANSEN et al., 2001), charge controller (MAHANTA; DEBNATH; RAHMAN, 2014) and (CHAUHAN; SAINI, 2015), regulator (MELLIT; BENGHANEM; KALOGIROU, 2007), DC-DC converter with MPPT and switch (DHANOWA; GARG, 2015), (YATIMI; AROUDAM, 2015), (ABDULATEEF, 2014), (ROY, 2013). However, in this study, in order to simplify, the term used is controller.

The controller is a set of items (DC-DC converter, a MPPT algorithm and switches) and can be defined as the responsible for managing the energy flow to the PV system, batteries and loads by collecting information on the battery voltage and knowing the maximum and minimum values acceptable for the battery voltage. Controllers aim to protect the battery (or batteries) against the excessive charge and discharge, improving its lifetime.

Currently, controllers with MPPT algorithms are the most widely used nowadays, and they maintain the PV operating at the stage of maximum power.

As defined by (HANSEN et al., 2001) and (MELLIT; BENGHANEM; KALOGIROU, 2007), all power systems must include a control strategy, which describes the interactions between its components. The use of a battery as a storage form thus implies the presence of a charge controller.

In general, there are two main operating modes for the controller: normal operating condition, when the battery voltage fluctuates between maximum and minimum

voltages; and overcharge or over-discharge conditions, which occur when the battery voltage reaches some critical values.

(MELLIT; BENGHANEM; KALOGIROU, 2007) established that the controller allows the management of energy between the load and the battery. The input signals for the regulator model are the battery current, the PV generator's voltage, the PV generator's current, and battery voltage. The outputs are battery current and used current.

According to (HANSEN et al., 2001), in order to protect the battery against an excessive charge, the PV arrays are disconnected from the system, whenever the terminal voltage increases above a certain threshold V_{max_off} and whenever the current required by the load is less than the current delivered by the PV arrays. PV arrays are connected again when the terminal voltage decreases below a certain value V_{max_on}. This can be done by using a switch with a hysteresis cycle, as illustrated in Fig. 2.13 in a on-off charge controller model.

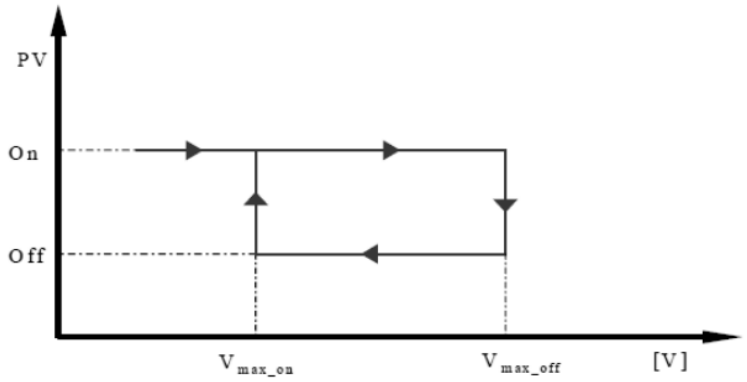


Figure 2.13: Operating principle of an overcharge protector. Source: (HANSEN et al., 2001).

To protect the battery against excessive discharge, the load is disconnected whenever the terminal voltage falls below a certain threshold i and when the current required by the load is larger than the current delivered by the PV arrays (HANSEN et al., 2001). The load is reconnected to the system, when the terminal voltage is above a certain value i, using a switch with a hysteresis cycle, as shown in Fig. 2.14.

According to (LORENZO, 1994), the switches may either be electromechanical (relay, contactors, etc.) or solid state (bipolar transistors, MOSFET's, etc.).

The steps in the modeling of the controller process are summarized in Table 2.3.

As to the DC-DC converter, the most basic idea is that the power is converted while altering the current and voltage.

As shown in (ABDULATEEF, 2014), the DC-DC converter is used to increase the

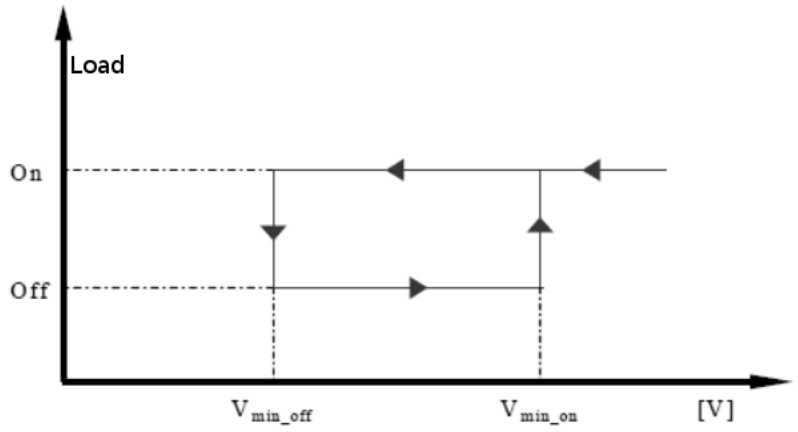


Figure 2.14: Operating principle of a discharge protector. Source: (HANSEN et al., 2001).

Table 2.3: Summary of the controller process. Source: adapted from (HANSEN et al., 2001)

Step	Constraint	Command
(1)	If $V > V_{max\_off}$ and $I_{load} < I_{pv}$	Disconnect PV array from the system
(2)	If command (1) is done and $V < V_{max\_on}$	Reconnect PV array to the system
(3)	If $V < V_{min\_off}$ and $I_{load} > I_{pv}$	Disconnect the load from the system
(4)	If command (3) is done and $V > V_{min\_on}$	Reconnect the load to the system

efficiency of the PV system by matching the voltage generated by the PV array to the voltage required by the load. The output power (P_{out}) of the DC-DC converter is given by Equation 2.33.

$$P_{in}\eta_c = P_{out} \tag{2.33}$$

Assuming that the efficiency of the controller (η_c) is data provided by the manufacturer (ideally constant in this thesis), from Equation 2.33 it is possible to reach Equation 2.34.

$$V_{in}I_{in}\eta_c = V_{out}I_{out} \tag{2.34}$$

where V_{in} is the voltage across the PV array, I_{in} is the current output of the PV array, V_{out} is the DC bus voltage, and I_{out} is the output current from the converter, when all the other values are known.

It is worth mentioning that, depending on the PV panel generation and on the batteries charge, V_{out} can be the voltage that charges the battery V_b (which is greater than the nominal voltage of the batteries), the voltage that is feed by the batteries (the same V_b but with levels that can be lower), or even the voltage that comes from the PV panels and depends on the MPPT system employed by the controller.

The output voltage is related to the input voltage as a function of the duty cycle of the switch ((ABDULATEEF, 2014)).

A DC-DC converter used in these applications can either be step-up (Boost), step-down (Buck), or both increase and decrease (Buck-Boost) the voltage, as defined by (MAHANTA; DEBNATH; RAHMAN, 2014). In addition, there is the Cúk converter, which is a Buck-Boost converter with an inverting topology (CATHERINE; BHASKAR, 2013).

For the Cúk converter, the relationship is expressed by Equation 2.35 as shown in (ABDULATEEF, 2014).

$$\frac{V_{out}}{V_{in}} = \frac{D}{D-1} \tag{2.35}$$

where D is the duty cycle or ratio of the circuit converter, i.e., it is defined as the ratio of the on time of the switch to the total switching period.

The DC/DC converter tries always to operate in the MPPT to maximize the PV array efficiency and consequently increase the efficiency of the PV system, as defined in (YATIMI; AROUDAM, 2015).

Various types of MPPT schemes are proposed by researchers, namely open circuit, short circuit, perturb and observe (P& O)/hill climbing, incremental conductance, and so forth, as shown in (HAQUE, 2014).

As the MPPT definition and the equations to get the maximum power from the PV panels were described at the end of the PV panel modeling, what is important here is that Equation 2.34 defines the relationship between the input signal, the efficiency of the controller and the output power.

2.3.7.5 The Inverter Model

As shown by (MELLIT; BENGHANEM; KALOGIROU, 2007), PV arrays produce DC voltage and therefore when the PV system contains an AC load, DC/AC conversion is required. An inverter is a converter, where the power flows from the DC to the AC side, having DC voltage as input and producing AC voltage as output. The role of the inverter is to keep the voltage constant on the AC side, i.e., at the

rated voltage (127 V or 220 V, for example), and to convert input power P_{in} into output power P_{out} with the best possible efficiency (HANSEN et al., 2001).

The inverter is characterized by a power dependent efficiency η_i given by (HANSEN et al., 2001) as shown by Equation in 2.36.

$$P_{out} = \eta_i P_{in} = \eta_i V_{DC} I_{DC} \eta_i = \frac{P_{out}}{P_{in}} = \frac{V_{AC} I_{AC} cos\varphi}{V_{DC} I_{DC}} \tag{2.36}$$

where I_{DC} is the current required by the inverter from the DC source in order to be able to keep the rated voltage on the AC side, V_{DC} is the input voltage to the inverter delivered by the DC source (PV panel or battery).

Therefore, with this equation it is possible to simulate the output power of the inverter, based on information from the inverter's data-sheet and from the DC module or the PV panel that feeds the inverter (which is obtained by this study model).

2.3.8 Availability of Stand-alone PV Systems

Each stand-alone PV system, like any other power system, has a specific level of availability. This reliability level impacts various issues: system performance, production, feasibility, and investment.

The availability of a stand-alone PV system can be defined as the percentage of time during which a power system is capable of meeting the load requirements (KHATIB; ELMENREICH, 2014). The number of hours that the system is available, divided by 8,760 h, gives the annual system availability. The definition of system availability depends on how critical the load application is. For critical loads, 99% is considered acceptable, while in an ordinary house electrical load, 95% is considered acceptable.

As an example, a system with 95% availability is expected to meet the load requirement of 8,322 h during an average year for the entire useful life of the system. The seasonal availability of 99% means that the system can operate the load for 8,672 h of the 8,760 h.

With that in mind, it is essential to mention that even if formal verification or a simulation shows that a PV system fails, this does not mean that the sizing is wrong. It is essential to evaluate how critical the load application is and the evaluation period. However, this analysis can be useful, for example, to improve sized battery autonomy.

2.3.9 Sizing Stand-alone Solar Photovoltaic Systems

The mathematical model presented in Section 2.3.7 is vital for project validation. However, if an approach is used to perform the sizing of stand-alone PV systems, then the tool proposed in this thesis can perform system validation (the intended behavior), and obtain optimal project sizing.

The sizing check stage can ensure that the system meets the standard project steps related to the critical period method for solar energy system sizing (PINHO; GALDINO, 2014) and adopting an MPPT (Maximum Power Point Tracking) charge controller, which is the most common in use. Firstly, we need to correct the daily energy consumption estimated for the load ($E_{consumption}$), which is carried out by Eq. (2.37), where the efficiency of batteries (η_b), controller (η_c), and inverter (η_i) are considered (PINHO; GALDINO, 2014) as follows

$$E_{corrected} = \frac{E_{consumption}}{\eta_b \times \eta_c \times \eta_i}. \tag{2.37}$$

We must estimate the total power that will be demanded from the PV panels, as defined by Eq. (2.38).

$$P_{min,panels} = \frac{1.25 \times E_{corrected}}{Insolation}, \tag{2.38}$$

where $Insolation$, also called solar irradiation or solar exposure, is expressed in terms of kWh/m^2 per day and depends on the site where the PV system will be deployed. A factor of 20% for losses is considered, because $1.25 = 1 \,/\, (1 - 0.2)$.

On the one hand, the sizing must meet this requirement of minimum power supplied from the PV panels $P_{min,panels}$. On the other hand, the arrangement, if in series or parallel connections, it will depend on the charge controller specification of current I_c and voltage V_c, as shown in Eq. (2.39) and (2.40).

$$I_c \geq I_{total,PVpanels}, \tag{2.39}$$

$$V_c \geq V_{total,PVpanels}, \tag{2.40}$$

Related to the batteries, the energy E_b to be demanded by the PV system, in order

to meet the load requirements, is defined by Eq. (2.41).

$$E_b = \frac{Autonomy \times E_{corrected}}{DOD}, \tag{2.41}$$

where *Autonomy* is the number of days expected for the PV system to work even when rain or clouds avoid the recharging of batteries.

In order to define the DOD_{day} we use Eq. (2.42). Moreover, the minimum current from DC bus is defined by Eq. (2.43). This equation is important to define the battery arrangement of the system (series and parallel connections).

$$DOD_{day} = \frac{E_{corrected} \times 100}{E_b}, \tag{2.42}$$

$$I_{min,DCbus} = \frac{E_b}{V_{system}}, \tag{2.43}$$

where V_{system} is the DC voltage of the bus. As for batteries, we must first define the total capacity of the battery bank, which can be described as

$$C_{bank} = \frac{Eq.\ (2.30)}{V_{system}}, \tag{2.44}$$

Equation 2.45 then performs the final sizing check, considering the number of batteries in series (N_{BS}) and the number of batteries in parallel (N_{BP}) adopted in the project.

$$(N_{BS} \times N_{BP}) \geq N_B total \tag{2.45}$$

The charge controller must initially meet the voltage requirement of the PV system, as described by Eq. (2.46).

$$V_c = V_{system}. \tag{2.46}$$

The short circuit reference information from the manufacturer's solar panel must be corrected to the cell temperature because field temperature is higher than nominal or laboratory temperature, and the PV system is temperature dependent, as shown by Equation (2.47).

$$I_{sc,amb} = \frac{G}{G_{ref}} \left[I_{sc,ref} + \mu_I \times (T - 25) \right]. \tag{2.47}$$

The controller must meet the maximum current from the PV array given by Eqs. (2.48) and (2.49).

$$I_{c,min} = I_{sc,amb} \times N_{PP}, \tag{2.48}$$

$$I_c \geq I_{c,min}. \tag{2.49}$$

The number of controllers required for the off-grid PV system, as defined by (YATIMI; AROUDAM, 2015), is calculated using Equation 2.50. Besides, the final sizing check is performed by Equation 2.51, which validates the number of controllers adopted.

$$number_{controllers} = \frac{Total\,max\,power\,of\,PV}{Controller\,max\,power} = \frac{P_{m,ref} \times N_{TP}}{V_{system} \times I_{controller,max}} \tag{2.50}$$

$$N_{controller} \geq number_{controllers} \tag{2.51}$$

The inverter sizing check is performed through three equations. Eq. (2.52) ensures that the input voltage of the controller meets the system voltage. Eq. (2.53) ensures that the output voltage of the controller meets the AC voltage of the load. Finally, Eq. (2.54) ensures that the controller can support the total demand of the load ($Demand$) and the surge power (P_{surge}), where $V_{in}DC$ is the nominal input voltage, and $V_{out}AC$ is the nominal output voltage of the inverter; $MAX_{AC,ref}$ is the peak power that the inverter can support.

$$V_{in}DC = V_{system}. \tag{2.52}$$

$$V_{out}AC = V_{AC}. \tag{2.53}$$

$$\left[(Demand \leq P_{AC,ref})\,and\,(P_{surge} \leq MAX_{AC,ref})\right]. \tag{2.54}$$

Some inverter models allow parallel operation of more than one unit, besides the integration in order to create bi-phase and three-phase circuits. It is advisable to use pure sine wave inverters, especially for electronic loads sensitive to harmonic distortion waves.

Besides that, it is crucial to verify the compatibility between the charge controller and inverter because some models are not compatible with equipment from other manufacturers. Furthermore, it is vital to select an inverter power that is lower than (or equal) the charge controller power, because the demand from the electric load causes the inverter to transfer this demand from the DC side. Then the controller can be overcharged during this operation and to burn.

2.3.10 PV Systems Optimization Criteria

In order to select an optimal combination to meet sizing constraints, it is necessary to evaluate power reliability and analyze system cost for the underlying system. A PV system is best produced when there is an ideal compromise between these two objectives.

During the PV system design, one of the essential aspects of ensuring power system reliability is to analyze power supply availability (ALSADI S.; KHATIB, 2018). The reason is that solar energy production is intermittent and, therefore, the energy generated will usually not match the load demand. Reliable power is a generation system that has sufficient power to feed load demand in a period.

There are different methods of expressing system reliability, where the most popular ones are the loss of load probability (LOLP) and the loss of power supply probability (LPSP) (ALSADI S.; KHATIB, 2018). In both methods, if the probability is zero, then the load will always be fulfilled; otherwise (i.e., probability of one), the load will never be fulfilled.

LOLP is the probability for the case when load demand exceeds the power generation of the PV system. On the one hand, we claim that we have a reliable PV system when it can generate sufficient power to fulfill the demanded load within a period. On the other hand, LPSP is defined as the probability of the system generating insufficient power to satisfy the load demand. The main approaches to LPSP demand simulation or probabilistic treatment of time series data to predict dynamic changing in system performance. However, data is not always available, and dynamic analysis is complex; this is a drawback of both LOLP and LPSP (ALSADI S.; KHATIB, 2018).

Various methods of economic analysis are available. Their main objective is to determine whether the project is an acceptable investment. The usual way is to perform economic analysis after reliability analysis to propose a system with high reliability at the lowest cost (ALSADI S.; KHATIB, 2018). The most commonly used methods include: Net Present Cost (NPC) (PARK; KUMAR; KUMAR, 2004), the Levelized Cost of Energy (LCOE) (ZHOU et al., 2010), or the Life Cycle Cost

(LCC) (APPLASAMY, 2011).

The NPC is the present value of all the costs over the project lifetime, minus the present value of all the revenues that it earns over the project lifetime. The net present worth is found by discounting all cash inflows and outflows, including the cost of installation, replacement, and maintenance of the PV system, at an internal rate of return (IRR) (PARK; KUMAR; KUMAR, 2004). IRR is used to evaluate the attractiveness of a project or investment.

LCOE is defined as the average cost per kWh of useful electrical energy produced by the PV system when lifetime, investment cost, replacement, operation and maintenance, and capital cost are considered (KAMEL; DAHL, 2005). The LCOE method is useful in comparing different generations of technology with different operating characteristics (ZHOU et al., 2010).

LCC is the estimation of the current value of the sum of the installation cost, the operation and maintenance of a PV system for a period of time (APPLASAMY, 2011). Eq. (2.55) is used to calculate LCC of a PV system,

$$LCC = C_{PV} + C_{bat} + C_{charger} + C_{inv} + C_{installation} + C_{batrep} + C_{PWO\&M}, \quad (2.55)$$

where C_{PV} is the PV array cost, C_{bat} is the initial cost of batteries, $C_{charger}$ is the cost of the charger, C_{inv} is the inverter cost, $C_{installation}$ is the installation cost, C_{batrep} is battery replacement cost at current prices, and $C_{PWO\&M}$ is operation and maintenance costs at current prices.

2.3.11 Stand-alone PV System Optimization Technique

In order to recommend an optimal configuration for a PV system, the designer has to evaluate the design based on optimization variables. As the number of optimization variables increases, the computational effort will increase accordingly. Hence, to obtain the best PV system design as well as a simplified sizing process, three main techniques have been presented in the literature for system sizing calculation, namely intuitive, numerical, and analytical methods (ZHOU et al., 2010).

The intuitive method is simple, easily implemented, and can be used to give a rough suggestion for the preliminary design. The sizing rules are based on the designer's experience, using the lowest performance data for a time period or by directly using average value (daily, monthly, or annual) of solar irradiance. This method does not consider the battery's state of charge or even the random nature of solar irradiation and meteorological conditions (ALSADI S.; KHATIB, 2018).

In the numerical method, the design is simulated for each time step within a period. The battery state of charge is calculated and investigated. Although very accurate, it is also complex, demanding more calculation time (PARK; KUMAR; KUMAR, 2004).

Analytical methods are used to obtain a close relation or correlation in the form of an equation between capacities and reliabilities. The sizing task becomes much more straightforward than in the numerical technique. However, the relation cannot be applied to different sites since it is specific to one place of deployment of the PV system, thereby demanding adaptation if another site is analyzed.

2.4 Summary

In this chapter, all the background and theoretical base needed to understand the concepts, techniques, tools, and criteria used during the following Sections were presented.

A historical explanation was presented of the origin of formal methods, culminating with modern model checkers. The evolution of the computers was highlighted as essential to the automation of formal verification. If we were stuck in time, formal verification would probably still be being done by hand.

