

# Building a framework to perform automated verification of solar photovoltaic off-grid systems

Alessandro Trindade and Lucas Cordeiro

**Abstract**—The deployment of renewable energy in off-grid systems is growing in both developed and developing countries. With declining costs and increasing performance, off-grid renewable energy systems could become an important growth market for the future deployment of renewables according to International Renewable Energy Agency. The impact of off-grid renewable energy systems will not only be measured in terms of their usage or reduced costs for electricity consumption in rural areas, but also in the context of their effect on the lives of more than 1.3 billion people who today do not have access to electricity. Electrification of rural areas can improve the standard of education, health conditions, and living of this population. In addition, depending on the geography of the country, using renewable energy sources in off-grid mode is a feasible solution if compared to grid extension. Tools to aid the design and validation of rural electrification projects are available; however, there is no software that applies formal verification to analyze, and evaluate rural electrification projects. This paper aims to build the framework to apply automated verification methods to mathematical models of a complete system of rural electrification with solar photovoltaic energy to validate its design from a formal verification perspective.

**Index Terms**—Automated verification, model checking, energy systems, solar photovoltaic systems, off-grid systems.

## I. INTRODUCTION

ACCORDING to [1] there are presently 1.3 billion people with no access to electricity worldwide. The definition of energy access usually includes both electricity access and access to modern fuels for cooking and heating, in order to replace traditional biomass. Increased access to energy allows economic growth and poverty alleviation [2].

Among the options of renewable sources, there are hydro, wind, and solar photovoltaic (PV). Only a niche market a few years ago, PV are now becoming a mainstream electricity provider, changing the way the world is powered. Based in data from [3], there was an increase of approximately 30% from 2013 to 2014 in terms of new installations of PV. The use of the power generation technology with renewable energy source is developing rapidly due to the industrial development [4]. The renewable energy leads to an advance all over the world by protecting the environment: it is clean (low greenhouse gases emissions), operating silently, long lifetime, low maintenance, absence of fuel cost and inexhaustible [5]. Renewable energy, and particularly the power generation from solar energy using photovoltaic (PV) panels, has emerged as an alternative to fossil or nuclear fuel generation. The increase in a number

of PV systems installed all over the world brings the need for proper modeling and simulation tools for researcher and practitioners involved in their application.

In order to validate or even simulate a potential PV solution, there are a myriad of tools. However, to the best of our knowledge, this work can be the first work to perform automated verification of a solar PV off-grid project solution.

According to [6] and [7], solar is the most abundant source of renewable energy on earth. In SPV system, solar radiations are captured from the sun and turned into electricity using SPV cells made of silicon and other materials. SPV systems can be classified into grid-connected and stand-alone systems. At this proposal, only off-grid or stand-alone systems will be considered. The utilization of solar energy in off-grid mode has the potential to meet the energy need for remote rural areas of developing countries.

The PV cell in a solar photovoltaic system, as defined in [8], is a semiconductor device, which directly converts the solar radiation into electrical energy. Apart from PV modules; the PV systems consist of battery bank, controller, and inverter, plus the load. Moreover, the optimum sizing of these devices is important for reliable operation. Therefore, these systems need to be designed properly according to the site, land area available, load requirement, load pattern, environmental conditions and economics in order to utilize available resources efficiently and economically [8].

In order to address different aspects of a SPV project, there is a myriad of public domain and commercial software available. According to [9], the capabilities of these tools range from simple solar resource and energy production estimative, to site survey and system design tools, to complex financial analysis software (with optimization). Some tools also provide support to rebate programs applications and tax incentives (specific to each country or region), while other programs and worksheets focus on the technical aspects of system sizing and design.

Manufactures and integrators have yet their proprietary software to perform various system sizing, with the drawback of include just their own products among the possibilities of choice, what restrict the solution.

Changing the subject to the design of complex systems, more time is spent on verification than on construction, as shown in [10]. Formal methods based on model checking offer great potential to obtain a more effective and faster verification in the design process. Programs, and more generally computer systems, may be viewed as mathematical objects with behavior that is, in principle, well determined. This makes it possible to specify programs using mathematical logic, which constitutes the intended (correct) behavior. Then, one can try to give a

A. Trindade is with the Department of Electricity, Federal University of Amazonas, Manaus, Brazil e-mail alessandrotrindade@ufam.edu.br.

L. Cordeiro is with School of Computer Science, The University of Manchester, UK e-mail lucas.cordeiro@manchester.ac.uk.

Manuscript received September 19, 2018; revised November 26, 2018.

formal proof or otherwise establish that the program meets its specification, as defined by [11].

This area of research is referred as formal methods as defined in [12]. Their aim is to establish system correctness with mathematical rigor.

In the recent decades, research in formal methods has led to the development of very promising verification techniques that facilitate the early detection of bugs in order to ensure the correctness of the system.

Model-based verification techniques are based on models that describe the possible system behavior in a mathematically precise and unambiguous manner. Thus, such problems as incompleteness, ambiguities, and inconsistencies, which normally are discovered only in later stages of the design, can be detected in advance, as described by [11]. Algorithms can verify the system model through systematically explore all their states.

Bringing this idea to the field of solar photovoltaic systems, instead of finding a bug or a software inconsistency, the present work aim to verify a complete PV system project, informing in which condition the project do not meet user requirements or will fail. Among the non-deterministic variables evaluated, is it worth to mention environmental issues (like solar irradiance, and temperature), and the variation of the electrical load (power consumption) or the battery's state of charge, as shown in Fig.1. That is the main contribution of this paper.

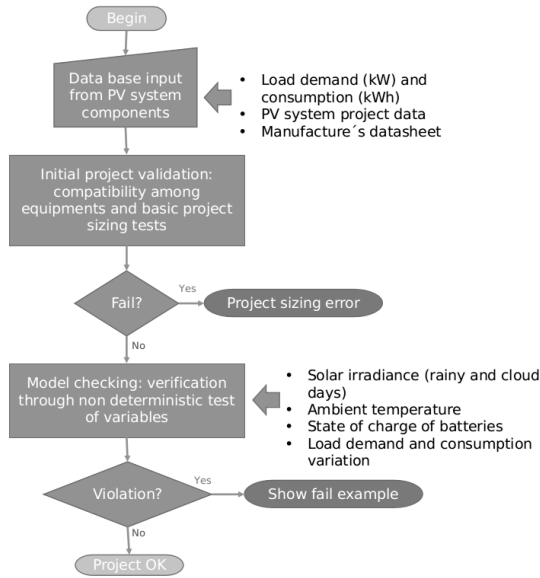


Fig. 1. Flowchart of the proposed computer verification of PV systems

The framework provided at this paper outlines the requirements, the process and the mathematical modeling to perform automated formal verification of a complete PV off-grid system, using the satisfiability modulo theories (SMT)-based verification method.

To the best of our knowledge, this is the first work to use an SMT-based verification tool to solve this kind of problem.

The implementation of the ideas will be done with the efficient SMT-based bounded model checker (ESBMC) tool that uses the front-end of the C Bounded Model Checker (CBMC)

or the clang-based front-end, which permits to describe the problem as a C/C++ language code.

The remainder of this paper is organized as follows: Sect. 2 gives a background of solar PV systems. Section 3 describes the design and validation of a solar PV system. Section 4 describes the formal mathematical modeling of a generic PV system. Section 5 presents the automated verification theory, including details of the ESBMC process. The methodology is presented at Section 6. Section 7 discusses the related work. Section 8 presents the conclusion and describes future work. At the end of the paper, there is a list of all nomenclatures used at the text.

## II. SOLAR PHOTOVOLTAIC SYSTEM

According to [13], a PV system is designed to give the electric supply to a load. This load can be either Alternate Current (AC) type or Direct Current (DC) type. Electric supply can be needed either in daytime or evening time (in particular cases, in both times). The most basic PV system can give supply only in daytime. For night hours or rainy days, one needed to have batteries, where power can be stored and used [14].

PV systems are broadly classified into three distinct types, as described by [15]:

- Standalone systems, where the energy is generated and consumed in the same place and which does not interact with the main grid. Normally, the electricity consuming/utilizing device is part of the system, i.e., solar home systems, solar street lighting system, 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 either a grid-tied system, which can only feed power into the grid and such 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 system: In a hybrid system, another source(s) of energy, such as wind, biomass or diesel, can work together with the solar PV system to provide the required demand. In such type of system, main objective is to bring more reliability into the overall system at an affordable way by adding one or more energy sources.

Specifically concerning isolated communities, depending on the type of load, cost, resources availability, and requirements of the load, standalone systems can be split into several categories, which are described in this section. As the goal of this Thesis is just the solutions aimed to isolated/ off-grid application, therefore it is not considered the on-grid or hybrid configurations.

There is a resource, called maximum power point tracking (MPPT), which is an electronic control mechanism that maintains the PV operating in a voltage that correspond to the voltage of maximum power, which maximizes the transfer of

power and avoiding lost at the PV cells [16]. That resource is found at modern PV systems and is strongly indicated due to its advantages.

#### A. Unregulated standalone PV system with DC load

Usually this type of system is for low power applications, as defined by [13]. The PV system is directly connected to the load without any MPPT controller, as shown at Fig.2 letter (a). At night hours, the system will not provide any supply because of the absence of the battery.