The importance of mathematical rigor to the models and representation of systems was emphasized, and that the automated verification method can be applied to any system (from the most simple to the most complex).

State-of-the-art automated verification tools were described with respect to their characteristics and techniques. Simulation tools were also described and the reason why HOMER Pro was chosen for the comparison was explained.

System validation methods were explained, comparing test, simulation and automated verification techniques. It was possible to demonstrate that only automated verification can prove the absence of system flaws.

The method of critical period solar energy sizing was presented as the one chosen for sizing solar PV systems. Following that, the criteria and techniques for optimal sizing of PV systems were also described.

Chapter 3

Automated Formal Verification of Stand-Alone Solar PV Systems

In this chapter, we detail the methodology adopted to perform formal verification of stand-alone solar PV systems using formal methods, more specifically model checking. Diagrams, flowcharts, and algorithms support and explain the solutions. The experimentation, case studies, and results are also presented. In addition, the chapter contains a comparison with a specialized simulation tool, and also the use of different verifiers that evaluate performance by means of automated verification. Tables, commented reports, and graphic outputs are presented to aid understanding.

Additionally, we present all assumptions and premises adopted, all of which support the results/conclusions, with direct impact on them. Usually, a premise is an unquestionable fact, however assumptions can be questioned. Unlike premises, assumptions are not explicit and need to be deciphered. With that in mind, we perform a detailed explanation of all assumptions over the course of the chapter.

It is important to emphasize that the theoretical basis of the subject presented in this chapter was discussed in chapter 2, Background. In addition, knowledge of the literature is essential to aid understanding.

3.1 Methodology for Automated Verification of Solar PV Systems

Fig. 3.1 illustrates how a stand-alone solar PV project can be validated, starting with the traditional techniques - manual, simulation, testing, only then including the proposed automatic verification that is detailed in this Section.

Note that, on the one hand, although the input information is the same for all

the techniques, automated validation is different in that it is possible to define the bound k to restrict the space-design search. Among the possible inputs are: weather data at the location (temperature, solar irradiance, and insolation); system sizing information regarding specifications and configuration of PV panels, charge controller, inverter, batteries, DC bus voltage; and requirements (battery autonomy, electric load demand, electric peak power demand, energy consumption, load curve, and AC voltage).

In contrast, the outputs are not equal: design-space coverage and the information presented as result. Design-space coverage was shown in chapter 2 (Background) to be the most complete when performed by automated verification, event using the bound k to restrict the search, it not being necessary to unbound the system completely in order to discover a design flaw. Moreover, testing and simulation depend on the test vector used as input to evaluate the system.

With respect to the results presented, a project validated manually is just a piece of paper; simulation software produces success - fail information and can additionally present the optimized design if the system evaluated has some flaw, with graphics and reports; testing (whether laboratory or field) uses measurement equipment and/or data from monitoring systems. Automated verification, as proposed in this thesis, presents success-fail information also, however the output is not graphic (as will be shown), is summarized in reports with details about the variables and states of the project that cause a project flaw (in this case, where a design flaw is detected).

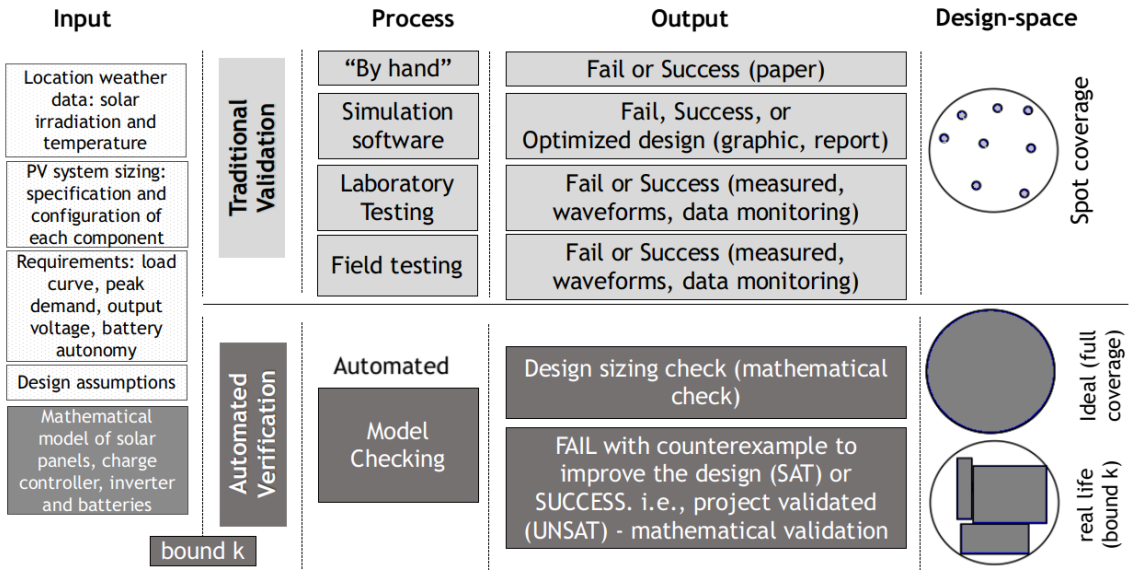


Figure 3.1: Project validation methods compared

Starting at this point, it will be shown how the proposed automated verification of stand-alone solar PV systems functions.

The process begins with the conversion of a real PV system into a model. Fig. 2.2 showed how a general system is converted to a model in order to be verified by model checking. Fig. 3.2 is an adaptation of the general diagram, replacing the original system by a PV system and detailing the inputs, outputs, and requirements.

It is important to bear in mind that the model checking process is the same whatever the system that is being validated; what is important is to choose an accurate model, define the requirements/constraints, and get correct information from the system components, in order to achieve sound and effective validation.

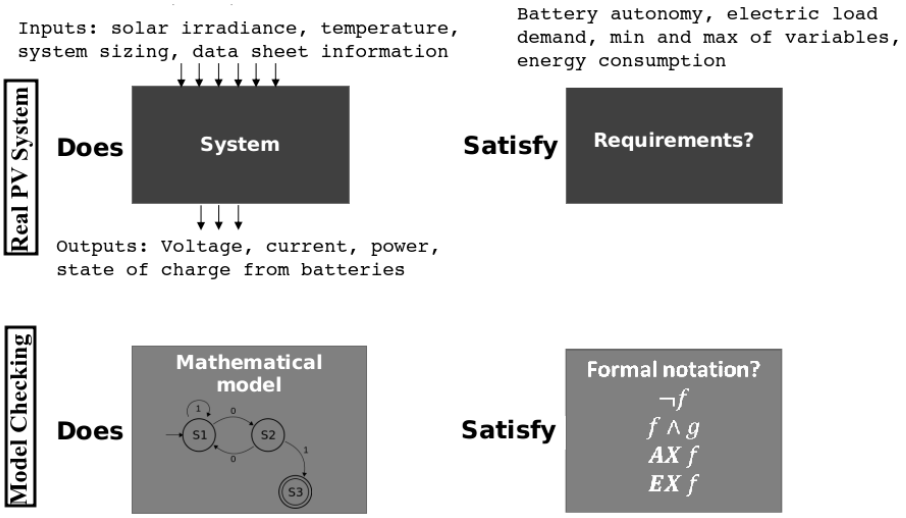


Figure 3.2: From real solar PV system verification to model checking. Source: adapted from (CLARKE, 2008).

The flowchart proposed for the automated verification method is illustrated in Fig. 3.3. The three steps are the high level description of how the automated verification method can be used to validate solar PV systems.

In **Step 1**, the PV input data and the formulae to check the sizing project, the mathematical model, the limits of the weather non-deterministic variables, are all written as an ANSI-C code (ISO, 2018).

In **Step 2**, the sizing check of the PV system takes place: indicating if there is a sizing error before performing the automated verification of the system. This stage ensures that the system follows the standard project steps of the critical period method of sizing (PINHO; GALDINO, 2014).

Assumption: before the automated verification, which performs the verification of the intended system behavior, the sizing check of the system is performed. On the discover of an error, the process stops, showing the sizing error.

Assumption: The sizing check is performed using critical period criteria.

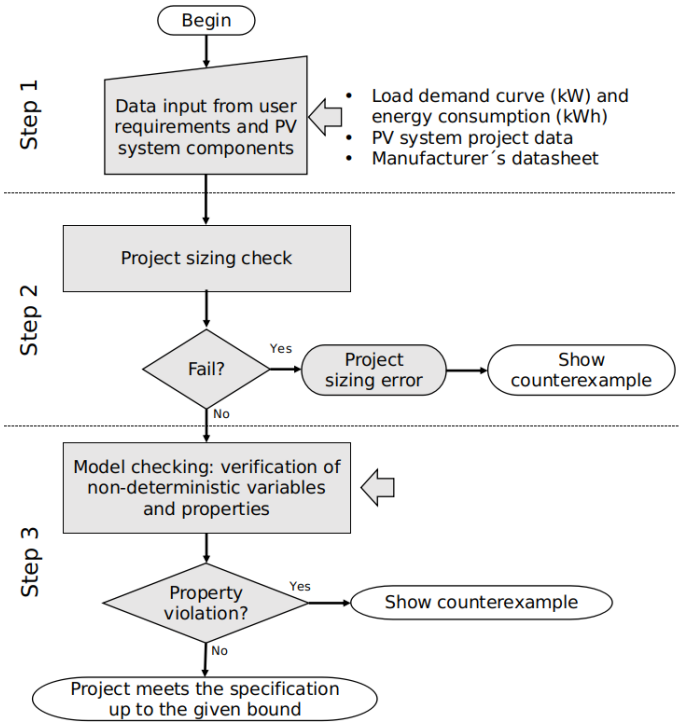


Figure 3.3: Flowchart of the proposed automated verification of PV systems.

In **Step 3**, weather variables (e.g., solar irradiance and ambient temperature) will be systematically explored by our verification engine based on maximum and minimum values from the site where the PV system will be deployed. In addition, depending on one of the desired properties of the system such as battery autonomy, energy availability, or even system power supply, our verification engine is able to indicate a failure if those properties are not met; in this particular case, it provides a diagnostic counterexample that shows in which conditions the property violation occurred.

In short, the model checker will process the ANSI-C code with constraints (C) and properties (P) of the PV system, and the tool will automatically verify if the PV system requirements are met. If it returns a failure (i.e. SAT), then the tool provides a counterexample, i.e. a sequence of states that leads to the property violation; this information can be used as feedback to improve the PV system design. However, if the verification succeeds (i.e. UNSAT), there is no failure up to the bound k; therefore, the PV system will present its intended behavior up to the bound k.

Algorithm 1 describes the equivalent pseudo-code. In order to reduce the computational effort of the algorithm, every 24 h-day was considered as a time-step of 1 hour, and it was split into two parts: (a) one where PV generation is possible, during daylight, with a duration in hours depending on each site (but dependent on the sun and weather conditions); and (b) one that includes the rest of the day (with no PV generation), when the batteries are required to supply energy to the house.

Assumption: The day is divided into two parts (representing presence or absence of PV generation, based on historical data from the location), considering average temperature and solar irradiance (for every hour of the day) and annual insolation (per day).

Lines 1 carries information from the location where the PV system will be/was deployed. We use the average annual minimum and maximum, both for temperature (T) and solar irradiance (G), hour by hour, from (WEATHERBASE, 2018), and (EnergyPlus, 2018).

Premise: Temperature, solar irradiance, and insolation data is available pertaining to the location where the solar PV system will be used.

Line 2 represents all the information that derives from the PV sizing and from the equipment manufacturer's data: PV, battery, inverter and charge controller specification and data. This item also includes information from the house's load curve.

Premise: The data sheet of every element of the solar PV system to be validated is available.

Premise: The detailed, sized project of the PV system is necessary in order to perform the validation (list of equipment and configuration, including voltage, current and how they are connected).

Assumption: The load curve of every house must be estimated or measured. Moreover, the time step is 1 hour, and seasonality is not considered, i.e. the load curve is the same for the entire year.

The first automated verification is the sizing check (line 3), if an error is found then the algorithm stops. In order to perform the sizing check, the algorithm uses Equations (2.37), and (2.38) to verify the PV panel, Equations (2.41), (2.42), (2.43), (2.44), and (2.45) to verify the batteries, Equations (2.39), (2.40), (2.46), (2.48) and (2.49) to verify the charge controller; and Equations (2.52), (2.53) and (2.54) to verify the inverter.

Then, if no sizing design flaw is found, two functions, called at lines 4 and 5, are responsible for discovering at which hour PV generation starts and when it stops. These functions receive this information from the array inputted to the Algorithm with the solar irradiance values.

The batteries are assumed to be charged, i.e. with SOC of 100% (line 6).

Assumption: Batteries are considered charged at the start of project validation.

The first for-loop at line 7 controls how many 24 hours cycleswill be performed by the Algorithm. And the for-loop from lines 8 to 11 is responsible for discharging the battery (according the load curve) and verifying the state of charge of the battery, hourly, from the first hour of the day after the sun sets to the next day before the sun rises (without PV generation). Then, at the next for-loop, from line 12 to 29, verification is performed where there is solar irradiance and the whole PV system works. The Algorithm generates information hourly related to average temperature (T) and solar irradiance (G), using non-deterministic variables from the model checker to explore all possible states and the *assume* macro to constrain the non-deterministic values using a given range (lines 15 and 16).

After that, the model of the PV generator is used in the function call of line 17, to produce the voltage and current considering the states of G and T. With respect to every hour considered, the conditional *if-elseif-endif* statements from lines 18, 20, 22, 24 and 26, will imitate the charge controller's work, as depicted in Table 2.3 of Section 2.3.7.4, performing the charge or discharge of batteries according to the value of the different variables: if there is PV generation, the updated battery state of charge, the house's load and the PV system set-up information.

At the end of the last for-loop, the state of the batteries is verified again (line 27) and the hour is adjusted to the next loop (line 28).

Nevertheless, if the verification engine does not fail, we can conclude that the PV system does not need further corrections up to the given bound k.

3.2 General Assumptions

In order to be clear, we now provide a list of the assumptions adopted by the scientific automated verification methods developed in this thesis.

The code in ANSI-C, was created to perform both methods, automated verification and automated synthesis, regardless of the verifier used. This means that:

- The '# include' preprocessor directive is not used, despite being very common in the C language, which allows the use of C language libraries, including the mathematical one;
- In the absence of the '# include' directive, it was necessary to create specific mathematical functions at the proposed code in order to calculate the exponential '$exp(x)$' or e^x, and natural logarithm '$log(x)$' functions, used for the solar PV model;
- There is no '# define' preprocessor directive. Therefore, global variables were

Algorithm 1 Model checking algorithm for validation of stand-alone PV systems

Input: mathematical model (PV, batteries, inverter, charge controller), weather data (temperature, solar irradiance), system sizing details, design requirements (load curve, peak demand, output voltage, battery autonomy), design assumptions (system availability, battery state of charge, 1-hour step of validation)

Output: design sizing check; FAIL with counterexample to improve the design; SUCCESS, saying that the project has no flaws

1: *declare min and max solar irradiance*[24*h*], *and temperature*[24*h*]
2: *declare case studies details* : *sizing and manufacturers data*
3: *sizing_check*()
4: $startPVgeneration \leftarrow findStartPVgeneration()$
5: $endPVgeneration \leftarrow findEndPVgeneration()$
6: $SOC \leftarrow 100\%$
7: **for** 1*st* 24*h loop* to *Nth* 24*h loop* **do**
8: **for** $endPVgeneration + 1$ to $startPVgeneration - 1$ **do**
9: $dischargeBattery\,in\,1h()$
10: $assert(SOC \geq SOC\_min)$
11: **end for**
12: **for** $startPVgeneration$ to $endPVgeneration$ **do**
13: $G \leftarrow nondet\_uint(\,)$ {G is non-deterministic variable}
14: $T \leftarrow nondet\_uint(\,)$ {T is non-deterministic variable}
15: assume $(Gmin \leq G \leq Gmax)$ {restricting G values}
16: assume $(Tmin \leq T \leq Tmax)$ {restricting T values}
17: $Imax, Vmax \leftarrow PVgenerationMODEL(G, T)$
 {If-then-else sequence to imitate charge controller work}
18: **if** (*battery is empty*) AND (*PV is generating*) **then**
19: $chargeBattery\,in\,1h()$ {PV feed the house}
20: **else if** (*battery is empty*) AND NOT(*PV is generating*) **then**
21: FAIL with assert macro {Battery is empty and there is not PV generation}
22: **else if** NOT(*battery is empty*) AND (*PV is generating*) **then**
23: stop battery charge {PV feed the house}
24: **else if** NOT(*battery is empty*) AND NOT(*PV is generating*) **then**
25: $dischargeBattery\,in\,1h()$ {Battery feed the house}
26: **end if**
27: $assert(SOC \geq SOC\_min)$
28: $hour \leftarrow hour + 1$
29: **end for**
30: **end for**
31: **return** ( )

used to replace it;

- The macro 'assert (expression);' must be replaced by 'if (!expression) { _ _ VERIFIER_ error();}';
- The macro 'assume (expression);' must be replaced by '_ _ VERIFIER_ assume(expression);';
- It is not possible to use the # if, # else, # elif, # endif or # ifdef, # ifndef commands.

Regarding the automated verification scientific method:

- A value of bound k was used to restrict the design-space and improve performance. The choice of value was empirical, following tests with the code;
- All the PV system model, user requirements, assumptions, and technical information from the PV system equipment are written as an ANSI-C code.

Of the off-the-shelf simulation tools, only HOMER Pro and Hybrid2 perform off-grid system with battery backup analysis. Additionally, HOMER and RETScreen include economic analysis or even optimization-sensitive analysis; however RETScreen does not have the capacity for stand-alone solar PV system analysis. Therefore, in this study, HOMER Pro will be the only simulation tool used to compare with our method.

For all case studies, the minimum battery state of charge was defined at 75%, and the efficiency of 86%, which is common to lead-acid batteries (adopted as standard here), and the AC voltage from the inverter at 127 V (Brazilian standard). According with the evaluated local of the PV systems installation, the considered insolation for the worst month is $3.8kWh/m^2$ per day.

3.3 Description of the Case Studies

Five case studies evaluated the proposed approach, as described in Table 3.1. These case studies were defined based on the usual electrical load found in riverside communities in the State of Amazonas, Brazil (TRINDADE; CORDEIRO, 2019; TRINDADE, 2013).

It is important to mention that the load curve represents an array of 24 integer numbers, one for each hour of the day (instant power in Watts) and it was estimated after visits to the communities and a survey applied in June 2017, before the houses were electrified.

Table 3.1: Case studies: stand-alone solar PV systems.

Item	House 1	House 2	House 3	House 4	House 5
PV Panels	3×325 W: (3S)				4×325 W: (2S-2P)
Batteries	4×220 Ah: (2S-2P) autonomy: 48 h				4×120 Ah: (4S) autonomy: 6 h
Charge Controller	With MPPT of 150 V/35 A				
Inverter	700 W, surge: 1,600 W				1,200 W, surge: 1,600 W
Load power peak (W)	342	253	263	322	814
Load power surge (W)	342	722	732	896	980
Load curve (W)	House 1: 118-118-118-46-46-46-95-95-170-170-296-242-242-95-95-95-95-342-288-288-288-288-118 House 2: 136-136-136-136-136-136-67-67-184-184-184-184-184-67-67-67-67-253-253-253-253-253-136 House 3: 113-113-113-113-113-113-67-67-217-97-97-97-97-97-97-97-97-263-113-113-113-113-113 House 4: 207-207-207-135-135-135-66-66-161-161-233-253-248-66-66-66-66-66-302-317-322-302-302-207 House 5: 45-16-16-16-16-16-0-0-0-72-72-222-150-150-0-0-72-72-814-814-814-742-742-16				
Consumption (kWh/day)	3.9	3.6	2.5	4.3	4.88
GPS Coordinates	2^o44'50.0"S 60^o25'47.8"W				3^o4'20.208"S 60^o0'30.168"W
Details	Riverside indigenous community Rural Area of Manaus - Brazil				Urban house Manaus-Amazonas-Brazil

Caption: (S): Series; (P): Parallel.

3.4 Objectives and Setup

The experimental evaluation aims to answer two experimental questions:

EQ1 **(soundness)** Does the automated verification approach provide correct results?

EQ2 **(performance)** How do the verifiers compare to each other and to a commercial simulation tool?

All experiments were conducted on an otherwise idle Intel Xeon CPU E5-4617 (8-cores) with 2.90 GHz and 64 GB RAM, running Ubuntu 16.04 LTS 64-bits. The setup of HOMER Pro v3.13.1 was an Intel Core i5-4210 (4-cores), with 1.7 GHz and 4 GB RAM, running Windows 10. The experiments were performed with a time out of 240 minutes.

The verification engine ESBMC, version v6.0.0 was used with the SMT solver Boolector version 3.0.1 (BRUMMAYER; BIERE, 2009)[1]; and an alternative ESBMC v6.0.0 was used with the 'incremental SMT' mode[2] enabled; with SMT solver Z3 version 4.7.1 (MOURA; BJØRNER, 2008).

The verification engine CBMC 5.11 and MiniSat 2.2.1 were used in the comparison (KROENING; TAUTSCHNIG, 2014)[3].

The verification engine CPAchecker 1.8 was used in a configuration of bound model

[1]Command-line: $ esbmc filename.c --no-bounds-check --no-pointer-check --unwind 100 --boolector

[2]Command-line: $ esbmc filename.c --no-bounds-check --no-pointer-check --unwind 100 --smt-during-symex --smt-symex-guard --z3

[3]Command-line: $ cbmc filename.c --unwind 100 --trace

Table 3.2: Summary of the case-studies comparative and the automated tools.

Model Checker (SAT/UNSAT: time, message, and used memory)				
Case	ESBMC 6.0.0 (Boolector 3.0.1)	ESBMC 6.0.0 (Z3 4.7.1)	CBMC 5.11 (MiniSat 2.2.1)	CPAchecker 1.8 (MathSAT 5.5.3)
House 1	Out of memory (UNKNOWN) $\geq$ 64 GB	$\leq$ 1 sec (SAT) 44 MB	Out of memory (UNKNOWN) $\geq$ 64 GB	4.36 s (SAT) 173 MB
House 2	Out of memory (UNKNOWN) $\geq$ 64 GB	$\leq$ 1 sec (SAT) 44 MB	Out of memory (UNKNOWN) $\geq$ 64 GB	4.34 s (SAT) 174 MB
House 3	Out of memory (UNKNOWN) $\geq$ 64 GB	$\leq$ 1 sec (SAT) 45 MB	Out of memory (UNKNOWN) $\geq$ 64 GB	4.38 s (SAT) 174 MB
House 4	Out of memory (UNKNOWN) $\geq$ 64 GB	$\leq$ 1 sec (SAT) 45 MB	Out of memory (UNKNOWN) $\geq$ 64 GB	4.46 s (SAT) 176 MB
House 5	Out of memory (UNKNOWN) $\geq$ 64 GB	$\leq$ 1 sec (SAT) 45 MB	Out of memory (UNKNOWN) $\geq$ 64 GB	4.19 s (SAT) 174 MB

Caption: (SAT): wrong sizing of PV panels (minimum total installed power not satisfied).

checking[4], with the SMT solver MathSAT version 5.5.3 (CIMATTI et al., 2013). An alternative CPAchecker configuration was also tried, using the BMC k-induction option, but the results show no improvement in performance or soundness (so it is not reported here).

3.5 Experimental Results and Discussion

This Section presents the results, with commented outputs produced by every tool, mainly related to the reports (CBMC and ESBMC) and some graphic resources (CPAchecker and ESBMC).

Table 3.2 summarizes the results. The times reported in Table 3.2 answer EQ2. Note that an UNKNOWN result from the proposed verification engines does not mean that a failure was found nor that the verification was successful: it indicates that the verification engine led to an *out of memory* situation and the design-state explosion was a issue during the run-time. It worth to mention that none of the verifiers and solvers reached the *time out* limit during the experimental evaluation.

The description of the experimental results have been broken down into three parts, one for each verification engine: ESBMC, CBMC, and CPAchecker.

3.5.1 ESBMC

Two alternatives were tested with ESBMC: one with Boolector and the another with Z3. The 'incremental SMT' option, which uses less memory, can be performed with

[4]Command-line: $ scripts/cpa.sh -heap 64000m -stack 10240k -config config/bmc-incremental.properties -spec config/specification/sv-comp-reachability.spc -benchmark filename.c

Z3 only since ESBMC does not support the 'incremental' mode with the Boolector solver yet.

Using ESBMC with Boolector led to an 'out of memory' situation in all the case studies. This result was obtained in less than six minutes of execution, i.e. the 64 GB of RAM were consumed by the verification engine and the processes were killed, thus leading to an UNKNOWN result returned by ESBMC as shown in the first column of Table 3.2.

However, running the same version of ESBMC, using 'incremental SMT' solving with Z3, the experimentation results returned were conclusive (SAT) in all the case studies. Moreover, the flaw identified was the same for all cases: the sized total PV panels power was not enough to meet the design requirements of the system (energy and power load demand). Put it simple, the designed PV systems will fail when deployed at the field. Based on the fact that the tool stops when the first flaw is identified, the rest of the validation procedure was not even reached. Worth to mention that if we remove the PV panels validation from the tool, the next flaw identified during experimentation was the battery array sizing (not enough to meet the design requirements as well).

In the houses that use a 975 W PV system (house 1, house 2, house 3, and house 4), ESBMC reached an error in all the four houses and the execution time took less than 01 s. Fig. 3.4 shows the output report of the tool for house 1, with the FAIL identified. The authors highlighted the result, which shows that the design flaw was found (SAT result or SUCCESS) within the bound k of 100, when a PV panels total power of $1,636.76$ W is expected and the sized system has only 975 W. This flaw is related to Eq. (2.38).

Related to the $1,300$ W PV system (house 5), the same fail occurred (SAT), thereby indicating that the system is *incorrectly* sized. Fig. 3.5 shows the report generated by the tool, with highlights for the result, showing the incompatibility of the sized PV panels array of $1,300$ W and the minimum necessary of $2,048.05$W. This verification took less than 1 s to perform, as indicated in the last line of Table 3.2.

3.5.2 CBMC

CBMC was unable to produce a SAT or UNSAT result. In less of the 25 minutes of code execution at the setup configuration, the verifier engine consumed all the available RAM memory and produced an UNKNOWN result, i.e the result was inconclusive for all houses.

```
ESBMC version 6.0.0 64-bit x86_64 linux
Enabling --no-slice due to presence of --smt-during-symex
file PVcode_rev3_house1.c: Parsing
Converting
Generating GOTO Program
GOTO program creation time: 0.134s
GOTO program processing time: 0.001s
Starting Bounded Model Checking
Symex completed in: 0.001s (76 assignments)
Slicing time: 0.000s (removed 1 assignments)
Generated 3 VCC(s), 1 remaining after simplification (75 assignments)
Encoding remaining VCC(s) using bit-vector arithmetic
Encoding to solver time: 0.001s
Solving with solver Z3 v4.7.1
Encoding to solver time: 0.001s
Runtime decision procedure: 0.001s

VERIFICATION FAILED
Building error trace

Counterexample:

State 1 file PVcode_rev3_house1.c line 73 thread 0
------------------------------------------------
  minSolarIrrad = { 0, 0, 0, 0, 0, 0, 25, 135, 274, 422, 509, 537, 503, 505, 430, 281, 80, 10, 0, 0, 0, 0, 0, 0 }
.
.
.
main
------------------------------------------------
  Pminpanels = 1636.7652740478515625f (00000110011001001100001111101001)

State 66 file PVcode_rev3_house1.c line 317 function sizing_check thread 0
sizing_check at PVcode_rev3_house1.c line 381
main
------------------------------------------------
Violated property:
  file PVcode_rev3_house1.c line 317 function sizing_check
  assertion
  FALSE
```

Figure 3.4: Report generated by ESBMC (Z3 solver) after validation of House 1.

3.5.3 CPAchecker

Finally, the CPAchecker tool presented some similar qualitative results but with different performance when compared with ESBMC (with Z3 solver). The verification engine presented an SAT result (a FAIL) for all the 975 W systems (house 1, house 2, house 3, and house 4) and to 1,300 W system.

Once again the problem of the minimum PV panels sizing expected was indicates as a flaw. The time to result was obtained in less than 5 seconds, around 4 seconds slower than ESBMC (with Z3); and the consumed memory by the CPAchecker was 3.86 time bigger than ESBMC (with Z3).

Figure 3.6 reproduces one of the text reports issued by CPAchecker. Unlike ESBMC and CBMC, CPAchecker produces many different reports, notably for log, statistical, and counterexample files. There is a HTML version of the counterexample which produces graphic output, not only text format.

Figure 3.7 shows the result for the 1,300 W house, with highlights on the line that causes the fail and part of the CFA (control-flow automata, represented as a control-flow graph) diagram pointing to this fail.

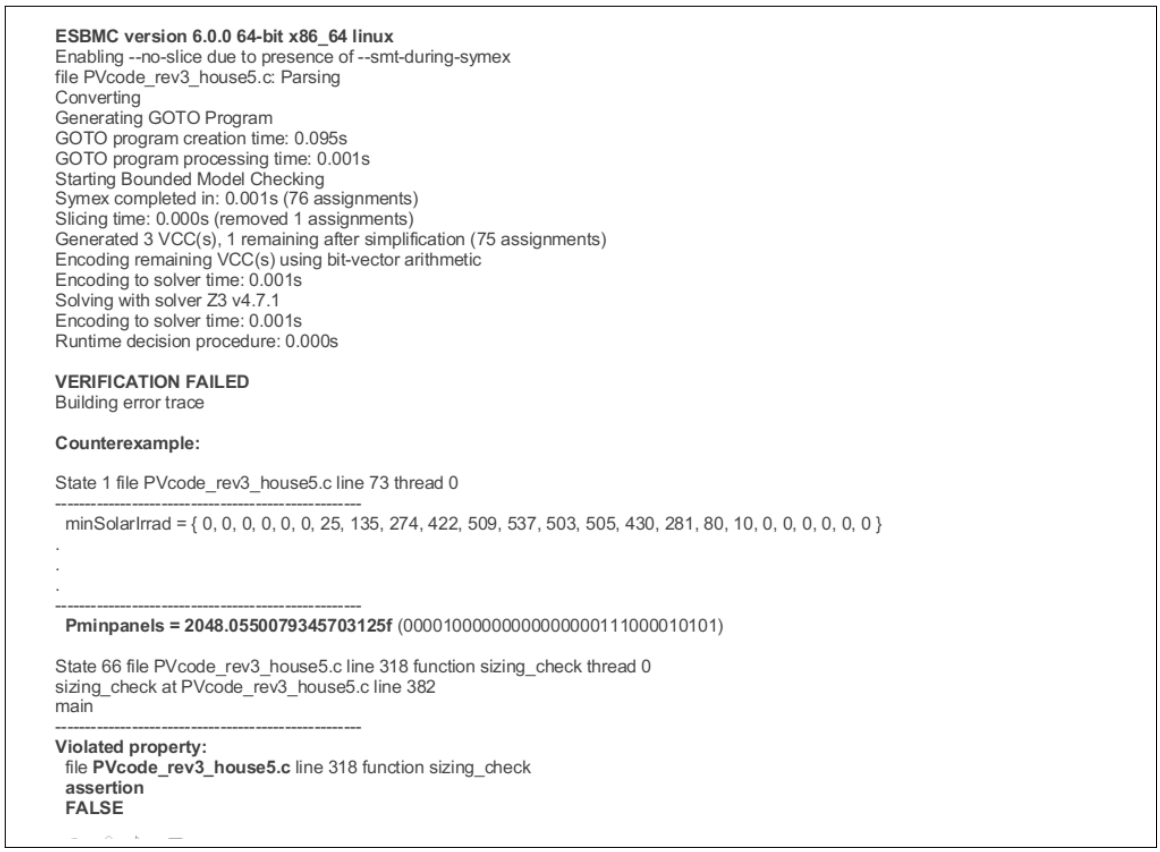


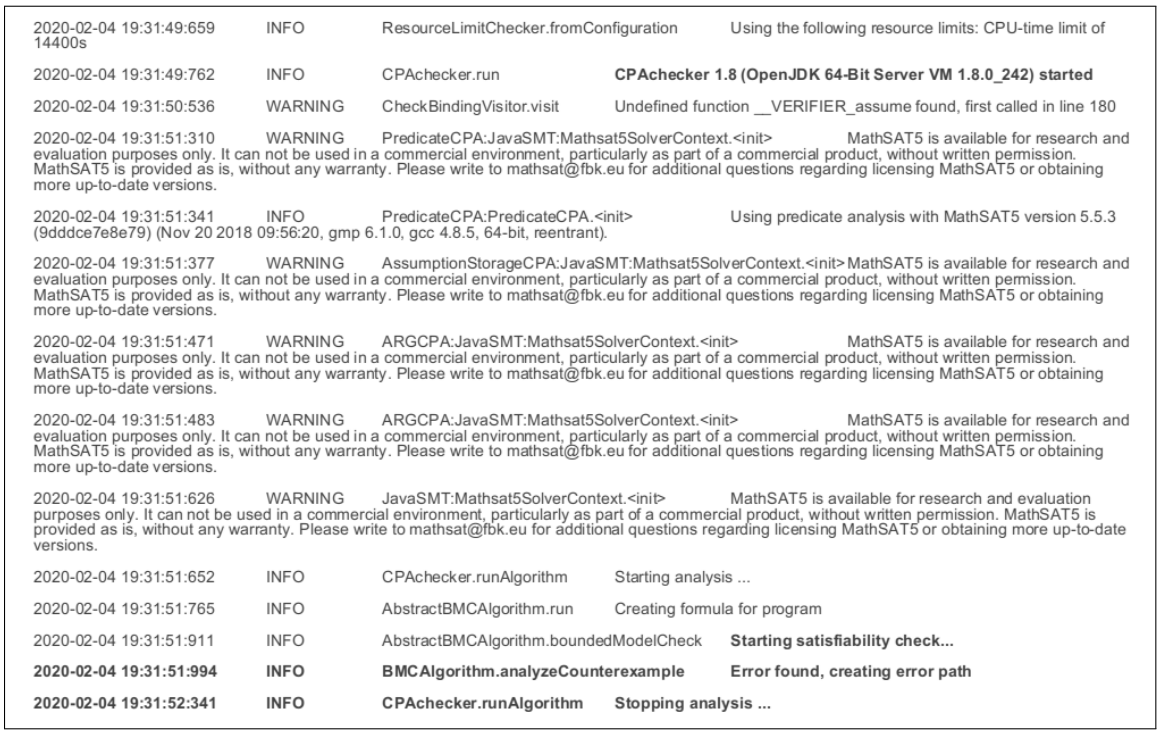
Figure 3.5: Report generated by ESBMC (Z3 solver) after validation of House 5.

3.5.4 HOMER Pro Specialized Simulation Tool

HOMER Pro is a powerful specialized electrical systems simulator, however even with the simulation characteristic, it performs optimization only when a given design is inputted to the schematic of the tool. This means that HOMER does not maintain the characteristic of the system under evaluation. It starts with the inputted information, but uses the optimization default set up to increase or decrease the characteristic of each component until the load is fulfilled by the electrical generation system, and at the lowest cost.

Therefore, in order to perform a limited simulation and not exceed the values of each component (for example, the total power of the PV array under evaluation), the 'search space' must be used for sizing instead of the 'HOMER Optimizer'. This is done on the menu of each component inputted in the schematic of the tool.

The PV panels and batteries: simulation of a particular capacity for PVs and batteris is possible using the 'search space' for sizing instead of the 'HOMER Optimizer'. It is possible either to aggregate all the PV panels as a single PV component, in which case, the search space must vary from 0 to 1, i.e. from the option without PV panels until it reaches the power inputted to the component in the HOMER schematic, or



2020-02-04 19:31:49:659 INFO ResourceLimitChecker.fromConfiguration Using the following resource limits: CPU-time limit of 14400s

2020-02-04 19:31:49:762 INFO CPAchecker.run **CPAchecker 1.8 (OpenJDK 64-Bit Server VM 1.8.0_242) started**

2020-02-04 19:31:50:536 WARNING CheckBindingVisitor.visit Undefined function __VERIFIER_assume found, first called in line 180

2020-02-04 19:31:51:310 WARNING PredicateCPA:JavaSMT:Mathsat5SolverContext.<init> MathSAT5 is available for research and evaluation purposes only. It can not be used in a commercial environment, particularly as part of a commercial product, without written permission. MathSAT5 is provided as is, without any warranty. Please write to mathsat@fbk.eu for additional questions regarding licensing MathSAT5 or obtaining more up-to-date versions.

2020-02-04 19:31:51:341 INFO PredicateCPA:PredicateCPA.<init> Using predicate analysis with MathSAT5 version 5.5.3 (9dddce7e8e79) (Nov 20 2018 09:56:20, gmp 6.1.0, gcc 4.8.5, 64-bit, reentrant).

2020-02-04 19:31:51:377 WARNING AssumptionStorageCPA:JavaSMT:Mathsat5SolverContext.<init> MathSAT5 is available for research and evaluation purposes only. It can not be used in a commercial environment, particularly as part of a commercial product, without written permission. MathSAT5 is provided as is, without any warranty. Please write to mathsat@fbk.eu for additional questions regarding licensing MathSAT5 or obtaining more up-to-date versions.

2020-02-04 19:31:51:471 WARNING ARGCPA:JavaSMT:Mathsat5SolverContext.<init> MathSAT5 is available for research and evaluation purposes only. It can not be used in a commercial environment, particularly as part of a commercial product, without written permission. MathSAT5 is provided as is, without any warranty. Please write to mathsat@fbk.eu for additional questions regarding licensing MathSAT5 or obtaining more up-to-date versions.

2020-02-04 19:31:51:483 WARNING ARGCPA:JavaSMT:Mathsat5SolverContext.<init> MathSAT5 is available for research and evaluation purposes only. It can not be used in a commercial environment, particularly as part of a commercial product, without written permission. MathSAT5 is provided as is, without any warranty. Please write to mathsat@fbk.eu for additional questions regarding licensing MathSAT5 or obtaining more up-to-date versions.

2020-02-04 19:31:51:626 WARNING JavaSMT:Mathsat5SolverContext.<init> MathSAT5 is available for research and evaluation purposes only. It can not be used in a commercial environment, particularly as part of a commercial product, without written permission. MathSAT5 is provided as is, without any warranty. Please write to mathsat@fbk.eu for additional questions regarding licensing MathSAT5 or obtaining more up-to-date versions.

2020-02-04 19:31:51:652 INFO CPAchecker.runAlgorithm Starting analysis ...

2020-02-04 19:31:51:765 INFO AbstractBMCAlgorithm.run Creating formula for program

2020-02-04 19:31:51:911 INFO AbstractBMCAlgorithm.boundedModelCheck **Starting satisfiability check...**

2020-02-04 19:31:51:994 INFO BMCAlgorithm.analyzeCounterexample Error found, creating error path

2020-02-04 19:31:52:341 INFO CPAchecker.runAlgorithm Stopping analysis ...

Figure 3.6: CPAchecker text result for house 1 validation (file CPALog.txt).

add each PV component to represent the series' configuration. For the batteries, it is necessary to find the equivalent capacity of the series and parallel configuration since HOMER considers only one battery component per simulation even if multiple battery components are added to the schematic.

Another drawback of HOMER Pro is the fact that the user cannot set battery autonomy, as the HOMER controller will decide when it is economical to discharge the batteries during the simulation year. It follows that simulation of battery autonomy cannot to be evaluated.

As part of the comparison proposed, four 975 W PV systems (houses 1, 2, 3 and 4) were evaluated by HOMER Pro (EQ2). The simulation results showed that for each of the sized systems, HOMER Pro concluded that no viable solution was found with the components inputted in the schematic. Fig. 3.8 shows one of the screens presented by HOMER Pro software, specifically by house 2.