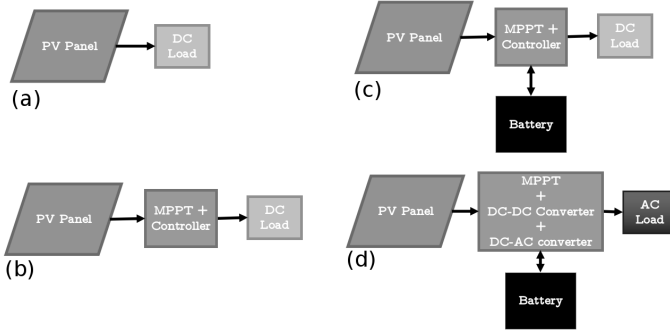


Fig. 2. PV standalone systems: (a) regulated standalone SPV system with DC load; (b) regulated standalone SPV system with DC load; (c) regulated standalone SPV system with battery DC load; and (d) regulated standalone system with battery and AC load (Source: [13])

#### B. Regulated standalone PV system with DC load

It is similar to unregulated standalone system with DC load, but the main difference between this and the previous one is that this system requires a MPPT technique, as illustrated by Fig.2 letter (b). Usually system with MPPT should have battery; otherwise, extra power will be waste.

#### C. Regulated standalone system with battery and DC load

Configuration with PV array, battery, MPPT and DC load, as shown in Fig.2 letter (c). Battery is used to store the extra power of PV system. A charge controller is necessary for this type of system because the useful life of the battery is less than that of the PV module. Extra charging and deep discharging can reduce the battery life [17].

According to [16], PV systems that are used to feed loads with low variation on the consumption can be sized to operate without the controller. That is called of self-regulated standalone PV system with battery. However, the voltage from the PV panel must be compatible with the batteries voltage. Normally, the bank of batteries are oversized related to the PV panel and to the load. The drawback is the operation of the batteries, normally overloaded or with excessive discharges (that can damage the batteries).

#### D. Regulated standalone system with battery, AC load

This system is similar to the previous one, but here AC load draws the power from PV system and, because of the AC load, an inverter (DC to AC converter) is required, as seen in

Fig.2 letter (d). This solution has an increase of cost because has more equipments. However, the AC availability brings the advantage of a higher number of AC appliances to use at the houses or consumer units. That configuration is the focus of the present thesis since it is the most common nowadays all over the world, when off-grid or isolated regions is the target.

### III. DESIGN AND VALIDATION OF A SOLAR PHOTOVOLTAIC SYSTEM

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

In order to address different aspects of project system and systems design, there is a myriad of public domain and commercial software available for the SPV market. According to [9], the capabilities of these tools range from simple solar resource and energy production estimative, to site survey and system design tools, to complex financial analysis software (with optimization). Some tools also provide support to rebate programs applications and tax incentives (specific to each country or region), while other programs and worksheets focus on the technical aspects of system sizing and design.

Manufacturers and integrators have yet their proprietary software to perform inverter string sizing and various system sizing and design tools, with the drawback to just include their own products among the possibilities of choice. In this study, the most popular tools are presented: PVWatts, SAM, Homer, RETScreen, and Hybrid2.

#### A. PVWatts Calculator

According to [18] and [19], it 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 [19], to use the calculator, it is necessary to provide 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 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 do not account for many factors that are important in the design of a photovoltaic system, therefore it is necessary 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 a period of 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; Array azimuth angle.

#### B. SAM

SAM or System Advisor Model is a software from the U.S. Department of Energy and National Renewable Energy Laboratory. According to [20] and [21], SAM is intended to help users to determine whether the model meets their project

constraints/specifications, and (also) to provide information for readers who do not plan to use the model, but want to learn about its capabilities. 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 cost of energy estimates for grid-connected power projects based on installation and operating costs and system design parameters that you specify 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 electric 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 electric load. It does not model thermal energy systems that meet a thermal process load. As mentioned in [20], SAM does not model isolated or off-grid power systems, and does not model systems with electricity storage batteries.

### C. HOMER

As defined in [22], actually 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 HOMER web site, but only available for students and nonprofit organizations, as defined in [22], and has not support available. At the short-term only the commercial version will remain.

The commercial version (paid), called HOMER Pro, as defined in [23], is a tool for optimizing micro-grid design in all sectors, from village power and island utilities to grid-connected campuses and military bases. HOMER Pro put together three tools in one product: optimization, simulation, and sensitivity analysis. HOMER Pro provides the detailed rigor of chronological simulation and optimization in a model that is intended to be easy to use. It 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. Chronological 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.

### D. RETScreen

As mentioned in [24], 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 Suit, 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 actual (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. However, this version works is only Windows-based. This version has a complete paid version, in an annual subscription way.

As described by [24], RETScreen performs a five-step standard 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.

### E. Hybrid2

The Hybrid2 software package, as described in [25], is a user-friendly tool to perform 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, solar insolation, and temperature; and the power system is designed or selected by the user, in order to predict the performance of the hybrid power system. 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 [26], probably after the user performs the free download of the tool, it will not work on Windows platforms later than Windows XP, what is a limitation.

TABLE I  
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	X
Main technical (T) or economical(E)	T	T	E	E	T
Optimization			X	X	
Sensitive analysis			X	X	

Table I summarizes the tools described in this paper. Where just Hybrid2 do not have technical support; only HOMER, RETScreen and Hybrid2 perform off-grid system or battery backup analysis; all the tools perform solar photovoltaic analysis; and that only HOMER and RETScreen are complete, including economical analysis or even optimization-sensitive analysis. However, only paid version of those softwares have all the features, and they run only at Windows-based operational systems.

#### F. Paper Proposal *x* Reference Tools

Considering that the focus of this research is on off-grid solutions and supported tools, only HOMER and RETScreen remain for comparative. All tools need some parameters inherently from manufacturer's catalog, so the project starts with manufacturers and integrators tool to define the basic items of the project: panels, inverters, controllers, and batteries. After that, the potential solution is analyzed in another tool to validate or even optimize the solution. Therefore, that is the challenge of this work, i.e., to prove that is possible to use automated verification to validate an off-grid PV solution and to prove that this verification is more efficient and complete than the listed-more common tools.

#### IV. COMPONENT MODELS FOR STAND-ALONE PV SYSTEM

The main purpose of this section is to describe the models for the elements of a standalone PV system: PV generator, battery, controller, inverter, and load. The modeling of the PV system is based on modular blocks, as illustrated in Fig.3, from [27]. The modular structure facilitates the modeling of the other system structures and replacing of elements as, for instance, a DC load instead of an AC load.

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.

##### A. PV generator (cell, module, 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 cell/module/array.

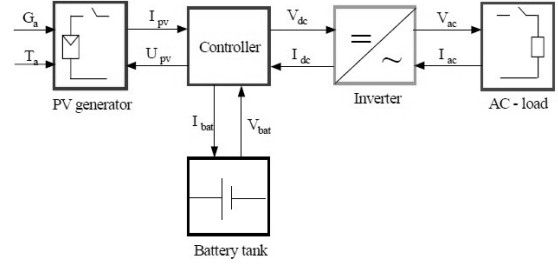


Fig. 3. Block diagram for the stand-alone PV system (Source: [27])

The basic 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 [28]. 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).

To increase their utility, a number of 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. In addition, to achieve the desired voltage and current, modules are wired in series and parallel into what is called a PV Array, as shown in Fig.4. The flexibility of the modular PV system allows designers to create PV systems that can meet a wide variety of electrical demands.

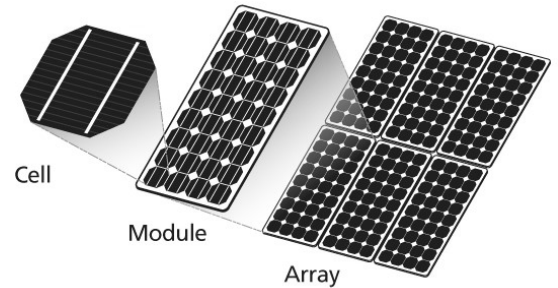


Fig. 4. PV cell, module and array (Source: [29])

The PV modules are generally rated under standard test conditions (STC), which leads to the following specification by the manufacturers: solar radiation of  $1000 \text{ W/m}^2$ , cell temperature of  $25^\circ\text{C}$ , and solar spectrum of 1.5. The parameters required for the input of the PV modules are relying on the meteorological conditions of the area to be serviced by photovoltaic solution. However, the climatic conditions are unpredictable due to the random nature of their occurrence [30].

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

of PV modules [31]. The growing demand of photovoltaics technologies led to research in the various aspects of its components from cell technology to the modeling, size optimization, and system performance [32], [33]; [4], [34], [35], [36], [37], [38]. Modeling PV modules is one of the major components responsible for 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, perfect modeling is essential to estimate the performance of PV modules in different environmental conditions [30].

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

The performance of photovoltaic systems (solar cell/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 [39]. Based on these electronic components, four basic configurations are normally used when studying photovoltaic systems, as shown by Fig. 5.

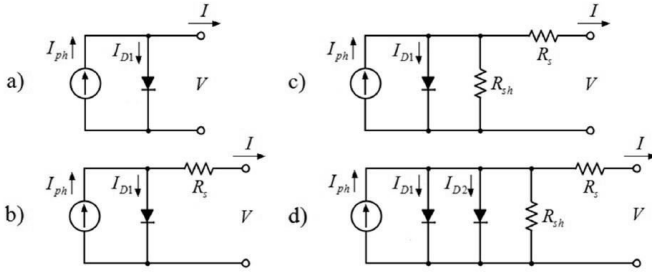


Fig. 5. Four different equivalent circuit models: (a) 1-diode; (b) 1-diode/1-resistor; (c) 1-diode/2-resistor; (d) 2-diode/2-resistor (Source: [39])

The 1-diode model, whose equation to relate the output current,  $I$ , to the output voltage,  $V$ , is described in Equation 1.

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

Where:

- $I_{ph}$  is the photocurrent delivered by the constant current source;
- $I_0$  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 ( $1.3806503 \times 10^{-23} J$ );

- $T$  the temperature of the p-n junction (or cell temperature) expressed in Kelvin;
- $q$  is 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 by Equation 2.

$$I = I_{ph} - I_0 \left[ \exp \left( \frac{V + IR_s}{NaV_T} \right) - 1 \right] \quad (2)$$

Where  $R_s$  is the series resistor.

At 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 by Equation 3.

$$I = I_{ph} - I_0 \left[ \exp \left( \frac{V + IR_s}{NaV_T} \right) - 1 \right] - \frac{V + IR_s}{R_{sh}} \quad (3)$$

Where  $R_{sh}$  is the shunt resistor.

At 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 by Equation 4.

$$I = I_{ph} - I_{01} \left[ \exp \left( \frac{V + IR_s}{Na_1V_T} \right) - 1 \right] - I_{02} \left[ \exp \left( \frac{V + IR_s}{Na_2V_T} \right) - 1 \right] - \frac{V + IR_s}{R_{sh}} \quad (4)$$

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 [30].

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 that reason, researchers gradually focused on searching out the approximate methods for the calculation of unknown parameters and walked through three different paths. The analytical methods give exact solutions by means of algebraic equations, as done by [39] and [40]. However, due to implicit nature and nonlinearity of PV cell or module characteristics, it is hard to find out the analytical solution of all unknown parameters, as described by [36]. Thus, numerical methods such as Newton-Raphson method or Levenberg-Marquardt algorithm were preferred, as described by [38]. This happens because numerical methods give approximate solution of the nonlinear problems without searching for exact solutions. However, numerical methods are time consuming and need long term time series data, which is not available in developing countries. [30] did a mixed methodology using analytical and numerical steps together. [41] create a method to discover the unknown parameters of the PV panels through experimentation (essays). And [42] did a mix of analytical and experimental methodology to get the unknown parameters, but samples of

the PV modules are necessary to perform some essays, when we use experimental technique.

Therefore, a wide variety of models exists for estimation of power output of PV modules (and  $I - V$  or  $P - V$  curves). However, the present work will rely on the simplified model of 1-diode, that was shown by [35] that has a small error rate, between 0.03% and 4.68% from selected PV panels tested. In addition, this mathematical modeling has the advantage of being an explicit model, which does not use iterative numerical calculation.

### B. Proposed PV Panel Model

With the proposed model, an explicit set of equations is derivate from the ideal PV model given by Equation 2.

A single-diode without series and shunt resistances is considered. Equation 1 is used to write down expressions for currents and voltages at each key point shown in Fig. 6.

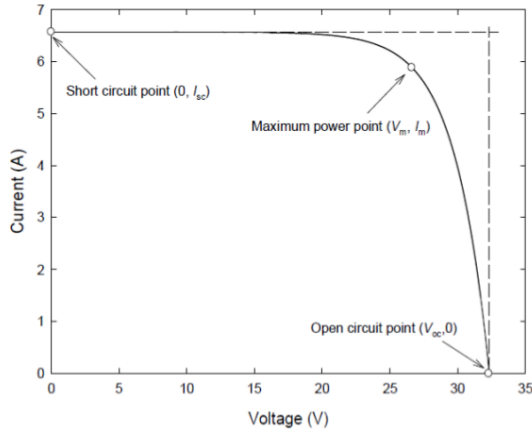


Fig. 6.  $I - V$  characteristic curve of an ideal PV cell (Source: [35])

Hence, the short-circuit current, the open-circuit voltage, the maximum power voltage and current are written as defined by [35] and shown from Equations 5 to 8.

$$I_{sc} = I_{ph}|_{V=0} \quad (5)$$

$$V_{oc} = \frac{aNk_B T}{q} \ln \left( 1 + \frac{I_{sc}}{I_0} \right) \quad (6)$$

$$\exp \left( \frac{qV_{oc}}{aNk_B T} \right) = \left( 1 + \frac{qV_m}{aNk_B T} \right) \exp \left( \frac{qV_m}{aNk_B T} \right) \quad (7)$$

$$I_m = I_{ph} - I_0 \left[ \exp \left( \frac{qV_m}{aNk_B T} \right) - 1 \right] \quad (8)$$

Equation 7 is not explicit with the PV key parameters, therefore need 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 voltage-source at the open-circuit point [35]. These two regions are characterized by two asymptotes of the  $I - V$  curve in Fig. 6, 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 increasing the output voltage (Fig. 6).

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 9.

$$\frac{dI}{dV} = -\frac{qI_0}{aNk_B T} \exp \left( \frac{qV}{aNk_B T} \right) \quad (9)$$

The derivative of Equation 9 is used to calculate the output voltage that corresponds to the maximum power operation condition of the cell. Therefore, the Equation 10 is generated.

$$V_m = \frac{aNk_B T}{q} \ln \left( -\frac{qNk_B T}{qI_0} \left( \frac{dI}{dV} \right)_{V_m} \right) \quad (10)$$

It is evident that Equation 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 on  $V_m$  has a relatively small effect on the maximum power of the cell [35]. Consequently, considering the asymptotic behavior of the  $I - V$  curve at short- and open-circuit conditions, the derivative required by Equation 10 can be calculated as shown in Equation 11.

$$\frac{dI}{dV}|_{V_m} \cong -\frac{0 - I_{sc}}{V_{oc} - 0} = \frac{I_{sc}}{V_{oc}} \quad (11)$$

Replacing the Equation 11 into Equations 10 and 8, the voltage and the current at the maximum power point and consequently the maximum output power, are expressed by Equations 12, 13, and 14. These equations are used to calculate the key cell parameters at the maximum power point as function cell temperature and parameters from the manufacturer's data-sheet.

$$V_m = \frac{aNk_B T}{q} \ln \left( \frac{aNk_B T}{qI_0} \frac{I_{sc}}{V_{oc}} \right) \quad (12)$$

$$I_m = I_{ph} + I_0 - \frac{aNk_B T}{q} \left( \frac{I_{sc}}{V_{oc}} \right) \quad (13)$$

$$P_m = V_m I_m = \left[ \frac{aNk_B T}{q} \ln \left( \frac{aNk_B T}{qI_0} \frac{I_{sc}}{V_{oc}} \right) \right] \times \left[ I_{ph} + I_0 - \frac{aNk_B T}{q} \left( \frac{I_{sc}}{V_{oc}} \right) \right] \quad (14)$$

However, the photocurrent delivered by the constant current source ( $I_{ph}$ ) or even the reverse saturation current ( $I_0$ ) are not given by PV manufacturers. Therefore, Equation 15 is used to calculate the photocurrent as function of irradiance and temperature [43].

$$I_{ph} = \frac{G}{G_{ref}} [I_{ph,ref} + \mu_I (T - T_{ref})] \quad (15)$$

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

In Equation 15,  $\mu_I$  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 [30].

The cell temperature ( $T$ ) can be obtained from [44] and is shown in Equation 16.

$$T = T_{air} + \frac{NOCT - 20}{800}G \quad (16)$$

Where  $T_{air}$  is the ambient temperature,  $NOCT$  is the nominal operating cell temperature (in  $^{\circ}C$ ) that is found at the PV manufacturer's data-sheet, and  $G$  is the solar irradiance ( $W/m^2$ ) of the place where the PV system is deployed.

Furthermore, [43] 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 as well as the open-circuit voltage temperature coefficient (Equation 17).

$$I_0 = \frac{I_{sc,ref} + \mu_I(T - T_{ref})}{\exp\left[\frac{q(V_{oc,ref} + \mu_V(T - T_{ref}))}{a n k_B T}\right] - 1} \quad (17)$$

Where  $V_{oc,ref}$  is the reference open-circuit voltage, and  $\mu_V$  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 [43], is determined in the reference state. Using the maximum power point current equation (Equation 14) and the saturation current in the reference temperature given by Equation 17, the diode quality coefficient is determined by Equation 18.

$$a = \frac{q(V_{m,ref} - V_{oc,ref})}{n k_B T} \frac{1}{\ln\left(1 - \frac{I_{m,ref}}{I_{sc,ref}}\right)} \quad (18)$$

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 manufacturers.

The model is now completely determined; it 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. Therefore, there will be a change at the  $I_{ph}$  and  $I_0$  and the resulting current is given by Equation 19, as demonstrated in [35].

$$I_{array} = (N_{cells \text{ in parallel}})(I_{onecell}) \quad (19)$$

Where  $I_{onecell}$  is the current from Equation 13.

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

$$V_{array} = (N_{cells \text{ in series}})(V_{onepanel}) \quad (20)$$

Where  $V_{onepanel}$  is the voltage from Equation 12.

Besides the model verification, there is the prior stage of project sizing check, which is simple, based on manufacturer's data and information from the sizing and the site.

First is necessary to correct the energy consumption that was estimated to the load ( $E_{consumption}$ ). That is done by Equation 21, according [16]. Where the efficiency of batteries ( $\eta_b$ ), controller ( $\eta_c$ ), and the inverter ( $\eta_i$ ) are considered.

$$E_{corrected} = \frac{E_{consumption}}{\eta_b \eta_c \eta_i} \quad (21)$$

The total minimum number of solar panels necessary ( $N_{TPmin}$ ) is calculated by Equation 22, and the check is done with the Equation 23, where the sized number of panels ( $N_{TP}$ ) must be greater than the result from Equation 22.

$$N_{TPmin} = \frac{E_{corrected}}{E_p} \quad (22)$$

$$N_{TP} \geq N_{TPmin} \quad (23)$$

Particularly, the total number of panels is series ( $N_{PSmin}$ ) and parallel ( $N_{PPmin}$ ) are given by the Equations 24 and 25, respectively. With the check did by Equations 26 and 27.  $V_{system}$  is the DC voltage of the bus, and is a design decision (normally 12, 24 or 48 V).

$$N_{PSmin} = \frac{V_{system}}{V_{m,ref}} \quad (24)$$

$$N_{PPmin} = \frac{N_{TPmin}}{N_{PSmin}} \quad (25)$$

$$N_{PS} \geq N_{PSmin} \quad (26)$$

$$N_{PP} \geq N_{PPmin} \quad (27)$$

### C. The Battery Storage Model

Because of the fluctuating nature of the output delivered by the PV arrays, batteries are necessary in a PV system. Thus, during the hours of sunshine, the PV system feeds directly the load and the excess electrical energy is stored in the battery. During the night, or during a period with low solar irradiation, energy is supplied to the load from the battery [38].