Moreover, in order to evaluate the problem associated with the lack of a viable solution, some empirical changes were made to the project under evaluation. Fig. 3.9 shows that a configuration with higher battery capacity (3 strings of 2 batteries each, 2S-3P, of 220 Ah; instead of 2 strings of 2 batteries each as in the original sized system) can solve the problem of house 2. The same is true of the other three houses of $975W$, when improved battery capacity is presented by HOMER Pro as

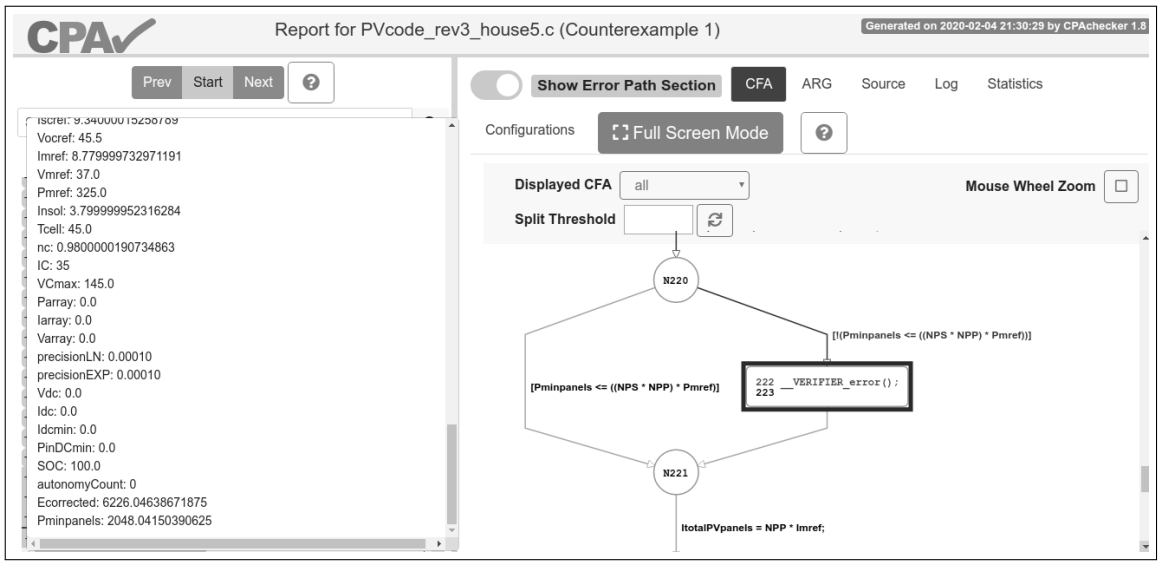


Figure 3.7: CPAchecker graphic result for house 5 validation (file Counterexample.html).

an optimal solution.

Further tests were performed in HOMER Pro. For example, to the case 1, that consumes 3.9 kWh/day of energy: with the optimizer feature turned on, the total of PV panels minimum power expected to meet the load requirement was 2, 530 W. For the 3.6 kWh/day of case 2, the PV panel solution indicated was 2, 420 W. For the 2.5 kWh/day of case 3, the PV panel solution indicated was 1, 590 W. And, finally, for the 4.3 kWh/day of case 4, the PV panel solution was 3, 150 W.

The case study that was not possible to simulate was the 1, 300 W (house 5). The reason is that HOMER Pro does not allow battery autonomy setup (because it always sizes and optimizes the electrical system in order to meet the load requirements for an entire year of simulation); therefore, it was not possible to obtain any indication about the failures of this specific PV system with simulation (EQ2).

3.5.5 Comparing Automated Verification and Simulation Results with Real PV Systems

Results do not diverged for houses 1, 2, 3, 4, and 5 with respect to the proposed approach: the automated evaluation shows there is project flaws, from the PV panels to the battery arrays sizing. Simulation demanded improvement of battery capacity (adding one more string of batteries to the system) and to the PV panels array (when the HOMER Pro tool was doing optimization of the original sized solution).

In order to evaluate the validation results, it was necessary to consult the interviews

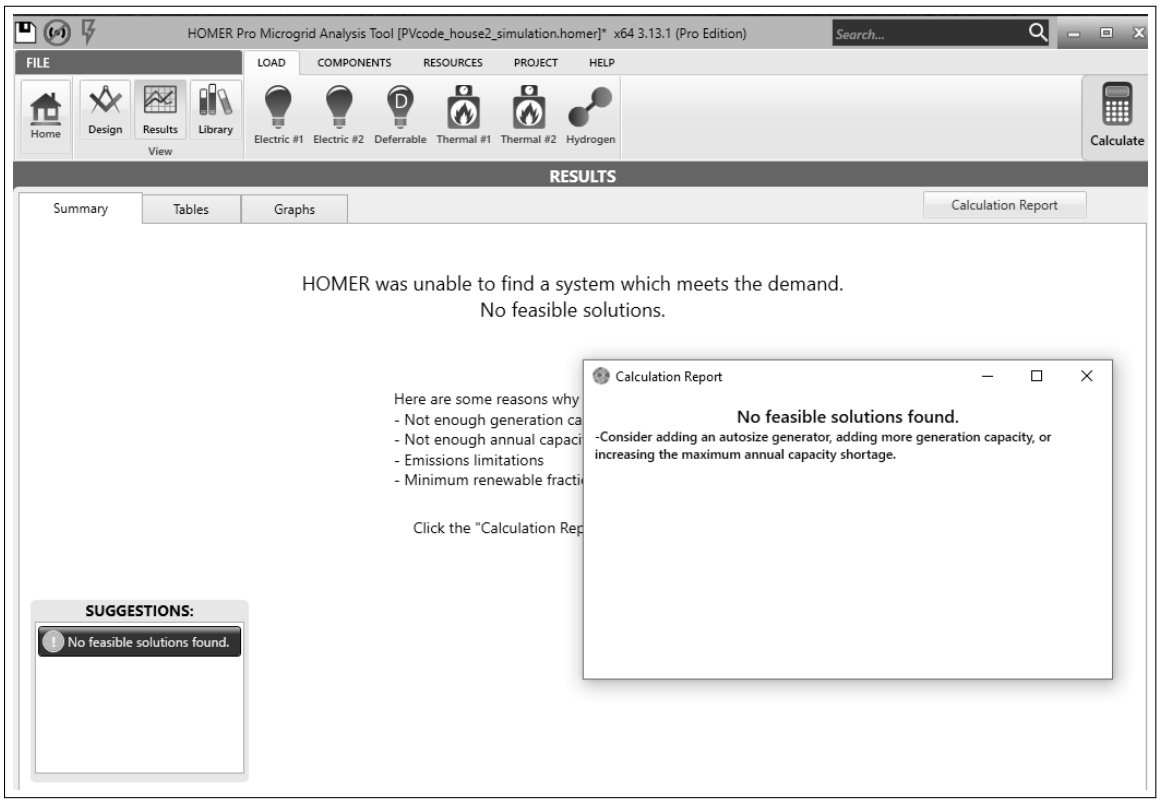


Figure 3.8: HOMER simulation screen presented for house 2 with no feasible solution found.

conducted with the residents of the 975 W deployed systems. From July 2018 to March 2019, surveys were applied to the residents on a monthly basis and data was collected from a local monitoring system: energy interruptions of the PV systems were not reported every month and, in some, cases, the maximum of 3 interruptions were reported. In one hand, when only one interruption is reported per month, this represents around 3.33% interruption for the entire period (1/30), which indicates 96.97% of availability of the PV system (96.67% = 100% − 3.33%). In other hand, when 3 interruptions are reported[5], this means 10% of interruption, or 90% of availability. In accordance to what was described in Section 2.3.8, the type of electrical load of the houses is not critical. Therefore any availability below of 95% is a sizing flaw, further affirming EQ1. Moreover, the proposed approach using automated verification provides the correct evaluation of the PV system, as the simulation did, thus answering EQ2.

The automated verification tools indicated flaws in house 5 as well; and was not possible to get HOMER Pro evaluation because of the autonomy restriction, further answering EQ2.

[5]Crossed information from interviews and collected data from the systems shows that the residents try to hide (or minimize) the interruption problems of the PV systems. Probably because they need of the system and they try not to emphasize this issue

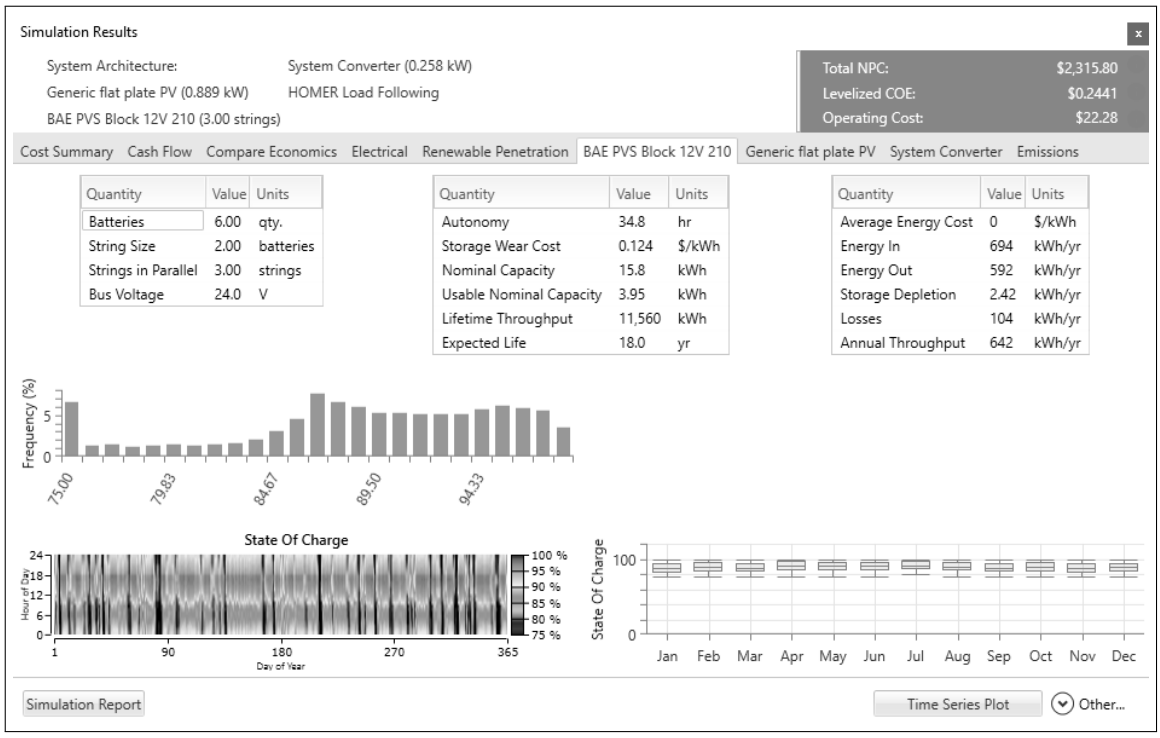


Figure 3.9: HOMER simulation screen presented for house 2 with viable solution.

In order to validate the possible flaw identified by the automated verification in house 5, the owner of the 1,300 W system was interviewed. From this it became clear that, in fact, the system does not meet the battery autonomy when all loads are turned on, and this was double-checked against the monitoring system of the charge controller, which showed that the maximum power or surge power was not exceeded. Even during the daylight, there is not enough energy produced by the solar PV panels to feed the loads, thus affirming EQ1; this behavior is to be expected since the system was purchased as an off-the-shelf solution and not as a customized design for the house's electrical charges.

3.6 Threats to Validity

This chapter presented a favorable assessment of the proposed method. Nevertheless, it also identified six threats to the validity of the results that bear further assessment.

Model precision: each component of the PV system is mathematically modeled. The adoption of more complex models, or even an evaluation in a PV laboratory to validate the model could add more reliability to the results.

Time step: The run-time complexity of the proposed method is an issue; the time step of one hour could be further reduced to approximate the algorithm to the real-world scenario.

Case studies: The case studies were conducted in only one municipality. A more complete evaluation can be done with more case studies.

Simulation tool: Only HOMER Pro was used. The inclusion of other specialized simulation tools or even a general simulation tool that uses the same mathematical model adopted by the automated verification could change the comparison.

Temperature and solar irradiance data: Information relating to the temperature and solar irradiance of each case study, whether it is using simulation or formal verification, comes from databases available online (WEATHERBASE, 2018; EnergyPlus, 2018). However, in view of the fact that riverside communities do not have weather stations, the data used in the present study comes from the municipality closest (Manaus in all the case studies), where stations regularly collect the data. Nevertheless, accuracy recommends the use of weather stations in each location.

Energy consumption estimation: All the PV size verification procedure start from the value of energy demanded from every house. However, this is a estimated value. Most recent information from field indicates that this value is overestimated and probably must be adjusted based on real data from a monitoring systems. The fact of the residents receive training to save energy is other issue that can reduce the total demanded energy from the PV systems. Therefore a most accurate validation can be performed when a real load curve is available for every case study.

3.7 Conclusion

This chapter showed in details how the state-of-the-art computer science method of automated verification was adapted to be used in stand-alone solar PV systems in order to validate their behavior. Moreover, it was possible to illustrate the comparative differences in how the proposed method and the traditional method of simulation work.

Detailed diagrams, flowcharts and algorithm with pseudo-code were presented in support of the proposed work and to aid the understanding. Additionally, the assumptions adopted in relation to the automated verification and simulation software were listed.

Furthermore, in this chapter verifiers with the proposed automated verification using model checking were compared, which demonstrated that 'incremental' ESBMC had the best overall performance. CBMC was not conclusive, presented out of memory issues in all cases. CPAchecker had similar quality results than ESBMC, but with a slower performance for all cases.

A comparison with a specialized simulation tool was also performed, although some of the tool's limitations, mainly related to not allowing battery autonomy setup. The need of better PV panels and/or battery sizing by HOMER Pro and the verifier engines showed the flaws identified by the tools.

Based on the fact that all case studies were deployed in the field, with regular visits to conduct interviews with the residents (house 1, house 2, house 3, and house 4), it was possible to compare the computational validation (by simulation or automated verification) with the real world employment of stand-alone solar PV systems. Although residents report few monthly problems in homes that use a 975 W solution, it is clear that in fact the interruption problem is much greater, as the original sized PV system is underestimated when it comes to power generation from solar panels and energy storage in batteries. The $1,300$ W system was classified as problematic by the owner, and our automated verification technique shows the design flaws. The final conclusion was that the proposed tool is sound and has an acceptable performance.

Chapter 4

Optimal Sizing of Stand-alone Solar PV Systems via Automated Formal Synthesis

In this chapter, we detail methodology used for optimal sizing of stand-alone solar PV systems, more specifically using model checking. Diagrams, flowcharts, and algorithm give support and explain the solutions.

In addition, we present all the underlying assumptions and premises. We also present the case studies used to evaluate the proposed approach for optimal sizing of stand-alone solar PV systems using automated synthesis. Moreover, the approach is compared with the HOME Pro simulation tool. The version and command-line of each verifier, the computing setup, the objectives of the experimental phase, and the results are also described.

It is important to emphasize that the complete explanation of the theoretical basis of the subject discussed here is presented in the chapter Background. In addition, knowledge of the literature is essential to aid understanding.

4.1 Methodology for Optimal Sizing of Solar PV Systems

Fig. 4.1 illustrates how to obtain optimal sizing of a stand-alone solar PV system, beginning with the traditional techniques (manual, and simulation), and moving on to the proposed automatic synthesis detailed in this Section.

Once more, as described in the automated verification process, the input information

is the same for all the methods, with the difference that in automated validation it is possible to define the bound k to restrict the design-space search. And the outputs are not equal, as in the design-space coverage and, mainly, in the final result.

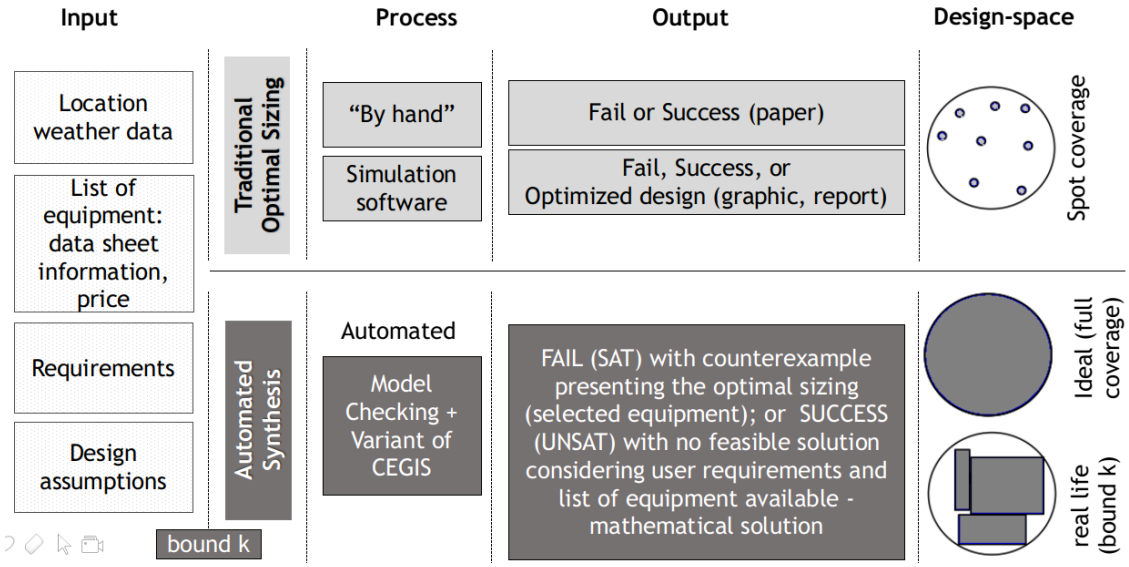


Figure 4.1: Comparison of optimal sizing methods

4.1.1 A Variant of CEGIS

Figure 4.2 illustrates a variant of CEGIS, previously presented in Section 2.2.5, Fig. 2.3. This variant was created during the research for this thesis and will be detailed in this Section.

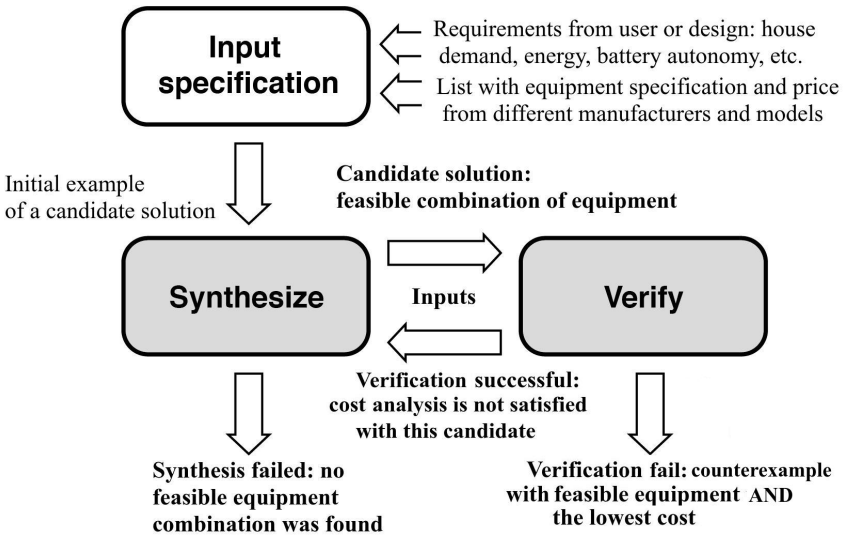


Figure 4.2: CEGIS applied to PV system sizing.

Examples of specification used by the proposed method include insolation (site dependent), house power demand, house energy consumption, estimated load curve, AC

voltage, and battery autonomy; we also provide a list of equipment specifications and prices from different manufacturers and models. Design assumptions are considered additional project specifications. The assumptions underlying optimal sizing are listed in Section 4.1.2.

The variant CEGIS used in the proposed approach differs in four specific aspects from the traditional CEGIS described in Figure 2.3:

- There is no test vector and every candidate is generated during the run-time in the SYNTHESIZE phase and sent to the VERIFY phase;
- If the VERIFY phase is unsuccessful, a new candidate is generated by SYNTHESIZE
- The lower bound of the VERIFY phase is incremented to search for the lowest cost;
- As a result, there is no refinement from the VERIFY phase back to the SYNTHESIZE phase, i.e. a new counterexample is not added to the INPUT set since a failure during the VERIFY phase will only discard a given candidate that could be viable in the next iteration with a new lower bound.

Program synthesis engines that implement the CEGIS approach (SOLAR-LEZAMA, 2013) can automatically produce solutions for a large variety of specifications; here we have used symbolic software verifiers based on SMT solvers.

Algorithm 2 describes our pseudo-code to synthesize stand-alone PV systems using symbolic model checking. The analytical method of optimization was adopted, with LCC economic analysis and power reliability based on critical period criteria.

Our synthesis algorithm will synthesize constant values; it starts with the input of the manufacturer's data and prices of PV panels, batteries, charge controllers and inverters. Moreover, we define design (house) requirements and design assumptions. The assumptions are present in sections 3.2 and 4.1.2.

The *for*-loop started at line 4 controls the lowest cost of the PV solution. In particular, it starts with cost 0 and stops only when the algorithm finds a feasible solution in which the cost breaks the *assertion* stated in line 21; if that happens, then our algorithm has found an optimal solution, thereby stating that the VERIFY phase reached a satisfiable condition (SAT). The $MaxCost$ value at line 2 is just a very high value inserted as a limit to the *for*-loop, that: (1) it will never be reached because the optimal solution will be found first (SAT result); or (2) it will be reached when the search engine did not find a feasible solution for the optimization (UNSAT result).

Algorithm 2 Synthesis algorithm

Input: weather data (temperature, solar irradiance); list of PV panels, controllers, batteries, and inverters datasheet information and cost; design requirements (load curve, peak demand, load surge power, daily energy consumption, battery autonomy, AC voltage); design assumptions (battery state of charge, depth of discharge, criteria and objectives for technical and cost analysis)

Output: FAIL (SAT) with counterexample showing the optimal sizing; SUCCESS (UNSAT), saying that the project has no feasible solution considering the requirements and the list of equipment

1: Initialize variables
2: Declare the maximum possible cost $MaxCost$
3: Calculate min possible Cost $MinCost$, based on the equipment list
4: **for** $HintCost = MinCost$ to $MaxCost$ **do**
5: Declare non-deterministic variable to select PV Panel from list
6: Declare non-deterministic variable to select Controller from list
7: Declare non-deterministic variable to select Battery from list
8: Declare non-deterministic variable to select Inverter from list
9: Calculate $E_{corrected}$, $P_{min,panels}$
10: Calculate and define PV panels arrangement N_{TP}, N_{PS}, N_{PP}
11: Requirement enforced by **assume**$(P_{min,panels} \leq (N_{PS} \times N_{PP} \times P_{m,ref}))$
12: Calculate $I_{c,min} = N_{PP} \times I_{m,ref}$
13: Calculate $V_{c,min} = N_{PS} \times V_{m,ref}$
14: Calculate C_{bank} and E_b
15: Calculate and define battery arrangement N_{BS}, N_{BP}, N_{BTotal}
16: Requirement enforced by **assume** $(I_{min,DCbus} \leq (N_{BP} \times Capacity))$
17: Controller requirements enforced by **assume**$((P_{controller} \geq P_{inverter}) \wedge V_c \wedge I_c)$
18: Inverter requirements enforced by **assume**$(V_{in}DC \wedge V_{out}AC \wedge Demand \wedge P_{surge})$
19: non-deterministic variables hold feasible equipment and cost
20: $F_{obj} \leftarrow N_{TP} * Panel_{Cost} + N_{TB} * Battery_{Cost} + Controller_{Cost} + Inverter_{Cost} + Installation_{Cost} + batrep_{Cost} + PWO\&M_{Cost}$
21: Violation check with **assert**$(F_{obj} > HintCost)$
22: **end for**
23: **return** ( )

Our synthesis algorithm uses non-deterministic variables to choose one specific constant from a given list of PV panels, controllers, batteries and inverters (lines 5 to 8). This procedure ensures that our synthesis engine checks all combinations of items from each equipment, and combines them to assemble a viable (candidate) PV solution, which meets user requirements.

Next, we use from Eq. (2.37) to Eq. (2.48) to calculate the sizing variables (lines 9 to 15). The directive *assume* (lines 11, 16, 17 and 18) ensures compatibility of the items chosen from the list of equipment: the VERIFY phase uses only items (among all the possible ones) that satisfy the statements of lines 11, 16, 17 and 18. Line 11 is specific to PV panels. Line 16 is for the battery bank. Line 17 is the charge controller voltage check. Line 18 does the inverter check and ensures the power demand and the surge power of the inverter. Therefore, our synthesis algorithm reaches line 19 with one feasible solution, and the cost of that solution is calculated in F_{obj} (line 20). This cost is the equivalent to 2.55, as described in Section 2.3.10.

If our algorithm does not find a feasible solution among the item of equipment that were provided for our SYNTHESIZE phase, then the result is unsatisfiable ($UNSAT$), i.e. the program finishes without finding a solution, indicating that it was not possible to combine the specific items of equipment in order to create a feasible solution.

The main challenge for the SYNTHESIZE phase is to find a feasible candidate solution for the constraints and user requirements. The challenge for the VERIFY phase is to find the lowest acquisition cost from a list of equipment and components provided by the SYNTHESIZE phase.

Note that the process described here is completely automated and that a validation is performed by our VERIFY phase to ensure that the approach is sound.

The Algorithm 2 is transformed by the verification engines into the Boolean expressions that are passed to the solver in order to verified ($C \wedge \neg P$). The quantifier-free formulae C and P are shown in Eq. 4.1 and Eq. 4.2, respectively. Where *nd_uchar* denotes a non-deterministic unsigned char value. The conditional expression $ite(f, t_1, t_2)$ takes a Boolean formula f and depending on its value selects either the second or the third argument. The function $select(a, i)$ denotes the value of a at index position i. Worth to mention that it was removed from the representation all the bounds checks for arrays and matrices, and the division by zero check. During the verification process, the arithmetic operators induce checks to ensure that the arithmetic operations

do not overflow and/ or underflow.

$$
C := \left[\begin{array}{l}
panelchoice = nd\_uchar \\
\wedge\ batterychoice = nd\_uchar \\
\wedge\ controllerchoice = nd\_uchar \\
\wedge\ inverterchoice = nd\_uchar \\
\wedge\ P_{m,ref} = select\,(PanelData\,[panelchoice][8]) \\
\wedge\ I_{m,ref} = select\,(PanelData\,[panelchoice][9]) \\
\wedge\ V_{m,ref} = select\,(PanelData\,[panelchoice][10]) \\
\wedge\ Panel_{Cost} = select\,(PanelData\,[panelchoice][12]) \\
\wedge\ \eta_b = select\,(BatteryData\,[batterychoice][0]) \\
\wedge\ Vbat = select\,(BatteryData\,[batterychoice][1]) \\
\wedge\ Capacity = select\,(BatteryData\,[batterychoice][2]) \\
\wedge\ Battery_{Cost} = select\,(BatteryData\,[batterychoice][5]) \\
\wedge\ \eta_c = select\,(ControllerData\,[controllerchoice][0]) \\
\wedge\ I_c = select\,(ControllerData\,[controllerchoice][1]) \\
\wedge\ V_{c,max} = select\,(ControllerData\,[controllerchoice][4]) \\
\wedge\ Controller_{Cost} = select\,(ControllerData\,[controllerchoice][5]) \\
\wedge\ \eta_i = select\,(InverterData\,[inverterchoice][0]) \\
\wedge\ V_{in,DC} = select\,(InverterData\,[inverterchoice][1]) \\
\wedge\ V_{out,AC} = select\,(InverterData\,[inverterchoice][2]) \\
\wedge\ P_{AC,ref} = select\,(InverterData\,[inverterchoice][3]) \\
\wedge\ MAX_{AC,ref} = select\,(InverterData\,[inverterchoice][4]) \\
\wedge\ Inverter_{Cost} = select\,(InverterData\,[inverterchoice][5]) \\
\wedge\ E_{corrected} = E_{consumption}/(\eta_b \times \eta_c \times \eta_i) \\
\wedge\ P_{min,panels} = 1.25 \times E_{corrected}/Insolation \\
\wedge\ N_{Pmin} = (int)(P_{min,panels} - 1)/Pm,ref + 1 \\
\wedge\ N_{PP} = 2 \\
\wedge\ N_{PS} = ite(((N_{Pmin}\%2) == 0), (N_{Pmin}/2), ((N_{Pmin} + 1)/2)) \\
\wedge\ N_{TP} = N_{PP} \times N_{PS} \\
\wedge\ I_{total,PVpanels} = 2 \times I_{m,ref} \\
\wedge\ V_{total,PVpanels} = N_{PS} \times V_{m,ref} \\
\wedge\ DOD_{max} = (100 - SOC_{min}) \times 2 \\
\wedge\ E_b = (autonomy/24) \times E_{corrected} \times 100/DOD_{max} \\
\wedge\ I_{min,DCbus} = E_b/(float)V_{system} \\
\wedge\ N_{BS} = 24/Vbat \\
\wedge\ N_{BP} = ((I_{min,DCbus} - 1)/Capacity + 1) \\
\wedge\ N_{Btotal} = N_{BS} \times N_{BP} \\
\wedge\ F_{obj} = N_{TP} \times Panel_{Cost} + N_{Btotal} \times Battery_{Cost} + Controller_{Cost} + Inverter_{Cost} + O\&M_{Cost} + Inst_{Cost}
\end{array}\right] \tag{4.1}
$$

$$
P := \left[\begin{array}{l}
((panelchoice \geq 0)\ \wedge\ (panelchoice \leq 9)) \\
\wedge\ ((batterychoice \geq 0)\ \wedge\ (batterychoice \leq 9)) \\
\wedge\ ((controllerchoice \geq 0)\ \wedge\ (controllerchoice \leq 9)) \\
\wedge\ ((inverterchoice \geq 0)\ \wedge\ (inverterchoice \leq 9)) \\
\wedge\ (I_c \geq I_{total,PVpanels}) \\
\wedge\ (V_{c,max} \geq V_{total,PVpanels}) \\
\wedge\ ((V_{c,max} \times I_C \times \eta_c) \geq P_{AC,ref}) \\
\wedge\ (V_{in,DC} \geq 24) \\
\wedge\ ((V_{out,AC} \geq OutletVoltage - 10\%)\ \wedge\ (V_{out,AC} <= OutletVoltage + 10\%)) \\
\wedge\ ((MaximumLoadPower \leq P_{AC,ref})\ \wedge\ (SurgeLoadPower \leq MAX_{AC,ref})) \\
\wedge\ (F_{ob} > Value_{Iterative,optimal})
\end{array}\right] \tag{4.2}
$$

4.1.2 Optimization Premises and Assumptions

This section contains premises and assumptions underlying the optimal sizing method, in relation to automated synthesis and simulation software.

As input of Algorithm 2, the synthesis engine is provided with a list of forty items of equipment from ten different manufacturers in order to allow the engine to choose from among all items of PV sizing. The necessary technical information was collected from data sheet of each item. In addition, the price of each item was obtained from available market quotation from internet, in US dollars. Preferably, we sought equipment sold in the Brazilian market, in Reais R\$, having been converted to US dollars using the exchange rate of the day (May and June of 2019).

With respect to power reliability, the critical period solar energy method will be used (PINHO; GALDINO, 2014), as described in Section 2.3.9. The usual way is to use loss of load probability (LOLP) or loss of power supply probability (LPSP). However, due to the fact that in this study we are considering neither site characteristics nor load changes over time, which demand historical data, the reliability analysis will be developed only by the critical period method of PV sizing.

Financial analysis details:

- LCC lifetime considered: 20 years;
- Installation costs: these include delivery to the isolated community and the actual installation costs: 5% of total cost (TRINDADE, 2013);
- Value of the discount rate or interest rate: 10%, a reasonable rate, considering financial investments in developing countries;
- Annual operation and maintenance costs: based on past PV projects of similar size in the Brazilian Amazon, the sum of US\$ 289.64 (TRINDADE, 2013) will be adopted. This cost includes battery replacement based on a lifetime of 4 years for lead-acid batteries, plus inverters and controller replacement (every 8 years). This means that there will be four battery bank and two inverter-controller replacement during the LCC analysis.

The PV system optimization technique adopted here is the intuitive method, since the average daily value of solar irradiance is used in the mathematical model, without considering battery state of charge, or even the random nature of solar irradiation and meteorological conditions. Therefore, all the computational effort will be concentrated on our automated synthesis algorithm.

With regard to batteries, the voltage of the system (DC bus) is set at 24 V DC, but this can be adjusted to 12 or 48 V in the code.

HOMER Pro details:

- HOMER Pro does not provide for the LCC cost in its reports. However, it has NPC and LCOE. For this reason, NPC was used to obtain LCC in order to allow the comparison among tools;

- The optimization analysis of HOMER Pro permits the definition of a load curve and temperature on the basis of data collected automatically from online databases. However, in order to enable a correct comparison, the curve load and the temperature were defined exactly the same as automated synthesis tools;

- Battery autonomy is not a parameter that the user can set when using HOMER Pro. The tool will always try to meet the system requirement, i.e. the load curve, 365 days a year;

- HOMER Pro does not have an item of equipment explicitly labeled charge controller. It uses a controller resource that can operate in two different ways, depending on optimization choice or user choice: 'load following' or 'cycle charging' (HOMER, 2017). During the tests 'load following' controller was chosen: it produces only enough power to meet the demand (HOMER, 2017);

- A 5% of capacity shortage was assumed, equivalent to 95% availability of the PV system. By definition, availability is the percentage of time during which a power system is capable of meeting the load requirements (KHATIB; ELMENREICH, 2014). For critical loads, 99% is considered acceptable, while in an ordinary residential electrical load, 95% is considered acceptable;

- A string of two batteries was assumed in order to match the system voltage of 24 V DC used by the automated synthesis tool;

- The premise adopted when using HOMER Pro was that the user does not know the optimal solution, and that in order to obtain this solution it is necessary to include (at the design phase of the tool) generic PV and battery modules that HOMER Pro will search for the optimized power of each component. With that in mind, a generic flat plate PV of 1 kW was included and generic lead-acid batteries of 1 kW also (and with capacity of 83.4 Ah in accordance with HOMER Pro modeling). HOMER, during run-time, decides the size in kW of each module, based on feasibility and lower cost.

4.2 General Assumptions

The general assumptions for the scientific method of automated synthesis are the same presented in Section 3.2, with respect to the code in ANSI-C, to the bound k, to the simulation tool comparison, and to the battery state of charge.

4.3 Description of the Case Studies

The proposed synthesis approach was evaluated in seven stand-alone PV system case studies:

- Case Study 1: Power peak: 342 W, power surge: 342 W, energy consumption: 3,900 Wh/day, battery autonomy: 48 h;
- Case Study 2: Power peak: 814 W, power surge: 980 W, energy consumption: 4,880 Wh/day, battery autonomy: 48 h;
- Case Study 3: Power peak: 815 W, power surge: 980 W, energy consumption: 4,880 Wh/day, battery autonomy: 12 h;
- Case Study 4: Power peak: 253 W, power surge: 722 W, energy consumption: 3,600 Wh/day, battery autonomy: 48 h;
- Case Study 5: Power peak: 263 W, power surge: 732 W, energy consumption: 2,500 Wh/day, battery autonomy: 48 h;
- Case Study 6: Power peak: 322 W, power surge: 896 W, energy consumption: 4,300 Wh/day, battery autonomy: 48 h;
- Case Study 7: Power peak: 1,586 W, power surge: 2,900 W, energy consumption: 14,000 Wh/day, battery autonomy: 48h.

These case studies were defined based on the usual electrical load found in riverside communities in the State of Amazonas, Brazil (TRINDADE; CORDEIRO, 2019; TRINDADE, 2013), with the exception of case 7, that was idealized as a small town solution to support a few lamps and a 12 kBTUs air-conditioner.

4.4 Objectives and Setup

The evaluation aims to answer two experimental questions:

EQ1 **(soundness)** does the proposed automated synthesis approach provide correct results?

EQ2 **(performance)** how do the software verifiers compare to each other?

All experiments regarding the verification tools were conducted on an otherwise idle Intel Xeon CPU E5-4617 (8-cores) with 2.90 GHz and 64 GB RAM, running Ubuntu 16.04 LTS 64-bits. For HOMER Pro, an Intel Core i5-4210 (4-cores) was used, with 1.7 GHz and 4 GB RAM, running Windows 10. The experiments were conducted with a predefined time-out of 660 minutes.

Three start-of-the-art verification tools, CBMC[1] version 5.11 with MiniSat 2.2.1 (KROENING; TAUTSCHNIG, 2014), ESBMC[2] version 6.0.0 configured for incremental loop unwinding verification with the Boolector 3.0.1 solver (BRUMMAYER; BIERE, 2009), and CPAchecker[3] version 1.8 in a configuration as bound model checking and with MathSAT 5.5.3 (CIMATTI et al., 2013), were used as verification engines to compare the proposed approach effectiveness and efficiency. An "incremental" ESBMC with the SMT solver Z3 version 4.7.1 (MOURA; BJØRNER, 2008) was tried[4] as an alternative to use less computing memory.

The Simulation tool HOMER Pro version 3.13.1 was used for comparative purposes.

4.5 Experimental Results

The results are presented at Table 4.1.

The violation (SAT result) indicated all over the Table 4.1 is the *assert* of line 21 from Algorithm 1 and indicates that an optimal solution was found.

CBMC was unable to produce any conclusive result. *Out of memory* situations occurred in all case studies.

CPAchecker was able to synthesize optimal sizing in three out of the seven case studies (cases 1, 4, and 5): the result was produced within the time limit, which varied from 254.25 to 548 minutes. Fig. 4.3 illustrates the result of case 5 with the optimal sizing appearing on the left size as the integer 3 for the solar panel (which is the Canadian CS6U-330P model of 330 W from the manufacturer Victron Energy), battery 0 refers to the model 12MF80 of 80 Ah from Moura, charge controller 0 refers to the model 35A-145V MPPT from Victron Energy, and the inverter number 2 refers to the Epever model IP350-11 of 280 W (nominal power) and 750 W of surge power. The variables NTP, NPS, NPP, NPS, NBP, and NBTotal, also presented in the counterexample as well, shows the number of panels and batteries and how they

[1]Command-line: $ cbmc --unwind 100 filename.c --trace

[2]Command-line: $ esbmc filename.c --incremental-bmc --boolector

[3]Command-line: $ scripts/cpa.sh -heap 64000m -config config/bmc-incremental.properties -spec config/specification/sv-comp-reachability.spc -benchmark filename.c

[4]Command-line: $ esbmc filename.c --no-bounds-check --no-pointer-check --unwind 100 --smt-during-symex --smt-symex-guard --z3

Table 4.1: Case studies and results: optimization of stand-alone PV systems.

Tools	CBMC 5.11 (MiniSat 2.2.1)	ESBMC 6.0.0 (Boolector 3.0.1)	CPAchecker 1.8 (MathSAT 5.5.3)	HOMER Pro 3.13.1
Specification	Result	Result	Result	Result
Case Study 1 Peak:342W Surge:342W E:3,900Wh/day Autonomy:48h	OM	SAT (620 min) NTP:6×330W (2P-3S) NBT:16×105Ah (2S-8P) Controller 35A/145V Inverter 700W/48V LCC: US$ 10,214.04	SAT (548 min) NTP:6×330W (2P-3S) NBT:16×105Ah (2S-8P) Controller 35A/145V Inverter 700W/1600W/48V LCC: US$ 10,214.04	(Time: 0.33 min) 2.53 kW of PV NBT:12×83.4Ah (2S-6P) 0.351kW inverter LCC: US$ 7,808.04
Case Study 2 Peak:814W Surge:980W E:4,880Wh/day Autonomy:48h	OM	TO	TO	(Time: 0.18 min) 3.71 kW of PV NBT:20×83.4Ah (2S-10P) 0.817kW inverter LCC: US$ 12,861.75
Case Study 3 Peak:815W Surge:980W E:4,880Wh/day Autonomy:12h	OM	SAT (63 min) NTP:14×150W (7P-2S) NBT:6×105Ah (2S-3P) Controller 35A/145V Inverter 1,200W/48V LCC: US$ 9,274.07	TO	Not possible
Case Study 4 Peak:253W Surge:722W E:3,600Wh/day Autonomy:48h	OM	SAT (147 min) NTP:6×330W (2P-3S) NBT:16×105Ah (2S-8P) Controller 35A/145V Inverter 280W/48V LCC: US$ 9,678.63	SAT (605 min) NTP:6×330W (2P-3S) NBT:16×105Ah (2S-8P) Controller 35A/145V Inverter 280W/48V LCC: US$ 9,678.63	(Time: 0.23 min) 2.42 kW of PV NBT:12×83.4Ah (2S-6P) 0.254kW inverter LCC: US$ 7,677.95
Case Study 5 Peak:263W Surge:732W E:2,500Wh/day Autonomy:48h	OM	SAT (36.70 min) NTP:4×330W (2S-2P) NBT:14×80Ah (2S-7P) Controller 35A/145V Inverter 280W/24V LCC: US$ 8,900.70	SAT (254.25 min) NTP:4×330W (2S-2P) NBT:14×80Ah (2S-7P) Controller 35A/145V Inverter 280W/24V LCC: US$ 8,900.70	(Time: 0.18 min) 1.59 kW of PV NBT:10×83.4Ah (2S-5P) 0.268kW inverter LCC: US$ 6,175.57
Case Study 6 Peak:322W Surge:896W E:4,300Wh/day Autonomy:48h	OM	SAT (380.93 min) NTP:6×320W (2P-3S) NBT:18×105Ah (2S-9P) Controller 35A/145V Inverter 400W/24V LCC: US$ 10,136.61	TO	(Time: 0.22 min) 3.15 kW of PV NBT:14×83.4Ah (2S-7P) 0.328kW inverter LCC: US$ 9,112.45
Case Study 7 Peak:1,586W Surge:2,900W E:14,000Wh/day Autonomy:48h	OM	UNSAT (0.48 min)	TO	(Time: 0.20 min) 12.5 kW of PV NBT:66×83.4Ah (2S-33P) 1.60kW inverter LCC: US$ 41,878.11

Legend: OM = out of memory; TO = time out; IF = internal failure, E = energy.

are connected.