Several models have been presented in the literature. However, regardless of the model, normally the following parameters are required:

- 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: for instance, discharge regime is 30h for a battery of 150 Ah that is discharged at 5A.

- Efficiency ( $\eta_b$ ), 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. The battery efficiency was described in this section, and the liability 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 [45].

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. 7.

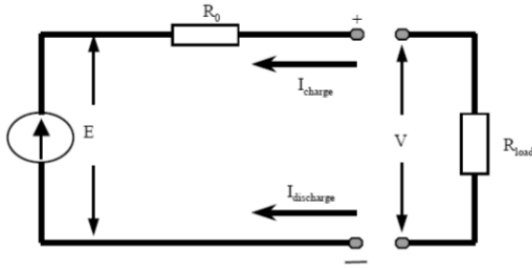


Fig. 7. Schematic diagram of the battery (Source: [27])

The terminal voltage of the battery can be expressed in terms of its open circuit voltage and the voltage across the internal resistance of the battery [46], as shown by Equation 31.

Where  $V_b$  is the battery terminal voltage,  $E_{oc}$  is the battery circuit voltage,  $I_b$  is the battery current, and  $R_b$  is the internal resistance of the battery.

The battery model, which describes the relationship between the voltage, current and the state of charge, can be found in [45], [47], and [48].

The Kinetic Battery Model (KiBaM) of Manwell and McGowen [47] 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 tested batteries in laboratory. Therefore is not suitable to this study.

Most of the created models were used to simulate and optimize PV storage system based on lead-acid batteries. That kind of battery is the most common used batteries in PV application, owing to their relative low cost and wide availability [45].

Here, the model adopted is the based on [45], 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 28. 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.

$$V_d = [2.085 - 0.12(1 - SOC)] - \frac{I}{C_{10}} \left( \frac{4}{1 + I^{1.3}} + \frac{0.27}{SOC^{1.5}} + 0.02 \right) (1 - 0.007\Delta T) \quad (28)$$

Where  $C_{10}$  means 10h of rated capacity, which is standard on the manufacturer's data-sheet,  $\Delta T$  is temperature variation ( $\Delta T = T - T_{ref}$ ,  $T_{ref} = 25^\circ C$ , SOC indicates how much electric charge is stored in the cell at given time, defined by Equation 29.

$$SOC = \left( 1 - \frac{Q}{C} \right) \quad (29)$$

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.,  $DOD = 1 - SOC$ .

The efficiency of the battery discharge is assumed to be 100%, according [45]; however, the total amount of useful charge available during discharge is limited by the current rate and temperature given by Equation 30. This equation, called of capacity, is normalized with respect to discharge current corresponding to  $C_{10}$  rated capacity ( $I_{10}$ ).

$$\frac{C}{C_{10}} = \frac{1.67}{1 + 0.67 \left( \frac{I}{I_{10}} \right)^{0.9}} (1 + 0.005\Delta T) \quad (30)$$

Note that when the discharge current tends to zero at  $25^\circ C$ , the maximum capacity that can be removed is about 67% over the capacity.

For the charging process, however, the parameters are presented in Equation 31.

$$V_c = [2 + 0.16SOC] + \frac{I}{C_{10}} \left( \frac{6}{1 + I^{0.86}} + \frac{0.48}{(1 - SOC)^{1.2}} + 0.036 \right) (1 - 0.025\Delta T) \quad (31)$$

SOC can be calculated easily at any point during the discharge period; however, during recharge it is much more difficult [45].

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 ( $\eta_c$ ) variation with state of charge and current rate is given in Equation 32.

$$\eta_c = 1 - \exp \left[ \frac{20.73}{\frac{I}{I_{10}} + 0.55} (SOC - 1) \right] \quad (32)$$

[45] show that, during overcharge, gassing occur and tests demonstrated that the final charge voltage ( $V_{ec}$ ) increases with the current intensity and with the decreasing of the temperature

(Equation 33). It was created a function for the gassing voltage as well, as shown in Equation 34. In addition, the overcharge phenomenon can be represented by Equation 35.

$$V_{ec} = \left[ 2.45 + 2.011 \ln \left( 1 + \frac{I}{C_{10}} \right) \right] (1 - 0.002 \Delta T) \quad (33)$$

$$V_g = \left[ 2.24 + 1.97 \ln \left( 1 + \frac{I}{C_{10}} \right) \right] (1 - 0.002 \Delta T) \quad (34)$$

$$V_c = V_g + (V_{ec} - V_g) \left[ 1 - \exp \left( \frac{Ah_{restored} - 0.95C}{I\tau} \right) \right] \quad (35)$$

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 was already restored at the start of overcharge.

The time constant of the phenomenon ( $\tau$ ) is reversely proportional to the charge intensity and can be written by Equation 36.

$$\tau = \frac{17.3}{1 + 852 \left( \frac{I}{C_{10}} \right)^{1.67}} \quad (36)$$

Therefore, to model the voltage ( $V_c$ ) evolution of the battery, equation 31 can be used up the start of gassing ( $V_c \leq V_g$ ). And during overcharging ( $V_c > V_g$ ), Equation 35 can be used until a constant final voltage ( $V_{ec}$ ) is reached.

The storage capacity of the battery can be calculated using Equation 37, as defined in [49].

$$Storagecapacity = \frac{N_C E_{load}}{DOD \eta_b} \quad (37)$$

Where  $DOD$  is the maximum possible depth of battery discharge,  $E_{load}$  is the average energy consumed by the load,  $N_C$  is the largest number of continuous cloudy days of the area, and  $\eta_b$  is the efficiency of the battery.

As an example of this formula application, as shown by [50], 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 discharge for the battery  $DOD$  of 0.8 and battery efficiency 80%.

Then the storage capacity using Equation 37 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 of 12 V and 350 Ah is considered, then four batteries are connected in series ( $4 \times 350Ah = 1400Ah$ ).

At this research, it was considered a simplified model for charging (Equation 38) and discharging (Equation 39) of the batteries, even considering that the process is not linear and depends on the temperature. 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

at the charging equation, and is necessary to express that during the charging process is usual to reach 115% of the battery capacity

$$SOC_{charge} = SOC_{previous} + \frac{100 * P_m * Num_h}{V_{system} * capacity * N_{BP} * 1.15} \quad (38)$$

$$SOC_{discharge} = SOC_{previous} - \frac{100 * I_{drained} * Num_h}{capacity} \quad (39)$$

Besides the model verification, there is also the prior stage of project sizing check, as did to the solar panel. First is necessary to define the total capacity of the battery bank, as shown by Equation 40.

$$C_{bank} = \frac{E_{corrected} \times autonomy}{V_{system} \times DOD} \quad (40)$$

The variable *autonomy* is a design definition and normally has a value ranging from 6 to 48h. The other variables were discussed previously.

Following, is done the total (minimum) number of batteries, as show by Equation 41. Moreover, the Equation 42 perform the final sizing check, considering the number of batteries in series ( $N_{BS}$ ) and the number of batteries in parallel ( $N_{BP}$ ) that were established to the project.

$$N_{Btotal} = N_{BSmin} \times N_{BPmin} = \frac{V_{system}}{V_{bat}} \times \frac{C_{bank}}{C_{20}} \quad (41)$$

$$(N_{BS} \times N_{BP}) \geq N_{Btotal} \quad (42)$$

#### D. Controller Model

Depending on the literature, the controller can receive different names: controller [27], charge controller [51] and [7], regulator [38], DC-DC converter with MPPT and switch [52], [4], [50], [13]. 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 and switches) and can be defined as the responsible to manage the energy flow to PV system, batteries and loads by collecting information on the battery voltage and knowing the maximum and minimum values acceptable for the battery voltage. According to [16], controllers aim to protect the battery (or batteries) against the excessive charge and discharge, improving its lifetime.

As defined by [27] and [38], all power systems must include a control strategy, which describes the interactions between its components. The use of battery as a storage form implies thus 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.

In [38] was established that the controller allows the management of energy between the load and the battery. The input signals for regulator model are the battery current ( $I_{br}$ ), PV generator's voltage ( $V_{PV}$ ), PV generator's current ( $I_{PV}$ ), and battery voltage ( $V_b$ ). The outputs are battery ( $I_{rb}$ ) current and used current ( $I_u$ ).

According [27], to protect the battery against an excessive charge, the PV arrays are disconnected from the system, when the terminal voltage increases above a certain threshold  $V_{max\_off}$  and when 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. 8.

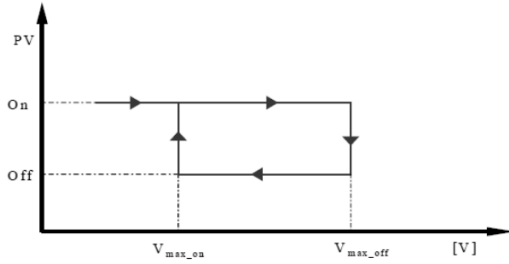


Fig. 8. Operating principle of an overcharge protector (Source: [27])

To protect the battery against excessive discharge, the load is disconnected when 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 [27]. 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. 9.

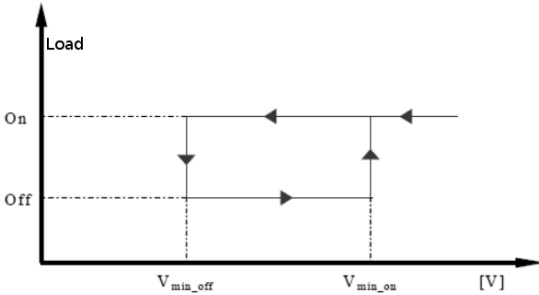


Fig. 9. Operating principle of a discharge protector (Source: [27])

According to [28], 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 II.

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

As shown in [50], the DC-DC converter is used to increase the efficiency of the PV system by matching the voltage generated by PV array to the voltage required by the load. The output power ( $P_{out}$ ) of DC-DC converter is given by Equation 43.

TABLE II  
SUMMARY OF THE CONTROLLER PROCESS (SOURCE: [27])

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

$$P_{in}=P_{out} \quad (43)$$

Assuming that the efficiency of the controller ( $\eta_c$ ) is a manufacturer's data, from Equation 43 is possible to reach Equation 44.

$$V_{in}I_{in}\eta_c = V_{out}I_{out} \quad (44)$$

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

The output voltage is related to the input voltage as a function of duty cycle of the switch ([50]).

A DC-DC converter can either be step-up (Boost), step-down (Buck), or both increase and decrease (Buck-Boost) the voltage, as defined by [51]. In addition, there is the Cuk converter, which is a Buck-Boost converter with an inverting topology [53].

For the Cuk converter, the relationship is expressed by 45 as show in [50].

$$\frac{V_{out}}{V_{in}} = \frac{D}{D-1} \quad (45)$$

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

The DC/DC converter should always operate in the MPPT to maximize the PV array efficiency and consequently increase the efficiency of the PV system, as defined in [4].

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 by [54].

As the MPPT definition and the equations to get the maximum power from the PV panels was described at the end of the PV panel modeling, the important here is to notice that the Equation 45 defines the relationship between the input signal, the efficiency of the controller and the output power.

One more time, some steps must be done to check the sizing project of the controller, prior the verification phase. Initially the controller must meet the voltage requirement of the PV system, as shown by Equation 46.

$$V_c = V_{system} \quad (46)$$

Following, the short circuit reference information from the manufacturer's solar panel must be corrected to the cell temperature, as shown by Equation 47.

$$I_{sc,amb} = I_{sc,ref} \times [1 + \eta_I \times (T - 25)] \quad (47)$$

The controller must meet the maximum current from the PV array (Equations 48 and 49).

$$I_{c,min} = I_{sc,amb} \times N_{PP} \quad (48)$$

$$I_c \geq I_{c,min} \quad (49)$$

The number of controllers required for the off-grid PV system, as defined by [4], is calculated using Equation 50. In addition, the final sizing check is did by Equation 51, who validate the number of controllers adopted.

$$number_{controllers} = \frac{\text{Total max power of PV}}{\text{Controller max power}} = \frac{P_{m,ref} \times N_{TP}}{V_{system} \times I_{controller,max}} \quad (50)$$

$$N_{controller} \geq number_{controllers} \quad (51)$$

#### E. Joint work: controller and batteries

When batteries are charged, they go through three main different states - bulk, absorption and float, and it is necessary to understand those three states in order to comprehend the joint work of the controller with the battery. With impact to the inverter and to power supplied to the house.

All the explanation at this section is related to *SOC* and to the voltage at the DC-bus (controller output, which will be bigger than  $V_{system}$ ) where the batteries are connected, all ruled by the charge controller. Moreover, it is important to mention that smart chargers, the most usual nowadays, will detect voltage and resistance from the battery prior to charging.

Bulk - is the first stage of charging. Bulk begins when the sun comes out, or other source of electrical generation is turned on. This stage occurs when the batteries are at a lower *SOC*, generally anything not smaller than 75% to 80% (defined during system sizing). This stage is typically where the highest voltage and current the charger is rated for will actually be used. The bulk stage basically allows the PV panel to put as much current into the batteries as possible. For a typical 12 V battery, the charging voltage going into a battery will reach 14.16 V to 14.40 V at this stage, and there is no risk of overcharging in this stage because the battery hasn't even reached full yet.

Absorption - once the batteries reach the programmed absorb voltage, usually somewhere between 14.16 and 14.40 V, the batteries will go into the absorption stage. Typically, when a battery reaches this stage its *SOC* is around 85–95%. During this stage, the batteries are kept at the programmed voltage, and the current going into the batteries reduces as the batteries become more full. The absorption stage ends after the programmed time is reached or the number of Ampere

going into the battery falls below a preset number. The lower current going into the battery safely brings up the charge on the battery without overheating it. Usually, this stage takes more time than the bulk stage.

Float - upon the completion of the absorption stage, the charge controller will drop down the voltage and maintain at a steady 13.20 V to 13.38 V (manufacturer's defined value), which is the maximum voltage a 12 volt battery can hold, and begin the float stage. The float stage brings the battery all the way through and maintains *SOC* at 100%. The current will also decrease to a point where it's considered just a pulse. It's essentially the float stage where there is charge going into the battery at all times, but only at a safe rate to ensure a full state of charge and nothing more. Most smart chargers do not turn off at this point, however it is completely safe to leave a battery in float mode for months to even years at a time.

Battery charging technology relies on smart micro-controlled devices. Therefore it is extremely important, that the program settings for the charge controller or inverter/charger are correct, and the list includes the bulk and absorption values. This will help preserve and extend the battery life. There are always default settings to the equipment, set by software, however those settings are not 100% correct because there are difference among models of equipment and among manufacturer's. The ideal is to read the data-sheet from the manufacturers and perform a manual adjust.

There are the possibility to combine batteries in a bank, with parallel and series connections, all depending the sizing of the project:

- Batteries connected in parallel are seen by the controller as one large battery of the combined Ah capacity of all the batteries. For example, two 12 V and 220 Ah batteries in parallel are seen as one 12 V and 440 Ah battery;
- In the other hand, although the batteries connected in series are also seen as a single battery, the result is different: two 12 V and 220 Ah batteries in series are seen as one 24 V and 220 Ah battery;
- In addition, a mixed bank arrangement can be useful: a bank with two 12 V and 220 Ah batteries in series connected in parallel with a similar arrange will produce an equivalent battery of 24 V and 440 Ah.

#### F. The inverter model

As shown by [38], the PV arrays produce DC and therefore when the PV system contains an AC load, a DC/AC conversion is required. An inverter is a converter, where the power flows from DC to AC side, i.e., having a DC voltage as input; it produces 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 the input power  $P_{in}$  into the output power  $P_{out}$  with the best possible efficiency [27].

The inverter is characterized by a power dependent efficiency  $\eta_i$  given by [27] as shown by Equation 52.

$$\eta_i = \frac{P_{out}}{P_{in}} = \frac{V_{AC} I_{AC} \cos \varphi}{V_{DC} I_{DC}} \quad (52)$$

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),  $V_{AC}$  and  $I_{AC}$  are the output voltage and current, respectively, and  $\cos\varphi$  can be found from the inverter's data-sheet.

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 feed the inverter (which are obtained by this study model).

The sizing project check of the inverter is done through three Equations. Equation 53 ensure that the input voltage of the controller meet the voltage of the system. Equation 54 ensures that the output voltage of the controller meet the AC voltage of the load. Moreover, Equation 55 ensures that the controller can support the total demand of the load and the surge power.

$$V_{inDC} = V_{system} \quad (53)$$

$$V_{outAC} = V_{AC} \quad (54)$$

$$[(Demand \leq P_{AC,ref}) \text{ and } (P_{surge} \leq MAX_{AC,ref})] \quad (55)$$

## V. AUTOMATED VERIFICATION

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, resulting in a process known as formal verification. Verification may form part of a validation process, but in general, validation cannot be formalized because it relates a system design to intent.

Simulation may also be used for validation, but it is more problematic 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 is problematic for verification.

According [55], verification procedure is an intelligent exhaustive search of the state space of the design. In addition, according to [56], 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.

Model checking is an automatic verification technique, as defined by [55]. Model checking was originally developed reasoned about finite state of concurrent systems, nowadays is mainly used to 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 (normally mathematical) of the system is created;
- In specification, normally a list of properties to be satisfied by the system is established, i.e., the requirements, as reliability to performance for example.
- Normally is expressed 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  $\phi$  as input;
- Model checking algorithm provides all the states of model  $M$  which satisfies  $\phi$ ;
- It returns *YES* if  $\phi$  is *TRUE*, or returns *NO* if  $\phi$  is *FALSE*.

Specifically at the *FALSE* situation, the algorithm returns a counterexample that is useful as diagnostic of the system, in order to discover in which situation the model is violated. [55] consider that the most important advantage of the use of model checking.

Among the other advantages can be listed: there is no need of proofs (the algorithm is not a deductive procedure), there is no problem with partial specifications of the system, logics can easily express many concurrency properties, is fast (compared to other rigorous methods such as interactive theorem proving). However, there is a main disadvantage of model checking: the state explosion problem.

The model checking problem can be defined as shown in [55]:

- Let  $M$  be a Kripke structure (i.e., 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. 10 shows the structure of a typical model checking system. A preprocessor extracts a state transition graph from a system (program or circuit).

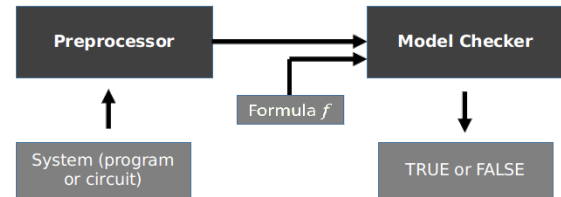


Fig. 10. Model Checker structure (Source: [55])

Here it is worth to mention that the term "model" is not the meaning taken from the dictionary. The problem is not dealing with an abstraction of the actual system under study.