Case studies 2, 3, 6, and 7 led to a *time out* result in CPAchecker, i.e. it was not solved within 660 minutes.

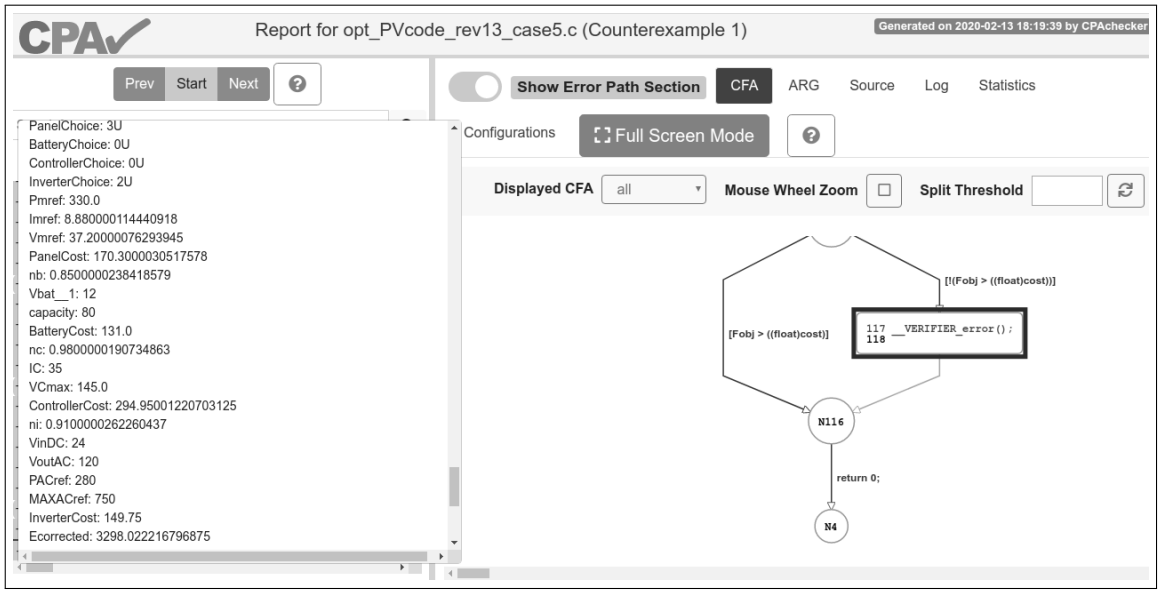


Figure 4.3: Counterexample generated by CPAchecker after validation of case 5 (file Counterexample.html).

An internal failure was presented by ESBMC when using Z3 solver in all case studies. It corresponds to a 'bug' in the Z3 solver which will require an updated version of ESBMC to fix it. However, when using the incremental BMC configuration of ESBMC with Boolector solver, the verifier engine was able to reach the optimal sizing of case studies 1, 3, 4, 5, and 6 with a FAIL/ SAT response varying from 36.70 to 620 minutes. Fig. 4.4 shows the counterexample generated. ESBMC with this configuration was not possible to obtain a optimal solution in cases 2 and 7. Case 2 reached a *time out* result after 660 minutes of processing. Moreover, case 7 resulted in a UNSAT result, i.e the verifier engine was not possible to get a feasible solution considering the list of equipment. This SAT situation was reached after just 0.48 minutes.

Those specific results partially answers the *EQ2*.

HOMER Pro was able to evaluate six case studies (cases 1, 2, 4, 5, 6, and 7), and in under 30 seconds, much faster than the proposed automated synthesis tool (cf. *EQ2*). Case study 3 could not be simulated since HOMER Pro does not have the battery autonomy adjustment feature, i.e. the tool always tries to feed the given load with electricity 365 days/year.

Fig. 4.5 illustrates parts of the 9-page PDF report presented by HOMER Pro,

Figure 4.4: ESBMC counterexample for case 3 optimization.

specifically for case 4.

Certain other HOMER Pro drawbacks were also noted:

- System equipment does not include an explicit charge controller . HOMER Pro includes a controller automatically just to simulate the charge/discharge of batteries and to meet the load requirement; however, without costs or even with electrical characteristics such as maximum current and voltage, which are common during PV sizing;

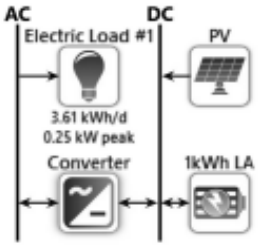
- HOMER Pro requires the inclusion of some battery specification to initiate optimization; however, it does not change the electrical specifications during simulation; the results presented are multiples of the original battery type suggested by the user. For example, it was started with a 83.4 Ah lead-acid battery and during simulation, HOMER Pro did not try to use other capacities or types;

- HOMER Pro does not present the optimal solution in terms of connections of PV panel arrays, just the total in terms of power, i.e. it presents neither the models and the power of each PV panel nor the total of panels in series or parallel.

System Architecture

Component	Name	Size	Unit
PV	Generic flat plate PV	2.42	kW
Storage	Generic 1kWh Lead Acid	6	strings
System converter	System Converter	0.254	kW
Dispatch strategy	HOMER Load Following		

Schematic



Net Present Costs

Name	Capital	Operating	Replacement	Salvage	Resource	Total
Generic 1kWh Lead Acid	$1,620	$1,192	$3,222	$0.00	$0.00	$6,034
Generic flat plate PV	$1,236	$240.31	$0.00	-$54.60	$0.00	$1,422
System Converter	$114.21	$0.00	$53.67	$0.00	$0.00	$167.88
System	$2,970	$1,432	$3,275	-$54.60	$0.00	$7,623

Figure 4.5: Optimization report (partial) from HOMER Pro (case 4).

4.5.1 Comparison Between Formal Synthesis and HOMER Pro

Comparing the results of the formal synthesis with CPAchecker, and ESBMC, against those of HOMER Pro, it was observed some distinct results, in terms of technical solution and cost (cf. Table 4.1). Related to performance there is a huge difference in favor of HOMER Pro that obtained the results in considerably less time: few seconds in opposite of an average of 234 minutes for the automated synthesis technique.

Particularly in the case of LCC, the cost varied from 11% to 44%, always having a higher estimation from the automated synthesis technique. However, considering that the cost of individual item of each database used to compose the optimal design is not the same among the tools, it is plausible to obtain distinct results.

In one hand, related to the PV panels sizing, the results presented by the automated synthesis were smaller in terms of power than the produced by the simulation tool. The difference varied from 19% to 65%. In the other hand, related to the battery bank, the results were smaller in terms of capacity to the HOMER Pro. The difference was among 34% to 68%.

Those discrepancies are not easy to address without some real systems validation. The mathematical models are different and particular parameters can be tuned as well in each technique, and that can justify the difference, which was presented in all the case studies.

As a comparison, consider case study 5: the optimal solution provided by HOMER Pro requires 17% more PV panels than the solution presented by the synthesis tool, and HOMER Pro does not show the arrangement of arrays (i.e. the number of series and parallel PV panels); the battery bank presented by HOMER Pro provides a capacity of 417 Ah (5×83.4 Ah), while the synthesis tool presented an optimal solution with a total capacity of 560 Ah (7×80 Ah) and this represent a difference of 34%.

Comparing the optimization results with those the real-world, the author had four PV systems deployed and monitored since June 2018 in a riverside community in State pf Amazonas, Brazil, with demands similar to those of case studies 1, 4, 5, and 6, always with a 3×325 W (3S) panels and 4×220 Ah (2S-2P = 440 Ah) lead-acid batteries. These solutions are closer to the results from PV panel, controller and inverter presented by the proposed formal synthesis approach than that of HOMER Pro, thereby showing that the solution is sound, which answers *EQ1*. However the solution presented for the battery bank is very different.

HOMER Pro suggests a value in kW for the inverters that is very close to the peak of every case study, and it is just a reference value and not a commercial value of the inverter employed. The proposed synthesis tool, however, presents inverters that are commercial and can be bought off-the-shelf. This is a clear advantage of the formal synthesis method.

As was reported in section 4.5, HOMER Pro does not include charge controllers as an explicit item of equipment in its mathematical model; only the synthesis tool presents a commercial controller and includes it during the cost analysis. The formal synthesis method, therefore, presents more reliable results than HOMER Pro.

Case study 7 was not solved by the synthesis tool within the time limit established during the experimental phase. Case study 2 was solved just by the ESBMC verifier engine. Case study 3 could not be simulated in HOMER Pro, because of its restriction on setting battery autonomy, thus leaving those three cases without parameters to compare.

Summarizing, the synthesis tool is capable of presenting a solution which is far more detailed and closer to commercial conditions than the solution presented by HOMER Pro. In particular, the automated synthesis method can provide all the details of

every component of a PV system solution, with complete electrical details from the manufacturer data sheet, including the model of the component, nominal current and voltage. In this respect, even the name of the manufacturer can be cited (in Table 4.1 it was removed to avoid unauthorized advertising).

4.6 Threats to Validity

In this Section, a favorable assessment was made of the proposed formal synthesis method. Nevertheless, three threats to the validity of the experimental results were also identified, which can be further assessed and constitute future work:

- (1) improvement to the power reliability analysis: inclusion of loss of load probability or loss of power supply probability, in order to increase the accuracy of the analysis;
- (2) the cost analysis is well tailored to Brazilian Amazon; however, a broad analysis of other isolated areas must be performed in order to make the optimization general in terms of applicability;
- (3) deployment in the field of PV systems sized using the synthesized results in order to validate them.

4.7 Conclusion

In this chapter we showed in detail how the state-of-the-art computer science method of automated synthesis has been adapted to use in stand-alone solar PV systems in order to obtain optimal project sizing. Moreover, it was possible to illustrate how the proposed method compares with the traditional use of a simulation tool for the same purpose.

Detailed diagrams, flowcharts and algorithm with pseudo-code were presented as support for the proposed work and to aid understanding. Also listed, were the assumptions underlying the automated verification and simulation software.

Comparison of verifiers with the proposed automated synthesis using model checking was made, in order to obtain the optimal sizing of stand-alone solar PV systems, which showed that ESBMC had the best overall performance. CBMC was not conclusive for all the cases, and CPAchecker was unable to present a conclusive result in cases 2, 3, and 7, (*time out*).

A comparison with a specialized simulation tool was also made, however only 4 case studies (case 1, case 4, case 5, and case 6) compared to the presented result.

Some limitations of the simulation tool, mainly related to not allowing battery autonomy setup, and the $time - out$ or $out\,of\,memory$ messages presented by the verifiers, restricted the comparison between the automated verification tools and the simulation tools. However, when the comparison was possible, it was noticeable that the LCC was not so distant, that HOMER Pro produces a bigger PV panel sizing, and moreover that the synthesis tool produces bigger battery bank.

Based on the fact that mostly case studies were deployed in the field (case 1, case 4, case 5, and case 6), with regular visits to conduct interviews with the residents, and consult the monitoring system in those houses, it was possible to compare the computer assisted results with the real-world employment of stand-alone solar PV systems. The final conclusion was that the proposed tool is sound, with an acceptable performance, and a higher quality output.

Chapter 5

Conclusions

Here we present the conclusions of our research and the results obtained during the Doctoral process. In particular, we described two scientific methods (or techniques) from computer science, which were used to tackle issues in PV systems, such as validation over time and sizing optimization. A list of future contributions concludes the study, although this list might foster the creation of a research group to investigate model checking techniques for electrical systems and not only renewable sources as has been the focus here.

5.1 Main Contributions

We split our contributions into two, each one concerning the scientific methods proposed with the use of model checking for solar PV systems, and the tools created to validate it.

The first contribution was presented in chapter 3, we described and evaluated an automated verification method of validating a given stand-alone PV system. Our method, using model checking, can perform a size check and verify, over time, whether a PV system meets electrical requirements. There are various possible ways to perform that validation, such as testing, laboratory measuring, or even simulation. However, we developed an algorithm and a tool using model checking that is feasible and effective, although our technique does not replace the others. Our results were supported by case studies, by data gathered from the field, and by interviews with PV owners.

We analyzed five case studies of real PV systems deployed in five different sites, ranging from 975 W to $1,300$ W (inverter specification), and the results of three state-of-the-art verification engines (ESBMC, CBMC, and CPAchecker). The verification

method with ESBMC was faster than the simulation method. We validate the overall results by the sized systems deployed in the field.

Moreover, considering that HOMER Pro always performs size optimization during the simulation process, the user always has to "adopt" this simulation tool in order to avoid the optimization and to validate the PV system as inputted in it. That adaptation is not necessary using our approach because optimization is a method separate from verification. Besides, only the proposed method of automated verification was able to validate a system with a specific battery autonomy since HOMER Pro does not have this feature, i.e., the user can not specify battery autonomy.

In the comparison among verification engines, ESBMC with the SMT solver Z3 executed in the "incremental SMT" configuration presented a better performance than CBMC, used less RAM than CBMC and CPAchecker as expected and explained in Section 2.2.3. All our experimental results are sound since the PV owners and the monitoring system validated the possible flaws that the system could present in the field.

Furthermore, we also have contributions from the method described in chapter 4, where we describe and evaluate an automated synthesis method of obtaining the optimal size of a PV system using software model checking techniques. The focus was on the synthesis method of obtaining the optimal solution based on formal methods, which can improve coverage of the design-space more effectively than simulation tools. Our thesis produced methodological research that innovated concerning the first use of automated synthesis for optimal sizing of solar PV systems.

We proposed a variant of the CEGIS synthesis process, working with just one feasible solution from the SYNTHESIZE phase and refinement from the iterative search from the VERIFY phase. Using this approach, we obtain the optimization of stand-alone PV systems, arriving at the best compromise between power reliability and system cost analysis. Our algorithm that implements this method uses a database of commercially available equipment, including the price. Furthermore, in order to validate our method, we conducted seven case studies of PV systems on two different sites in the State of Amazonas, Brazil, ranging from 253 W to a 1,586 W peak; and three state-of-the-art verification engines were considered (ESBMC, CBMC, and CPAchecker). Besides, we again used the specialized off-the-shelf simulation tool (HOMER Pro) to compare the results. In terms of performance and best results, ESBMC was the most successful and used for comparison with the simulation tool.

In summary, our synthesis proposal is capable of presenting a solution that is far more detailed and closer to commercial conditions than the solution presented by HOMER

Pro. In particular, our method can provide all the details of every component of a PV system solution, with complete electrical details from the manufacturer datasheet, including the component model, nominal current, and voltage. We also cover the charge controller, which is unavailable in HOMER Pro. Note that our automated synthesis tool took longer to find the optimal solution than HOMER Pro. However, the solution presented is sound and complete; it also provides a list of equipment that can be bought directly from the manufacturer.

5.2 Areas for Further Research

As the focus here was stand-alone solar PV systems, one promising area for further research will include analysis of other types of renewable energy, including hybrid to allow the method to verify and obtain optimal sizing of typical rural electrification systems. We will also investigate fault localization techniques (ALVES; CORDEIRO; FILHO, 2015; ALVES; CORDEIRO; FILHO, 2017) to automatically find the root cause of errors when our verification method detects a property violation in the design of PV systems.

In the area of automated sizing, we plan to improve the power reliability analysis, to address the restriction to only allow automated synthesis of riverside communities in the State of Amazonas (Brazil), and to validate some cases with deployed PV systems in isolated communities.

We plan to develop the code of a general-purpose simulation tool, like MATLAB, for example, using the same mathematical model employed in our two methods. A broader comparison can thus be made in order to cover all the possibilities of validation or optimization of solar PV systems.

One future long-term direction is to foster the creation of a research group in automated verification and synthesis applied to electrical systems.

5.3 Concluding Remarks

The automated verification and automated synthesis methods demonstrate their effectiveness and potential for use in stand-alone solar PV systems. Although there exist still issues to address in order to improve the tools, notably performance, and the human-computer interaction.

The licensing and use of tools mentioned in this thesis are different for HOMER Pro, which is available only for use with Microsoft Windows, and whose standard annual subscription is set at US$ 504.00 (HOMER, 2017). The verifiers, CPAchecker,

ESBMC, and CBMC, and their solvers, are based on open-source software licenses, which usually allow the users to use, modify, and distribute licensed products freely. Permission is granted, free of charge, to use this software for evaluation and research purposes (which is a great advantage when compared to commercial tools).

The tools we have described here in no way are similar to any other prior work. We may conclude that our research is original and expands the boundaries of knowledge. In particular, we have proposed cutting-edge computer science methodologies to solve typical electrical engineering problems and to improve the design of systems.

Moreover, last but not least, one of the essential aspects of Ph.D. work is that the student-researcher will be able to conduct this line of research in the future. In that direction, we can say that with the methods proposed here, formulated to tackle issues in the energy sector, there exists a vast field of possible future applications and developments, using different renewable sources. We can mention here biomass, wind, or even hybrid generation, with different mathematical models and requirements; and including smart grids, or electrical systems in general. Little research has been done on this until now, but the scenario is auspicious.

Appendix A

List of Publications

Papers written during thesis elaboration with the status of every submission up to the date of this document.

A.1 Journals

Trindade, A., Cordeiro, L. **Automated Formal Verification of Stand-alone Solar Photovoltaic Systems**, submitted: 8 May 2019, status: accepted 26 September 2019; published online 15 October 2019 by Elsevier Solar Energy (Impact Factor 4.674, CAPES A1 Interdisciplinary, CAPES A1 Engineering IV) <https://www.sciencedirect.com/science/article/abs/pii/S0038092X1930965X> (TRINDADE; CORDEIRO, 2019).

Trindade, A., Cordeiro, L. **Optimal Sizing of Stand-alone Solar PV Systems via Automated Formal Synthesis**, submitted: 28 September 2019, status: under review by IEEE Transactions on Sustainable Energy (Impact Factor 7.65, CAPES A1 Engineering IV), <https://www.journals.elsevier.com/applied-energy> (TRINDADE; CORDEIRO, 2019).

Trindade, A., Degelo, R., Santos Junior, E., Ismail, H., Silva, H., and Cordeiro, L. **Multi-core model checking and maximum satisfiability applied to hardware-software partitioning**, Online publication date: 15 November 2017, by International Journal of Embedded Systems (IJES), CAPES B2 Computing Engineering index. DOI: <https://doi.org/10.1504/IJES.2017.10008947> (TRINDADE et al., 2017).

A.2 Conference

Trindade, A., Cordeiro, L. **Automated Formal Synthesis of Optimal Sizing for Stand-Alone Solar Photovoltaic Systems**, submitted: 6 May 2019, status: rejected by The IEEE/ACM Automated Software Engineering (ASE 2019) Conference (New Ideas Papers) in <https://2019.ase-conferences.org/track/ase-2019-papers>. This paper is now the base of a new paper, for the 32nd International Conference on Computer-Aided Verification (CAV) 2020 with deadline on 23 January 2020 in <http://i-cav.org/2020/call-for-papers/>.