The Fig. 11 shows the process to convert a real system to a model in order to be verified by a model checking.

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

BMC is a method that checks the model up to a given path in the path length. BMC algorithms traverses a finite state machine for a fixed number of steps, and checks whether violation occurs with this bound. It uses fast SAT solvers, where SAT means satisfiability. SAT problem, as defined by [55], is a problem of determining if there are certain conditions

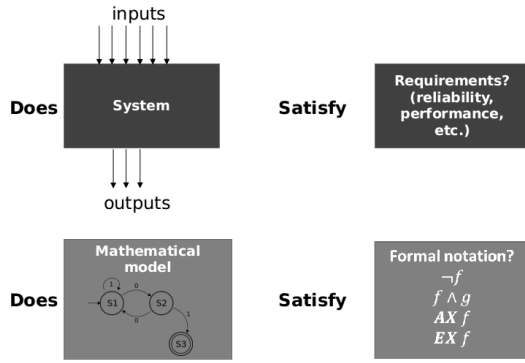


Fig. 11. From real system verification to model checking (adapted from [55])

or interpretation that satisfy a given Boolean expression. SAT solvers are used in BMC, such that if there are some Boolean function, the solver would search the model for conditions (value of variables) that would make the formula *TRUE*. If SAT Solver find a substitution for the formula/function then the substitute induces a counterexample.

CBMC is considering the best-known model verification tool to validate code in ANSI-C and C++, as can be seen in [57].

ESBMC is a context-bounded model checker for embedded C/C++ software based on Satisfiability Modulo Theories (SMT) solver, which can use CBMC as front-end. The use of SMT, instead of Boolean Satisfiability SAT from the original BMC, comes as an alternative to overcome limitations of the systems modeling, especially considering that the complexity of these is increasing and the SMT method has high level and richer theories than the SAT to represent the models.

#### A. ESBMC

According [58], ESBMC is an open source, permissively licensed (Apache 2), cross platform bounded model checking for C and C++ programs. the efficient SMT based model checker, is a software verification tool for C and C++ code bases. Fig. 12 shows the tool architecture.

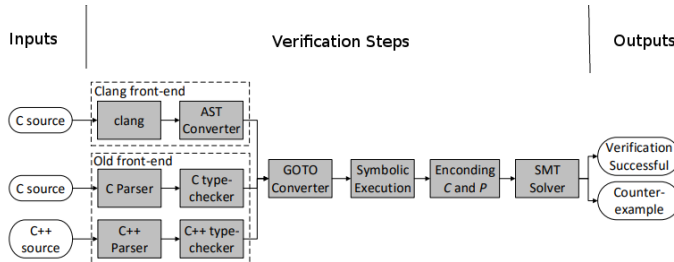


Fig. 12. ESBMC architecture (adapted from [58])

As shown by [58], ESBMC has two alternative front-ends to parse the input program and generate an Abstract Syntax Tree (AST). One legacy CBMC-based front-end that supports both C and C++ and a new clang-based front-end that currently only supports C. The data types and bit vectors are created in the front-end when parsing the code.

Regardless of the chosen front-end, the output is an AST (tree representation of the abstract syntactic structure of source code written in a programming language) that will be used by the GOTO converter to generate a GOTO program, which has simplified control flow and is suitable for bounded unwinding. The next step is the symbolic execution, when the GOTO program is executed (unrolling loops up to the bound  $k$ ) and converted to Static Single Assignments (SSA) form. SSA form enables the simplification of algorithms and reduction of computational complexity, where each variable is a target of exactly one assignment statement in the program text.

During the symbolic execution, ESBMC aggressively tries to simplify the program [59]; it propagates all constants and solves any assertions that can be statically determined. This is an important step for the verification, since ESBMC can fully verify programs without calling a solver, if the inputs are deterministic. That characteristic is useful at this work to perform the sizing check of the PV system project.

Following, the SSA expressions are then encoded using the chosen SMT solver supported by ESBMC: Boolector (default) [60], Z3 [61], MathSAT [62], Yices [63] and CVC4 [64].

Finally, the system attempting to determine whether a formula, which is the disjunction of all possible errors, can be satisfied. If the SMT formula is shown to be satisfiable, a counterexample is presented; if the formula is found to be unsatisfiable, there are no errors up to the unwinding bound, and this result is presented.

ESBMC can be invoked through the command-line interface or configured through the Eclipse plug-in. Specific problems and the solver can be selected that way.

ESBMC aims to support all of ISO/IEC 9899:1999 C programming language standard, and detects errors in software by simulating a finite prefix of the program execution with all possible inputs. Classes of problems that can be detected include: user specified assertion failures, out of bounds array access, illegal pointer dereferences, double-free of malloc'd memory, misaligned memory access; integer overflows, divide by zero, memory leaks, concurrent software, deadlock, and data races (i.e. competing writes).

#### B. Cyber-physical systems x Energy x ESBMC

According [65], cyber-physical systems (CPS) are integrations of computation, networking, and physical processes. Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice-versa. The economic and societal potential of such systems is vastly greater than what has been realized, and major investments are being made worldwide to develop the technology. The technology builds on the discipline of embedded systems (computers and software embedded in devices) whose principle mission is not computation, such as cars, toys, medical devices, and scientific instruments. CPS integrates the dynamics of the physical processes with those of the software and networking, providing abstractions and modeling, design, and analysis techniques for the integrated whole.

Energy production, distribution, and optimization are all CPS problems, as mentioned in [65]. For example, the

smart grid combines multiple electric power production plants with a multiplicity of loads using dynamic load balancing and dynamic pricing with demand-response strategies. Smart buildings integrate sensors into control systems for lighting, HVAC (heating, ventilation, and air conditioning), and safety (as fire monitoring and evacuation).

Therefore, in the context of this proposal, a cyber-physical system, specifically a solar photovoltaic system, will be mathematically modeled. Then a model checking will be performed to do a verification of the project in order to guarantee that the design of an off-grid SPV solution will meet the requirements, specifications and regulations.

As described in [66], in program verification, it is aimed to check if a program satisfies its logical specification. Contemporary verification tools vary widely in terms of source languages, verification methodology, and the degree of automation, but they all rely on repeatedly invoking an SMT solver. This work tracks the same path.

## VI. METHODOLOGY

Fig. 13 shows the state diagram for the proposed framework.

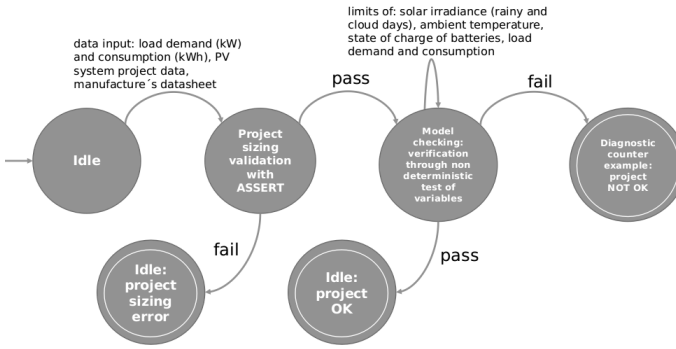


Fig. 13. State transition diagram to the proposed framework

It is worth mentioning that the steps described here are carried out shortly after the design of the solar photovoltaic system, as soon as the equipment and equipment specifications are defined; before buying and deploying them, as a way to ensure that the project will not fail.

The PV input data (load power demand, load consumption of energy, the PV system project data, and the manufacturer's data-sheet information), the formulas to check the sizing project, the mathematical model, the limits of the non-deterministic variables of the model, are all written as C Programming language code.

Then the tool ESBMC is put to perform the verification of this code. The nondeterministic variables (solar irradiance, and ambient temperature by default; as the state of charge of the batteries, load demand of power and the load consumption of energy if defined by the code) will suffer a heavy test at ESBMC runtime execution. In addition, depending if one of the requirements of the system, as autonomy of the batteries, or the offer of the energy, or even the power supply from the system, is not accomplished or satisfied, the tool accuses a fail,

and the diagnostic counterexample shows in which condition the fail/violation happened.

Simply put, instead of the ESBMC find the defect of a software or if a hardware does not meet the some requirements (as reliability, performance, or safety), the tool will verify if the requirements of the solar PV project are met. Detailing the condition of violation, of not meeting the requirements, if this occurs.

### A. The case studies and the Algorithm

At the present work, it was performed two case studies to evaluate the proposed technique: (1) one concerning a PV system (700 W, with 48h autonomy) deployed at four different houses at an indigenous community called Nova Esperança (GPS coordinates 2°44'50.0"S 60°25'47.8"W) sited around 1.5 h by speedboat from Manaus (Amazonas-Brazil), with each house having a different power demand (253 W, 263 W, 283 W, and 501 W); (2) and one case concerning a system deployed as an individual system at ordinary house in Manaus (GPS coordinates 3°4'20.208"S 60°0'30.168"W) as an alternative to pay less electric bill and as backup at night supporting 915 W of the house's load (1,200 W system, but with autonomy of just 6 h).

Related to 700 W system, the sized equipment were: 3 x 325 W PV panels connected in series, controller of 150 V/35 A with a DC-bus of 24 V, 4 x 220 Ah batteries (2 in series and 2 in parallel arrangement), and inverter of 700 W.

And to the 1,200 W PV system: 4 x 325 W connected in series PV panels, with controller of 150 V/35 A in a DC-bus of 48 V, 4 batteries of 120 Ah connected in series, and a 1,200 W inverter.

As demonstrated at this work, the performance of the system is highly dependent of solar irradiance and temperature, that are specific of the deployed local (latitude and longitude). At both case studies, it was considered historic data from Manaus city (because the indigenous community do not have data and is not far from Manaus). According with X and Y, the minimum average temperature during a entire year (21 years on record) is 23°C and the maximum average temperature is 32°C; and irradiance varies from 0 W/m<sup>2</sup> to 852 W/m<sup>2</sup> (with minimum of 274 W/m<sup>2</sup> during the daytime, when there is sunlight).

Another characteristic of Manaus, analyzing the data from a entire year (solar irradiance analysis), is that there is PV generation only between 8:00 h and 16:00 h every day, with zero electric energy generation from 18:00 h to 6:00 h of the next day; and with insignificant generation from 6:00 h to 8:00 h, and from 16:00 h to 18:00 h of the same day.

Therefore, in order to reduce the computational effort, caused by the state explosion inherent of the technique, the 24h-day was divided, at the tests, in 3 parts of 8 h each (8:00h-16:00h, 16:00h-24:00h, 00:00h-8:00h); and only the 8:00h-16:00h period is where there is PV generation and the non-deterministic variables  $G$  and  $T$  are considered during the formal verification of the system, otherwise, during the other two periods, there is no PV generation and just the power consumption from the backup batteries. Within this 8h-period,



$G$  and  $T$  are automated verified with different values, and change their value every 1 h according with the algorithm created using the technique.

Concerning the unique characteristic of each one of the case studies, close related to the autonomy of the batteries, it was created different algorithms to perform the automated verification of each one. Algorithm 1 describes the pseudo code used to the 700 W PV system, and Algorithm 2 the 1,200 W PV system.

There is a premise considering each algorithm: (1) to the 48 h autonomy, is considered that at the end of the first 8h-daytime cycle will be enough to charge the batteries. Otherwise, could be no sense to verify the system with the batteries low and with high probability to not meet the electric demand of the house. Therefore, that is the start point of this algorithm, beginning at 16:00h of the day 0 with the  $SOC$  of batteries in 100%. And perform the automated verification until the end of 48 h of sized autonomy; (2) related to the 6 h autonomy PV system, in the other hand, is considered a  $SOC$  of 80% at 8:00 h, and is made the automated evaluation during the 8h-daytime (from 8:00 h to 16:00 h) and plus the next 6 h, considering that the autonomy must ensure mainly the energy from the PV system during the 16:00 h to 22 h period. For both algorithms, there is a array, declared at the beginning of the C code, that defines the power demand/consumption at each hour of the day (aka load curve). Those are the same data used during the sizing of the project and ensure to consider the total energy (per day) supplied to the house during the verification.

In both cases, if the verification founds a combination ( $SOC$ ,  $G$  and  $T$ ) where the electrical demand of the house is not met, then the algorithm presents a counterexample of this combination. And this information can be used by the designer to evaluate if it was a project error or if it was a combination of variables from the nature that has a low probability to occur at the field, considering the intermittent characteristic of solar PV generation.

## VII. RESULTS

This section is split into two parts. The setup is described in Section VII-A, while Section VII-B describes all experimental results.

### A. Experimental setup

All experiments were conducted on an otherwise idle Intel Core i7-6700 (8-cores), with 3.4 GHz and 16 GB of RAM, running Ubuntu 64-bits.

Concerning the SMT tool, ESBMC v5.0 was used. And regarding the solver, version 4.6.0 of Z3 was used.

At the terminal, the command used to perform the verification is: `$ esbmc filename.c --no-bounds-check --no-pointer-check --no-div-by-zero-check --unwind 300 --smt-during-symex --smt-symex-guard -z3`

Where:

- The first three parameters, after the filename, are related to options that are usual to find bug in software, for example, like bound check to arrays, pointer check, and

---

### Algorithm 1 Algorithm for the 700 W - 48 h PV system

---

```

1: Perform project sizing validation( )
2: if (FAIL validation) then
3:   exit ("Project sizing error")
4: end if
5:  $SOC \leftarrow 100\%$ 
   {Now the 16:00h-08:00h period just discharging batteries}
6:  $SOC \leftarrow SOC - 16\text{ h discharge}$ 
   {Now the 08:00h-16:00h automated verification}
   LOOP Process
7: for  $h = 1$  to 8 do
8:    $G \leftarrow nondet\_uint()$  { $G$  is non-deterministic variable}
9:    $T \leftarrow nondet\_uint()$  { $T$  is non-deterministic variable}
10:  assume ( $G_{min} \leq G \leq G_{max}$ ) {restricting the values of  $G$ }
11:  assume ( $T_{min} \leq T \leq T_{max}$ ) {restricting the values of  $T$ }
12:   $I_{max}, V_{max} \leftarrow PVgenerationMODEL(G, T)$ 
   {Now, testing if battery is empty:}
13:  if ( $SOC \leq SOC_{limit}$ ) then
14:    assert (PV panel is generating energy) {if there is no enough solar energy generation, then FAIL}
15:     $house \leftarrow energy\ from\ PV\ panels$ 
16:     $battery \leftarrow energy\ from\ PV\ panels$  {bulk stage}
17:     $SOC \leftarrow SOC + 1\text{ h charge}$ 
18:  else if ( $PV\ Array \geq V_{bulk}$ ) then
19:    depending on the  $SOC$ , decide if Battery enter in absorption or floating stage
20:    adequate the voltage at the DC-bus (PV panel feed the bus)
21:     $house \leftarrow energy\ from\ PV\ panels$ 
22:     $battery \leftarrow energy\ from\ PV\ panels$  {bulk or floating stage}
23:     $SOC \leftarrow SOC + 1\text{ h charge}$ 
24:  else
25:    Adequate the voltage at the DC-bus (battery feed the bus)
26:     $house \leftarrow energy\ from\ batteries$ 
27:     $SOC \leftarrow SOC - 1\text{ h discharge}$ 
    {the last case: batteries are charged but the panels are not generating}
28:  end if
29:   $h \leftarrow (h + 1)$ 
30: end for
   {now enter again at the 16:00h-08:00h period}
31:  $SOC \leftarrow SOC - 16\text{ h discharge}$ 
32: assert ( $SOC \geq SOC_{limit}$ )
   {Finally, perform the same 8h for-LOOP as above}
   {Batteries must conclude the 48h with enough capacity}
33: assert ( $SOC \geq SOC_{limit}$ )
34: return ( )

```

---



**Algorithm 2** Algorithm for the 1,200 W - 6 h PV system

---

```

1: Perform project sizing validation( )
2: if (FAIL validation) then
3:   exit ("Project sizing error")
4: end if
5:  $SOC \leftarrow 80\%$ 
   {First the 08:00h-16:00h automated verification}
   LOOP Process
6: for  $h = 1$  to 8 do
7:    $G \leftarrow nondet\_uint()$  { $G$  is non-deterministic variable}
8:    $T \leftarrow nondet\_uint()$  { $T$  is non-deterministic variable}
9:   assume ( $Gmin \leq G \leq Gmax$ ) {restricting the values
    of  $G$ }
10:  assume ( $Tmin \leq T \leq Tmax$ ) {restricting the values
    of  $T$ }
11:   $Imax, Vmax \leftarrow PVgenerationMODEL(G, T)$ 
    {Now, testing if battery is empty:}
12:  if ( $SOC \leq SOC_{limit}$ ) then
13:    assert (PV panel is generating energy) {if there is no
    enough solar energy generation, then FAIL}
14:     $house \leftarrow energy\ from\ PV\ panels$ 
15:     $battery \leftarrow energy\ from\ PV\ panels$  {bulk stage}
16:     $SOC \leftarrow SOC + 1\ h\ charge$ 
17:  else if ( $PV\ Array \geq Vbulk$ ) then
18:    depending on the  $SOC$ , decide if Battery enter in
    absorption or floating stage
19:    adequate the voltage at the DC-bus (PV panel feed
    the bus)
20:     $house \leftarrow energy\ from\ PV\ panels$ 
21:     $battery \leftarrow energy\ from\ PV\ panels$  {bulk or float-
    ing stage}
22:     $SOC \leftarrow SOC + 1\ h\ charge$ 
23:  else
24:    Adequate the voltage at the DC-bus (battery feed the
    bus)
25:     $house \leftarrow energy\ from\ batteries$ 
26:     $SOC \leftarrow SOC - 1\ h\ discharge$ 
    {the last case: batteries are charged but the panels
    are not generating}
27:  end if
28:   $h \leftarrow (h + 1)$ 
29: end for
    {now 6h autonomy from 16:00h-22:00h period}
30:  $SOC \leftarrow SOC - 6\ h\ discharge$ 
31: assert ( $SOC \geq SOC_{limit}$ )
32: return ( )

```

---

division by zero, but unnecessary to check at this kind of problem (if not removed, there is lost of performance during the automated verification);

- The parameter *unmind* tells to ESBMC the limit to unroll the loops. This number was optimized (empirically) in order to reduce the running time and avoid to unwind unnecessarily the loops;
- The two parameters with *symex* tell to ESBMC to perform an incremental SMT solving. There are other options, but this parameter is necessary because the complexity of the algorithms. The incremental SMT solving uses few RAM memory, compared with other SMT solving. During empirical tests of the algorithms, the incremental solving was the only one who do not demanded 100% the RAM memory. The use of swap-memory, i.e., the use of hard disk, reduces the performance and must be avoided;
- And the last parameter says to the tool that the Z3 SMT solver will be used.

**B. Experimental results**

XXX

**VIII. RELATED WORK**

The CBMC, it is notable its prominence through international awards received in competitions. In the works related to the CBMC, much of the research is focused on the improvement of the tool (as software and hardware verifier) and its performance in the competitions, without addressing, however, the question of PV system validation, as can be seen in [57]. For example, in [67], CBMC received an extension to perform verification of embedded software.