Appendix B

Tools Description and Instructions on their Use

Here we describe the two tools created to implement and validate the two scientific methods/techniques of the thesis.

B.1 Tools

During the doctoral process, the techniques from computer science proposed to deal with issues inherent to solar photovoltaic systems became algorithms and programs. These programs became tools, and the tools were used for the purpose of comparison with simulation tools in order to validate the techniques.

One premise of our tools was that they will be written as an ANSI C program, and to follow the requirements established for the International Competition on Software Verification (SV-COMP) [1]. This allows us to write a code that can be executed into three different verifiers without adaptation, with the possibility that in the future, other verifiers can also be used.

With this in mind, the code was written without using of '# include', or '# define'. Moreover, based on the fact that a compiler library is not used, it was necessary to write in some functions which perform specific calculations depending on the mathematical models adopted (e.g. natural logarithms and exponents).

[1] https://sv-comp.sosy-lab.org/

B.1.1 The Automated Verification Tool (for PV System validation)

We split the code into four blocks:

- **Block 1**: global variables declaration, with weather data from the location (minimum and maximum solar irradiance and temperature, for 24 hours of the day) as array, requirements, and datasheet information pertaining to each item of equipment of the PV system to be verified;
- **Block 2**: support functions (e.g. natural logarithms, exponents);
- **Block 3**: PV system specific functions (charge battery, discharge battery, PV panel generation, sizing check);
- **Block 4**: main code with charge, discharge battery control, based on PV generation from solar panels and the battery state of charge, committing to deliver the power and energy that the house requires, in accordance with the restrictions from the sized system.

B.1.2 The Automated Synthesis Tool (for PV System Optimal Sizing)

We split the code into three blocks:

- **Block 1**: global variables declaration, with weather data from the location, user requirements, and datasheet information on each item of equipment of the PV system to be verified;
- **Block 2**: support functions (start value at the lowest cost for the list of equipment, synthesis phase function to obtain feasible technical solutions with model checking);
- **Block 3**: main code with the iterative control of the verify phase.

B.2 Instructions on How to Use it

The only block that requires user intervention to input data is Block 1 listed in B.1.1 and B.1.2 from the automated verification and synthesis tools respectively. If the intention is to change the mathematical model adopted, coding must be done in Block 3 of the automated verification tool Block 2 of the automated synthesis tool.

For the automated verification tool, it is necessary to input the following data (editing and modifying the code):

- Minimum and maximum solar irradiance (average value for the 24 hours of the day) at location;
- Minimum and maximum temperature (average for every month of the year);
- Local insolation (average number of sun hours per day at the location);
- Number of solar panels in series;
- Number of solar panels in parallel;
- Number of series-connected cells from each solar panel;
- Nominal Operating Cell Temperature;
- Reference solar irradiance;
- Reference temperature;
- Short-circuit current temperature coefficient;
- Open-circuit voltage temperature coefficient;
- Reference short-circuit current;
- Reference open-circuit voltage;
- Reference maximum current, voltage, and power;
- Minimum MPPT voltage;
- Operation current;
- Solar panel efficiency;
- Panel area in square meters;
- DC-bus voltage;
- Individual battery voltage;
- Battery bank capacity;
- State of charge limit;
- Number of batteries in series;
- Number of batteries in parallel;
- Battery autonomy;
- Battery efficiency;
- Float, absorption, and bulk battery voltages;

- Charge controller efficiency;
- Charge controller current;
- Charge controller maximum voltage;
- Inverter output voltage;
- Inverter AC reference power;
- Inverter maximum (surge) AC power reference;
- AC voltage (standard voltage from outlet);
- Power demand from the house;
- Surge demand from the house;
- Energy consumption from the house.

in the case of the automated synthesis tool, the following data must be added (editing and modifying the code):

- Minimum and maximum solar irradiance (average value for the 24 hours of the day) at location;
- Minimum and maximum temperature (average for every month of the year);
- Local insolation (average number of sun hours per day at the location);
- DC-bus voltage;
- Battery autonomy;
- State of charge limit;
- Individual battery voltage;
- Reference solar irradiance;
- Reference temperature;
- For every item on the solar PV panel list, as a matrix (area, efficiency, number of series-connected cells, Nominal Operating Cell Temperature, short-circuit current temperature coefficient, open-circuit voltage temperature coefficient, reference short-circuit current, reference open-circuit voltage, reference maximum power, reference maximum current, reference maximum voltage, maximum voltage on NOCT, cost in USD);
- For every item on the battery list, as a matrix (efficiency, voltage, capacity C_{20}), bulk voltage, float voltage, cost in USD);

- For every item on the charge controller list, as a matrix (efficiency, nominal current, voltage output, minimum MPPT voltage, maximum current, cost in USD);
- For every item on the inverter list, as a matrix (efficiency, input DC voltage, output AC voltage, reference AC power, reference maximum AC power, cost in USD);
- AC voltage (standard voltage from outlet);
- Power demand from the house;
- Surge demand from the house;
- Energy consumption from the house.

B.2.1 Automated Verification with Incremental ESBMC

Here we describe precisely how to use the code explained in the previous section (B.1.1) using the incremental ESBMC with the Z3 SMT solver that presented the best performance during the comparison with other verifiers. However, in 3.4 it was shown how to use the code with other verifiers.

ESBMC can be downloaded from its GitHub in <https://github.com/esbmc/esbmc>. Installation commands can be found in <https://ssvlab.github.io/esbmc/documentation.html>. There is only the Linux version of the verifier.

Command-line: $ esbmc filename.c --no-bounds-check --no-pointer-check --unwind 100 --smt-during-symex --smt-symex-guard --z3.

Where:

- -no-bounds-check: says to the verifier to not perform array bound check (the aim is not to check array bound violation);
- --no-pointer-check: says to the verifier to not check if there is pointer violation;
- --unwind 100: limits the bound of search (100 in this case);
- --smt-during-symex: enables incremental SMT solving;
- --smt-symex-guard: calls the solver during symbolic execution;
- --z3: uses Z3 as solver.

Some results issued by the ESBMC verifier were shown in 3.5.1 when conducting the case studies.

B.2.2 Automated Synthesis with CPAchecker

Here we describe precisely how to use the code explained in previous sections (B.1.2) using CPAchecker with MathSAT solver, which presented the best performance during the comparison with other verifiers. However, in 4.4 it was shown how to use the code with other verifiers.

The CPAchecker can be downloaded from its web site in <https://cpachecker.sosy-lab.org/>. There are versions available for Linux and MS-Windows operational systems.

Install a Java Runtime Environment, which is at least Java 8 compatible (e.g., Oracle JRE, OpenJDK). Cf. the web site <http://java.oracle.com/>.

The command-line in Linux is: $ scripts/cpa.sh -heap 64000m -config config/bmc-incremental.properties -spec config/specification/sv-comp-reachability.spc filename.c.

Note that the default heap size is 1200m (or 1.2 GB RAM). However, specifying a significant value with '-heap' is recommended if you have more RAM available. In our case, we adjusted the heap to 64000m (64 GB).

Another important issue is the time-out of the execution process: the parameter 'limits.time.cpu' in the file 'resource-limits.properties' in folder 'config/includes' must be edited according to the desired value. In our case, 'limits.time.cpu = 14400s', i.e. 14400 seconds or 240 minutes or 4 hours.

Some results, reports, and graphics, issued by the CPAchecker verifier were illustrated in 4.5.

Appendix C

Data from Equipment Used During Experimentation

Here we describe the detailed data from each one of the equipment there were used during the validation and/or optimization of Solar PV systems. The aim is to aid to understand where each variable or parameter came from.

According with the evaluated local of the PV systems installation (Manaus, Amazonas State, Brazil):

- The considered insolation for the worst month is $3.8\,kWh/m^2$ per day[1];
- The minimum average solar irradiance in W/m^2 per hour (in a day) is: 0, 0, 0, 0, 0, 0, 25, 135, 274, 422, 509, 537, 503, 505, 430, 281, 80, 10, 0, 0, 0, 0, 0, 0;
- The maximum average solar irradiance in W/m^2 per hour (in a day) is 0, 0, 0, 0, 0, 4, 87, 295, 487, 648, 751, 852, 817, 742, 610, 418, 128, 51, 0, 0, 0, 0, 0, 0;
- The average minimum temperature (in ^{o}C) per month is: 23, 23, 23, 23, 23, 23, 23, 23, 23, 24, 24, 23;
- The average maximum temperature (in ^{o}C) per month is: 30, 30, 30, 30, 30, 30, 30, 31, 32, 32, 31, 30;

[1] <http://www.cresesb.cepel.br/index.php?section=sundata>

Table C.1: PV Panels

η_p	N	T	μ_I	μ_V	$I_{sc,ref}$	$V_{oc,ref}$	P_m	I_m	V_m	V_{mp}	US$	Model
0.1620	72	45	0.053	-0.31	9.18	45.1	315	8.16	36.6	33.4	268.40	CS6U-315
0.1646	72	45	0.053	-0.31	9.26	45.3	320	8.69	36.8	33.6	190.00	CS6U-320
0.1672	72	45	0.053	-0.31	9.34	45.5	325	8.78	37.0	33.7	216.67	CS6U-325
0.1697	72	45	0.053	-0.31	9.45	45.6	330	8.88	37.2	33.9	170.30	CS6U-330
0.1515	80	47	0.020	-0.48	8.86	50.8	340	8.26	41.2	37.0	214.20	KU340-8BCA
0.1600	54	47	0.00318	-0.123	8.21	32.9	200	7.61	32.9	23.2	300.00	KC200GT
0.1690	60	47	0.039	-0.307	9.44	38.84	275	8.81	31.22	26.72	150.00	SA275-60P
0.1855	72	45	0.039	-0.307	9.73	47.60	360	9.33	38.59	34.96	237.24	SA360-72M
0.1515	36	45	0.033	-0.39	8.81	22.30	150	8.20	18.30	14.40	94.75	RSM36-6-150P
0.1500	36	46	0.060	-0.37	8.61	22.90	150	8.12	18.50	14.61	108.50	YL150P-17b

Caption: (T): NOCT, (CS): Canadian, (KC or KU): Kyocera, (SA): Sinosola, (RMS): Risen, (YL): Yingli.

Table C.2: Batteries

η_b	V_b	C_{20}	V_{bulk}	V_{float}	$Cost\,(US\$)$	Brand	Model
0.85	12	80	14.4	13.8	131.00	Moura	12MF80
0.85	12	105	14.4	13.8	150.00	Moura	12MF105
0.85	12	150	14.4	13.8	324.75	Moura	12MF150
0.85	12	175	14.4	13.8	299.75	Moura	12MF175
0.85	12	220	14.4	13.8	374.75	Moura	12MF220
0.85	12	60	14.4	13.8	114.75	Heliar	DF1000
0.85	12	80	14.4	13.2	138.00	Heliar	DF1500
0.85	12	150	14.4	13.2	275.00	Heliar	DF2500
0.85	12	170	14.4	13.2	299.01	Heliar	DF3000
0.85	12	220	14.4	13.2	330.73	Heliar	DF4001

Table C.3: Charge controllers

η_c	I_c	V_{OUT}	$V_{c,max}$	$Cost\,(US\$)$	Brand	Model
0.98	35	24	145	294.95	Victron	35-145
0.98	15	24	75	88.40	Victron	15-75
0.98	15	24	100	137.70	Victron	15-100
0.98	50	24	100	294.95	Victron	50-100
0.98	20	24	100	132.25	Epever	TRIRON 2210N 20A 12/24V
0.98	30	24	100	161.00	Epever	TRIRON 3210N 30A 12/24V
0.98	40	24	100	184.75	Epever	TRIRON 4210N 40A 12/24V
0.97	20	24	100	217.25	Epsolar	Tracer-2210RN 20A 12/24V
0.97	30	24	100	297.25	Epsolar	Tracer-3215BN 30A 12/24V
0.96	60	60	140	1072.50	SE	XW-MPPT 60/150
0.98	60	48	150	388.91	Epever	ET6415BND
0.98	50	48	150	347.82	Epever	54150AN

Caption: (SE): Schneider Electric.

Table C.4: Inverters

η_i	$V_{in}DC$	$V_{out}AC$	$P_{AC,ref}$	$MAX_{AC,ref}$	$Cost(US\$)$	Brand	Model
0.93	48	110	700	1600	400.00	Victron	24-800
0.93	48	110	1200	2400	750.00	Victron	48-1200
0.93	24	120	1200	2400	450.00	Epever	IP1500-11
0.91	24	120	280	750	149.75	Epever	IP350-11
0.91	24	120	400	1000	187.25	Epever	IP500-11
0.93	12	220	600	1350	342.25	Epsolar	SHI600-12
0.93	24	120	800	1200	500.00	Epsolar	STI1000-24-120
0.90	12	120	900	2000	649.75	Xantrex	SW 1000
0.82	12	120	1000	2000	1122.25	Xantrex	HFS 1055 1000W
0.90	24	125	1800	2900	1669.75	Xantrex	HF 1800W

References

CLARKE, E. 25 years of model checking. In: _____. : Springer, 2008. cap. The birth of model checking, p. 1–26.

ROY, T. *Simulation and Analysis of Photovoltaic Stand- Alone Systems*. Dissertação (Technology in Electrical Engineering) — Department of Electrical Engineering National Institute of Technology, Rourkela, Odisha, India, 2013.

HANSEN, A.; S‰RENSEN, P.; HANSEN, L.; BINDNER, H. *Models for a stand-alone PV system*. : Forskningscenter Risoe, 2001. (Denmark. Forskningscenter Risoe. Risoe-r, 1219).

SAMLEXSOLAR.COM. *Solar (PV) Cell, Module, Array*. 2017. <http://www.samlexsolar.com/learning-center/solar-cell-module-array.aspx>. [Accessed 28 December 2017].

CUBAS, J.; PINDADO, S.; SORRIBES-PALMER, F. Analytical calculation of photovoltaic systems maximum power point (MPP) based on the operation point. *Applied Sciences*, v. 7, n. 9, p. 870–884, 2017.

SALOUX, E.; TEYSSEDOU, A.; SORIN, M. Explicit model of photovoltaic panels to determine voltages and currents at the maximum power point. *Solar Energy*, v. 85, n. 5, p. 713–722, 2011.

PINHO, J.; GALDINO, M. *Manual de Engenharia para Sistemas Fotovoltaicos*. Rio de Janeiro/RJ: CEPEL – CRESESB, 2014.

UC. *Cyber-Physical Systems*. 2016. <http://cyberphysicalsystems.org/>. [Accessed 1st October 2016].

NSF National Science Foundation. *Cyber Physical Systems, NSF 15-541*. 2015. <http://www.nsf.gov/pubs/2015/nsf15541/nsf15541.pdf>. [Acessed 24 April 2019].

CHAVES, L. C.; ISMAIL, H. I.; BESSA, I. V. de; CORDEIRO, L. C.; FILHO, E. B. de L. Verifying fragility in digital systems with uncertainties using dsverifier $v2.0$. *Journal of Systems and Software*, v. 153, p. 22–43, 2019.

CORDEIRO, L. C.; FILHO, E. B. de L.; BESSA, I. V. Survey on automated symbolic verification and its application for synthesising cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications*, Institution of Engineering and Technology, August 2019.

METZGER, A.; POHL, K. Software product line engineering and variability management: Achievements and challenges. In: *FOSE*. 2014. p. 70–84.

FILIERI, A.; MAGGIO, M.; ANGELOPOULOS, K.; D'IPPOLITO, N.; GEROSTATHOPOULOS, I.; HEMPEL, A.; HOFFMANN, H.; JAMSHIDI, P.; KALYVIANAKI, E.; KLEIN, C.; KRIKAVA, F.; MISAILOVIC, S.; PAPADOPOULOS, A. V.; RAY, S.; SHARIFLOO, A. M.; SHEVTSOV, S.; UJMA, M.; VOGEL, T. Software engineering meets control theory. In: *SEAMS '15*. 2015. p. 71–82.

AL-JAROODI, J.; MOHAMED, N.; JAWHAR, I.; LAZAROVA-MOLNAR, S. Software engineering issues for cyber-physical systems. In: *SMARTCOMP*. 2016. p. 1–6.

SOFTWARE Networked European; NESSI Services I. *Software Engineering: Key Enabler for Innovation*. 2014. <http://www.nessi-europe.eu/Files/Private/NESSI_SE_WhitePaper-FINAL.pdf>. [Acessed 24 April 2019].

HUSSEIN, M.; LEAL FILHO, W. Analysis of energy as a precondition for improvement of living conditions and poverty reduction in sub-Saharan Africa. In: *Scientific Research and Essays*. 2012. v. 7(30), p. 2656–2666.

IEA. *World Energy Outlook 2018*. Paris: IEA, 2018.

EPE Empresa de Pesquisa Energética. *Sistemas Isolados - Planejamento Ciclo 2018 - 2023*. 2018. <http://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes>. [Acessed 04 April 2019].

COELHO, S.; SANCHES-PEREIRA, A.; TUDESCHINI, L.; ESCOBAR, J.; POVEDA, M.; COLUNA, N.; COLLIN, A.; ROVERE, E. L.; TRINDADE, A.; PEREIRA, O. Biomass residues as electricity generation source in low HD source in regions of Brazil. In: UNESP (Ed.). *The XI Latin. Cong. of Elec. Gener. and Transm. CLAGTEE*. 2015. p. 1–8.

KAREKESI, S.; LATA, K.; COELHO, S. Renewable energy - a global review of technologies, policies and markets. In: _____. : London: Earthscan, 2006. cap. Traditional Biomass Energy: Improving Its Use and Moving to Modern Energy Use, p. 231–261.

YATIMI, H.; AROUDAM, E. Modeling and simulation of a stand-alone photovoltaic system. In: MAROCCO, T. (Ed.). *Xth Int. Conf. on Int. Des. and Prod.* 2015.

NOROOZIAN, R.; GHAREHPETIAN, G. An investigation on combined operation of active power filter with photovoltaic arrays. *Int. J. of Elec. Power & Energ. Syst.*, v. 46, p. 392–399, 2013.

SEIA. *Solar Energy*. 2016. <http://www.seia.org/about/solar-energy>. [Accessed 1st October 2016].

CHAUHAN, A.; SAINI, R. Renewable energy based off-grid rural electrification in uttarakhand state of india: Technology options, modelling method, barriers and recommendations. *Ren. and Sust. Energ. Rev.*, v. 51, p. 662–681, 2015.

EPIA. *Global market outlook for photovoltaics 2017-2021*. Belgium: European Phototoltaic Industry Association, 2017.

RAWAT, R.; KAUSHIK, S.; LAMBA, R. A review on modeling, design methodology and size otimization of photovoltaic based motor pumping, standalone and grid connectec system. *Renew. and Sust. Energ. Rev.*, v. 57, p. 1506–1519, 2016.

PRADHAN, S.; SINGH, S.; CHOUDHURY, M.; DWIVEDY, D. Study of cost analysis and emission analysis for grid connected PV systems using RETSCREEN 4 simulation software. *Int. J. of Eng. Res. & Tech.*, v. 4, n. 4, p. 203–207, 2015.

SWARNKAR, N.; GIDWANI, L.; SHARMA, R. An application of HOMER Pro in optimization of hybrid energy system for electrification of technical institute. In: *Int. Conf. on Energ. Eff. Tech. for Sust. (ICEETS)*. 2016. p. 56–61.

DOBOS, A. *PVWatts Version 5 Manual*. Colorado, 2014.

BLAIR, N.; DOBOS, A.; FREEMAN, J.; NEISES, T.; WAGNER, M. *System Advisor Model, SAM 2014.1.14: General Description*. Colorado, 2014.

MILLS, A.; AL-HALLAJ, S. Simulation of hydrogen-based hybrid systems using Hybrid2. *Int. J. of Hydrog. Energy*, v. 29, n. 10, p. 991–999, 2004.

GOW, J.; MANNING, C. Development of a photovoltaic array model for use in power-electronics simulation studies. In: *Proceedings of the 14th IEE Electric Power Applications Conference*. 1999. v. 146(2), p. 193–200.

BENATIALLAH, A.; BENATIALLAH, D.; GHAITAOUI, T.; HARROUZ, A.; MANSOURI, S. Modelling and simulation of renewable energy systems in Algeria. *Int. J. of Sc. and App. Inf. Tech*, v. 7, n. 1, p. 17–22, 2017.

BROOKS, W.; DUNLOP, J. *NABCEP: PV Installation Professional Resource Guide*. Clifton Park, NY: NABCEP, 2013.

ZHOU, W.; LOU, C.; LI, Z.; LU, L.; YANG, H. Current status of research on optimum sizing of stand-alone hybrid solar-wind power generation systems. *Applied Energy*, v. 87, n. 2, p. 380–389, 2010.

CLARKE, E. M.; HENZINGER, T. A.; VEITH, H. Introduction to model checking. In: *Handbook of Model Checking.* : Springer, 2018. p. 1–26.

ABATE, A.; BESSA, I.; CATTARUZZA, D.; CORDEIRO, L.; DAVID, C.; KESSELI, P.; KROENING, D.; POLGREEN, E. Automated formal synthesis of digital controllers for state-space physical plants. In: *Comp. Aided Verif. (CAV)*. 2017. LNCS 10426, p. 462–482.

ABATE, A.; BESSA, I.; CATTARUZZA, D.; CORDEIRO, L. C.; DAVID, C.; KESSELI, P.; KROENING, D. Sound and automated synthesis of digital stabilizing controllers for continuous plants. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC 2017, Pittsburgh, PA, USA, April 18-20, 2017*. 2017. p. 197–206.

BESSA, I.; ISMAIL, H.; PALHARES, R.; CORDEIRO, L.; FILHO, J. C. Formal nonfragile stability verification of digital control systems with uncertainty. *In IEEE Trans. on Comp.*, v. 66, n. 3, p. 545–552, 2017.

CHAVES, L. C.; BESSA, I.; CORDEIRO, L. C.; KROENING, D.; FILHO, E. B. de L. Verifying digital systems with MATLAB. In: *Proceedings of the 26th ACM*

SIGSOFT International Symposium on Software Testing and Analysis, Santa Barbara, CA, USA, July 10 - 14, 2017. 2017. p. 388–391.

ABATE, A.; BESSA, I.; CATTARUZZA, D.; CHAVES, L. C.; CORDEIRO, L. C.; DAVID, C.; KESSELI, P.; KROENING, D.; POLGREEN, E. Dssynth: an automated digital controller synthesis tool for physical plants. In: *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017, Urbana, IL, USA, October 30 - November 03, 2017*. 2017. p. 919–924.

GADELHA, M. Y. R.; ISMAIL, H. I.; CORDEIRO, L. C. Handling loops in bounded model checking of C programs via k-induction. *STTT*, v. 19, n. 1, p. 97–114, 2017.

TRINDADE, A.; CORDEIRO, L. Applying SMT-based verification to hardware/software partitioning in embedded systems. *Des. Aut. for Emb. Syst. (DAES)*, v. 20, n. 1, p. 1–19, 2016.

TRINDADE, A. B.; DEGELO, R. D. F.; JUNIOR, E. G. D. S.; ISMAIL, H. I.; SILVA, H. C. D.; CORDEIRO, L. C. Multi-core model checking and maximum satisfiability applied to hardware-software partitioning. *IJES*, v. 9, n. 6, p. 570–582, 2017.

CLARKE, E.; EMERSON, E.; SIFAKIS, J. Model checking: algorithmic verification and debugging. *Commu. of the ACM*, v. 51, n. 11, p. 74–84, 2009.

GADELHA, M.; MONTEIRO, F.; MORSE, J.; CORDEIRO, L.; FISCHER, B.; NICOLE, D. ESBMC 5.0: An industrial-strength C model checker. In: 33^{rd} *ACM/IEEE Int. Conf. on Aut. Soft. Engin. (ASE'18)*. New York, NY, USA: ACM, 2018. p. 888–891.

RAMALHO, M.; FREITAS, M.; SOUSA, F. R. M.; MARQUES, H.; CORDEIRO, L. C.; FISCHER, B. Smt-based bounded model checking of C++ programs. In: *20th IEEE International Conference and Workshops on Engineering of Computer Based Systems, ECBS 2013, Scottsdale, AZ, USA, April 22-24, 2013*. 2013. p. 147–156.

CORDEIRO, L. C.; KESSELI, P.; KROENING, D.; SCHRAMMEL, P.; TRTÍK, M. JBMC: A bounded model checking tool for verifying java bytecode. In: *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I.* : Springer, 2018. (Lecture Notes in Computer Science, v. 10981), p. 183–190.

APPLASAMY, V. Cost evaluation of a stand-alone residential photovoltaic power system in malaysia. *2011 IEEE Symposium on Business, Engineering and Industrial Applications (ISBEIA)*, p. 214–218, 2011.

ALSADI S.; KHATIB, T. Photovoltaic power systems optimization research status: A review of criteria, constrains, models, techniques, and software tools. *Appl. Sci.*, v. 8, n. 1761, p. 1–30, 2018.

AKRAM, W.; NIAZI, M. A. A formal specification framework for smart grid components. *Complex Adaptive Systems Modeling*, v. 6, n. 1, p. 5, Sep 2018.

KROENING, D.; TAUTSCHNIG, M. CBMC – C Bounded Model Checker (competition contribution). In: *Tools and Alg. for the Const. and An. of Sys. (TACAS)*. 2014. LNCS 8413, p. 389–391.

YÜKSEL, E.; ZHU, H.; NIELSON, H. R.; HUANG, H.; NIELSON, F. Modelling and analysis of smart grid: A stochastic model checking case study. In: *Sixth International Symposium on Theoretical Aspects of Software Engineering*. 2012. p. 25–32.

KWIATKOWSKA, M. Z.; NORMAN, G.; PARKER, D. PRISM 4.0: Verification of probabilistic real-time systems. In: *CAV*. 2011. (LNCS, v. 6806), p. 585–591.

SENGUPTA, A.; MUKHOPADHYAY, S.; SINHA, A. Automated verification of power system protection schemes—Part I: Modeling and specifications. In: *IEEE Tran. on Power Del.* 2015. v. 30(5), p. 2077–2086.

CHENG, B.; WANG, X.; LIU, J.; DU, D. Modana: An integrated framework for modeling and analysis of energy-aware CPSs. In: *IEEE 39th Annual Computer Software and Applications Conference*. 2015. p. 127–136.

ABATE, A. Verification of networks of smart energy systems over the cloud. In: BOGOMOLOV, S.; MARTEL, M.; PRABHAKAR, P. (Ed.). *Num. Soft. Verif.* 2017. LNCS 10152, p. 1–14.

DRIOUICH, Y.; PARENTE, M.; TRONCI, E. A methodology for a complete simulation of cyber-physical energy systems. In: *IEEE Work. on Envir., Energ., and Struc. Monit. Syst. (EESMS)*. 2018. p. 1–5.

COLLINS, M. *Formal Methods*. 1998. <https://users.ece.cmu.edu/~koopman/des_s99/formal_methods/>. [Acessed 3 July 2019].

BOWEN, J.; STAVRIDOU, V. Safety-critical systems, formal methods and standards. *Software Engineering Journal*, v. 8, n. 4, p. 189–209, July 1993. ISSN 0268-6961.

BENTLEY, J. *Programming Pearls (2Nd Ed.)*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000. ISBN 0-201-65788-0.

LOWRY, M.; DVORAK, D. Analytic verification of flight software. *IEEE Intelligent Systems and their Applications*, v. 13, n. 5, p. 45–49, Sep. 1998. ISSN 1094-7167.

BOWEN, J. P.; HINCHEY, M. G. Seven more myths of formal methods. *IEEE Software*, v. 12, n. 4, p. 34–41, July 1995. ISSN 0740-7459.

KLING, R. Computerization and controversy (2nd ed.). In: KLING, R. (Ed.). Orlando, FL, USA: Academic Press, Inc., 1996. cap. Systems Safety, Normal Accidents, and Social Vulnerability, p. 746–763. ISBN 0-12-415040-3.

FOREJT, F.; KWIATKOWSKA, M.; NORMAN, G.; PARKER, D. Automated verification techniques for probabilistic systems. In: _____. *Formal Methods for Eternal Networked Software Systems: 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems, SFM 2011.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 53–113.

GADELHA, M. Y. R.; MONTEIRO, F. R.; CORDEIRO, L. C.; NICOLE, D. A. ESBMC v6.0: Verifying C programs using k-induction and invariant inference - (competition contribution). In: *Tools and Algorithms for the Construction and Analysis of Systems - 25 Years of TACAS: TOOLympics, Held as Part of ETAPS*

2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part III. : Springer, 2019. (Lecture Notes in Computer Science, v. 11429), p. 209–213.

MONTEIRO, F. R.; GARCIA, M.; CORDEIRO, L. C.; FILHO, E. B. de L. Bounded model checking of C++ programs based on the qt cross-platform framework. *Softw. Test., Verif. Reliab.*, v. 27, n. 3, 2017.

PEREIRA, P. A.; ALBUQUERQUE, H. F.; SILVA, I. da; MARQUES, H.; MONTEIRO, F. R.; FERREIRA, R.; CORDEIRO, L. C. Smt-based context-bounded model checking for CUDA programs. *Concurrency and Computation: Practice and Experience*, v. 29, n. 22, 2017.

MORSE, J.; CORDEIRO, L. C.; NICOLE, D. A.; FISCHER, B. Model checking LTL properties over ANSI-C programs with bounded traces. *Software and System Modeling*, v. 14, n. 1, p. 65–81, 2015.

MORSE, J. *Expressive and efficient bounded model checking of concurrent software.* Tese (Doutorado) — University of Southampton, 2015.

SCHRAMMEL, P.; KROENING, D.; BRAIN, M.; MARTINS, R.; TEIGE, T.; BIENMÜLLER, T. Incremental bounded model checking for embedded software. *Formal Asp. Comput.*, v. 29, n. 5, p. 911–931, 2017.

MOURA, L. D.; BJØRNER, N. Z3: An Efficient SMT Solver. In: *Tools and Alg. for the Const. and An. of Sys. (TACAS)*. 2008. LNCS 4963, p. 337–340.

BEYER, D.; KEREMOGLU, M. E. CPAchecker: A tool for configurable software verification. In: GOPALAKRISHNAN, G.; QADEER, S. (Ed.). *Lecture Notes in Computer Science*. : Springer, Berlin, Heidelberg, 2011. LNCS 6806, p. 184–190.

BORNHOLT, J. *Program Synthesis in 2019*. 2019. <https://blog.sigplan.org/2019/07/31/program-synthesis-in-2019/>. [Acessed 28 August 2019].

ALUR, R.; BODIK, R.; JUNIWAL, G.; MARTIN, M.; RAGHOTHAMAN, M.; SESHIA, S.; SINGH, R.; SOLAR-LEZAMA, A.; TORLAK, E.; UDUPA, A. Syntax-guided synthesis. In: *2013 Form. Meth. in Comp. Aid. Des.* 2013. p. 1–17.

GULES, R.; PACHECO, J.; HEY, H.; IMHOFF, J. A maximum power point tracking system with parallel connection for pv stand alone applications. *IEEE Tran. on Indust. Elect.*, v. 55, n. 7, p. 2674–2683, 2008.

MOHANTY, P.; SHARMA, K.; GUJAR, M.; KOLHE, M.; AZMI, A. Solar photovoltaic system applications. In: _____. : Springer International Publishing, 2016. cap. PV System Design for Off-Grid Applications, p. 49–83.

KIM, S. Sliding mode controller for the single-phase grid connected photovoltaic system. *Applied Energy*, v. 83, n. 10, p. 1101–1115, 2006.

FREEMAN, J.; WHITMORE, J.; BLAIR, N.; DOBOS, A. *Validation of Multiple Tools for Flat Plate Photovoltaic Modeling Against Measured Data.* Colorado, 2014.

CAMERON, C.; BOYSON, W.; RILEY, D. Comparison of PV system performance-model predictions with measured PV system performance. In: *33rd IEEE Photov. Spec. Conf.* 2008. p. 1–6.

HOMER. *The HOMER Microgrid software*. 2017. <http://www.homerenergy.com/software.html>. [Accessed 1st June 2019].

UMASS, U. of M. A. *Hybrid2*. 2016. <http://www.umass.edu/windenergy/research/topics/tools/software/hybrid2>. [Accessed 1st October 2016].

LORENZO, E. *Solar Electricity Engineering of Photovoltaic Systems*. Spain: Artes Gráficas Gala, S.L., 1994.

JAKHRANI, A.; SAMO, S.; KAMBOH, S.; LABADIN, J.; RIGIT, A. An improved mathematical model for computing power output of solar photovoltaic modules. *Int. J. of Phot.*, v. 2014, n. ID 346704, p. 9, 2014.

DURISCH, W.; TILLE, D.; WORZ, A.; PLAPP, W. Characterisation of photovoltaic generators. *Applied Energy*, v. 65, n. 1-4, p. 273–284, 2000.

RAJANNA, S.; SAINI, R. Modeling of integrated renewable energy system for electrification of a remote area in India. *Renew. Energ.*, v. 90, n. C, p. 175–187, 2016.

BADEJANI, M.; MASOUM, M.; KALANTA, M. Power engineering conference. In: UNIVERSITIES, A. (Ed.). *AUPEC 2007*. 2007. p. 1–6.

FERRARI, S.; PIURI, M. L. V.; SALMAN, A.; CRISTALDI, L.; FAIFER, M.; TOSCANI, S. Solar panel modelling through computational intelligence techniques. *Meas.: J. of the Int. Meas. Conf.*, n. 93, p. 572–580, 2016.

HASAN, M.; PARIDA, S. An overview of solar photovoltaic panel modeling based on analytical and experimental viewpoint. *Renew. and Sust. Energ. Rev.*, v. 60, p. 75–83, 2016.

KING, D.; BOYSON, W.; KRATOCHVILL, J. *Photovoltaic Array Performance Model*. Albuquerque, New Mexico: Sandia National Laboratories, 2004.

MELLIT, A.; BENGHANEM, M.; KALOGIROU, S. Modeling and simulation of a stand-alone photovoltaic system using an adaptive artificial neural network: Proposition for a new sizing procedure. *Renew. Energ.*, v. 32, n. 2, p. 285–313, 2007.

BRANO, V.; ORIOLI, A.; CIULLA, G.; GANGI, A. Modeling , design and simulation of stand-alone photovoltaic power systems with battery storage. *Solar Energ. Mat. & Solar Cells*, v. 94, p. 1358–1370, 2010.

SHENAWY, E.; ESMAIL, O.; ELBASET, A.; HAMED, H. Practical identification of photovoltaic module parameters. *ISESCO J. of Sci. and Tech.*, v. 11, n. 19, p. 66–71, 2015.

TIAN, H.; MANCILLA–DAVID, F.; ELLIS, K.; MULJADIC, E.; JENKINS, J. *A Detailed Performance Model for Photovoltaic Systems*. Colorado, 2012.

VILLALVA, M.; GAZOLI, J.; FILHO, E. Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Trans. on Power Elec.*, v. 24, p. 1198–1208, 2009.

ROSS, R. Flat-plate photovoltaic array design optimization. In: DIEGO, C. S. (Ed.). *14th IEEE Photovoltaic Specialists Conference*. 1980. p. 1126–1132.

COPETTI, J.; LORENZO, E.; CHENLO, F. A general battery model for PV system simulation. *Prog. in Photovoltaics: Res. and App.*, v. 1, n. 4, p. 283–292, 1993.

MANWELL, J.; MCGOWAN, J. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, v. 50, n. 5, p. 399–405, 1993.

MANWELL, J.; MCGOWAN, J. Extension of the kinetic battery model for wind/hybrid power systems. In: *5th Eur. Wind Energy Assoc. Conf.* 1994. p. 284–289.

WENHAM, S.; GREEN, M.; WATT, M. *Applied Photovoltaics*. Australia: Center for Photovoltaic Devices and Systems, 1994.

ABDULATEEF, J. Simulation of solar off- grid photovoltaic system for residential unit. *Int. J. of Sust. and Green Energy*, v. 4, n. 3-1, p. 29–33, 2014.

MAHANTA, P.; DEBNATH, K.; RAHMAN, M. Modeling and simulation of a pv module based power system using matlab/simulink. *Dhaka Univ. J. of Science*, v. 62, n. 2, p. 127–132, 2014.

DHANOWA, A.; GARG, V. Modeling and simulation of an off grid pv system for with battery backup for remote and rural area network. *Int. J. of Eng. Research & Technology (IJERT)*, v. 4, n. 06, p. 915–918, 2015.

CATHERINE, D.; BHASKAR, K. Simulation of a solar mppt charger using cuk converter for standalone application. *Int. J. of Elec., Electron. and Comp. Syst. (IJEECS)*, v. 1, n. 1, 2013.

HAQUE, A. Maximum power point tracking (MPPT) scheme for solar photovoltaic system. *Energ. Technology & Policy*, v. 1, n. 1, p. 115–122, 2014.

KHATIB, T.; ELMENREICH, W. Optimum availability of standalone photovoltaic power systems for remote housing electrification. *Int. Journal of Photoenergy*, n. Article ID 475080, p. 5 pages, 2014.

PARK, C.; KUMAR, P.; KUMAR, N. *Fundamentals of Engineering Economics*. : Prentice Hall, 2004.

KAMEL, S.; DAHL, C. The economics of hybrid power systems for sustainable desert agriculture in Egypt. *Energy*, v. 30, n. 8, p. 1271 – 1281, 2005. ISSN 0360-5442. Dubrovnik Conference on Sustainable Development of Energy, Water and Environment Systems.

ISO. *ISO/IEC 9899:2018 Information Technology – Programming Languages – C*. 2018. <https://www.iso.org/standard/74528.html>. [Acessed 14 November 2018].

WEATHERBASE. *Manaus, Amazonas Travel Weather Averages*. 2018. <https://www.weatherbase.com/weather/weather.php3?s=23328&cityname=Manaus%2C+Amazonas%2C+Brazil>. [Acessed 09 July 2018].

EnergyPlus. *Weather Data by Location*. 2018. <https://energyplus.net/weather-location/south_america_wmo_region_3/BRA//BRA_AM_Manaus-Gomez.Intl.AP.817300_INMET>. [Acessed 09 July 2018].

TRINDADE, A. B.; CORDEIRO, L. C. Automated formal verification of stand-alone solar photovoltaic systems. *Solar Energy*, v. 193, n. 1, p. 684–691, 2019.

TRINDADE, A. Ferramenta de análise comparativa de projetos de eletrificação rural com fontes renováveis de energia na amazônia. In: *IX Congresso sobre Geração Distribuída e Energia no Meio Rural - AGRENER GD*. 2013. p. n.pag.

BRUMMAYER, R.; BIERE, A. Boolector: An efficient SMT solver for bit-vectors and arrays. In: *Tools and Alg. for the Const. and An. of Sys. (TACAS)*. 2009. LNCS 5505, p. 174–177.

CIMATTI, A.; GRIGGIO, A.; SCHAAFSMA, B.; SEBASTIANI, R. The MathSAT5 SMT Solver. In: PITERMAN, N.; SMOLKA, S. (Ed.). *Proceedings of TACAS*. : Springer, 2013. (LNCS, v. 7795), p. 93–107.

SOLAR-LEZAMA, A. Program sketching. *Int. J. on Soft. Tools for Tech. Transf.*, v. 15, n. 5, p. 475–495, 2013.

ALVES, E. H. da S.; CORDEIRO, L. C.; FILHO, E. B. de L. Fault localization in multi-threaded C programs using bounded model checking. In: *2015 Brazilian Symposium on Computing Systems Engineering, SBESC 2015, Foz do Iguacu, Brazil, November 3-6, 2015*. : IEEE Computer Society, 2015. p. 96–101.

ALVES, E. H. da S.; CORDEIRO, L. C.; FILHO, E. B. de L. A method to localize faults in concurrent C programs. *Journal of Systems and Software*, v. 132, p. 336–352, 2017.

TRINDADE, A.; CORDEIRO, L. C. *Optimal Sizing of Stand-alone Solar PV Systems via Automated Formal Synthesis*. 2019. <http://arxiv.org/abs/1909.13139>.