ESBMC was developed to perform bounded model checks on both sequential and concurrent programs using a range of SMT solvers, and has a proven track of bug finding in real-world applications [68], [59]. The tool also implements a technique to prove the correctness of (some) unbounded programs, the k-induction algorithm; this approach has been applied to a large number of benchmarks and has produced more correct results than similar competing tools [58], [69]. [11] show that is possible to perform the partitioning of hardware-software (optimization) using ESBMC as well. [70] show that is possible to use ESBMC to check the stability of digital control systems with uncertainty. In addition, [71] show the approach to synthesize safe digital feedback controllers for physical plants represented as linear, time invariant models. However, the use of ESBMC to verify PV systems is unprecedented.

Concerning modeling and simulation of electrical energy systems, [72] used an agent-base model approach to evaluate the effect of loads (refrigerators mainly), shedding, grids, wind power generation, and smart grids, in a dynamic integral multi-scale case study. However, that is a software development approach for the simulation of complex systems, without the formalism of automated verification.

In the other hand, the use of formal verification related to modeling and validation or even simulation of power systems

or power networks is a young theme of research. According [73], where articles from seminars were issued at this thematic, the actual focus is on optimization and control of microgrids, HVAC systems (diagnostics and maintenance), models of generic wind turbine generation, modeling of transmission networks, and verification of smart grids. In addition, according with [74], recent research has attempted to formalize and implement a formal study of large population of PV panels (very common in European Union). The focus have been the modeling of the dynamics of PV panels and their interaction with the power grid. Those models do not use battery array, for example.

Therefore, the present work unites the versatility and power of the formal models, applied to a formal verification tool, with individual PV systems (off-grid), which is the most indicated and common option to solve the problem of energy in developing countries. Especially those that have isolated communities, either for long distances of the great centers, or for being riverside. This work includes, based on its off-grid characteristic, batteries and charge controllers, differently of on-grid systems of related work.

## IX. CONCLUSIONS AND FUTURE WORK

The proposed framework seems to be a coherent and promising tool to perform verification of a PV project. The next step is the deployment of the solution at the ESBMC environment, and compare the results with a real PV project deployed at the field.

The framework presented here establishes the automated verification and the project sizing check. However, it is possible to perform the optimum sizing of the PV system. Moreover, the authors target the use of synthesis to perform the optimum sizing. That is an interesting issue, considering that nowadays is possible to choose from many different manufacturers and many different models of equipment of a SPV system, the idea of just to validate one given solutions is not enough. Probably is more useful to design a tool which starts from a list of commercial equipments, i.e., inputs, where each equipment is verified during synthesis phase and the final solution, which satisfies the specifications (or constraints) of the project, is found from the inputs, automatically. This is what a Counterexample Guided-Inductive Synthesis (CEGIS) does.

In the future, it is possible to expand the type of renewable used for power generation. Allowing the verification of wind generation or even hybrid systems, which can transform the tool into an important instrument to complement the solar PV system design process, ensuring that it does not fail and meets the expected requirements.

Plus (need to think about it): verification in every-hour of a 48h cycle?, PV model math model can be improved, charge/discharge of batteries can be improved (is a simplified model), house's power consumption is not flat or constant, the dweller can buy more home appliances, synthesis need a data base of components, to test peak (surge) consumption (above the nominal), additional models for on-grid/off-grid farm, hybrid, and on-grid with batteries).

## ACKNOWLEDGMENT

The authors would like to thank Newton Fund (Star Energy project), Coventry University, and Sustainable Amazonas Foundation (FAS), for the financial support and for the possibility to test the designed model with real data from solar photovoltaic system, deployed in June of 2018 at a small community at Cuieira River, distant around 1.5 hour by speedboat from Manaus city. The authors would like to thank, as well, to FAPEAM (Amazonas State Foundation for Research Support) for the financial support to the research.

## REFERENCES

- [1] S. Coelho, A. Sanches-Pereira, L. Tudeschini, J. Escobar, M. Poveda, N. Coluna, A. Collin, E. L. Rovere, A. Trindade, and O. Pereira, "Biomass residues as electricity generation source in low hd source in regions of brazil," in *The XI Latin American Congress of Electricity Generation and Transmission CLAGTEE*, UNESP, Ed., 2015, pp. 1–8.
- [2] S. Karekesi, K. Lata, and S. Coelho, *Renewable Energy - A Global Review of Technologies, Policies and Markets*. London: Earthscan, 2006, ch. Traditional Biomass Energy: Improving Its Use and Moving to Modern Energy Use, pp. 231–261.
- [3] EPIA, *Global market outlook for photovoltaics 2014-2018*. Belgium: European Photovoltaic Industry Association, 2014.
- [4] H. Yatimi and E. Aroudam, "Modeling and simulation of a stand-alone photovoltaic system," in *Xth International Conference on Integrated Design and Production*, T. Marocco, Ed., 2015.
- [5] R. Noroozian and G. Gharehpetian, "An investigation on combined operation of active power filter with photovoltaic arrays," *International Journal of Electrical Power & Energy Systems*, vol. 46, pp. 392–399, 2013.
- [6] SEIA, "Solar energy," <http://www.seia.org/about/solar-energy>, accessed 1st October, 2016.
- [7] A. Chauhan and R. Saini, "Renewable energy based off-grid rural electrification in uttarakhand state of india: Technology options, modelling method, barriers and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 662–681, 2015.
- [8] R. Rawat, S. Kaushik, and R. Lamba, "A review on modeling, design methodology and size otimization of photovoltaic based motor pumping, standalone and grid connectec system," *Renewable and Sustainable Energy Reviews*, vol. 57, pp. 1506–1519, 2016.
- [9] W. Brooks and J. Dunlop, *NABCEP: PV Installation Professional Resource Guide*. Clifton Park, NY: NABCEP, 2013.
- [10] C. Baier and J.-P. Katoen, *Principles of model checking*. London: The MIT Press, 2008.
- [11] A. Trindade and L. Cordeiro, "Applying smt-based verification to hardware/software partitioning in embedded systems," *Design Automation for Embedded Systems (DAES)*, vol. 20, no. 1, pp. 1–19, 2016.
- [12] E. Clarke, E. Emerson, and J. Sifakis, "Model checking: algorithmic verification and debugging," *Communications of the ACM*, vol. 51, no. 11, pp. 74–84, 2009.
- [13] T. Roy, "Simulation and analysis of photovoltaic stand- alone systems," Technology in Electrical Engineering, Department of Electrical Engineering National Institute of Technology, Rourkela, Odisha, India, 2013.
- [14] R. Gules, J. Pacheco, H. Hey, and J. Imhoff, "A maximum power point tracking system with parallel connection for pv stand alone applications," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 7, pp. 2674–2683, 2008.
- [15] P. Mohanty, K. Sharma, M. Gujar, M. Kolhe, and A. Azmi, *Solar Photovoltaic System Applications*. Springer International Publishing, 2016, ch. PV System Design for Off-Grid Applications, pp. 49–83.
- [16] J. Pinho and M. Galdino, *Manual de Engenharia para Sistemas Fotovoltaicos*. Rio de Janeiro/RJ: CEPTEL CRESESB, 2014.
- [17] S. Kim, "Sliding mode controller for the single-phase grid connected photovoltaic system," *Applied Energy*, vol. 83, no. 10, pp. 1101–1115, 2006.
- [18] J. Freeman, J. Whitmore, N. Blair, and A. Dobos, "Validation of multiple tools for flat plate photovoltaic modeling against measured data," National Renewable Energy Laboratory, Colorado, Tech. Rep., 2014.
- [19] A. P. Dobos, "Pvwatts version 5 manual," National Renewable Energy Laboratory, Colorado, Tech. Rep., 2014.

- [20] N. Blair, A. Dobos, J. Freeman, T. Neises, and M. Wagner, "System advisor model, sam 2014.1.14: General description," National Renewable Energy Laboratory, Colorado, Tech. Rep., 2014.
- [21] C. Cameron, W. Boyson, and D. Riley, "Comparison of pv system performance-model predictions with measured pv system performance," in *33rd IEEE Photovoltaic Specialists Conference*, 2008, pp. 1–6.
- [22] HOMER, "The homer microgrid software," <http://www.homerenergy.com/software.html>, 1st October 2016.
- [23] N. Swarnkar, L. Gidwani, and R. Sharma, "An application of homer pro in optimization of hybrid energy system for electrification of technical institute," in *International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*, 2016, pp. 56–61.
- [24] S. Pradhan, S. Singh, M. Choudhury, and D. Dwivedy, "Study of cost analysis and emission analysis for grid connected pv systems using retscreen 4 simulation software," *International Journal of Engineering Research & Technology*, vol. 4, no. 4, pp. 203–207, 2015.
- [25] A. Mills and S. Al-Hallaj, "Simulation of hydrogen-based hybrid systems using hybrid2," *International Journal of Hydrogen Energy*, vol. 29, no. 10, pp. 991–999, 2004.
- [26] U. of Massachusetts Amherst UMASS, "Hybrid2," <http://www.umass.edu/windenergy/research/topics/tools/software/hybrid2>, 1st October 2016.
- [27] A. Hansen, P. Sørensen, L. Hansen, and H. Bindner, *Models for a stand-alone PV system*. Forskningscenter Risoe, 2001, no. 1219.
- [28] E. Lorenzo, *Solar Electricity Engineering of Photovoltaic Systems*. Spain: Artes Gráficas S.L., 1994.
- [29] SamlexSolar.com, "Solar (pv) cell, module, array," <http://www.samlexsolar.com/learning-center/solar-cell-module-array.aspx>, accessed 28th December, 2017.
- [30] A. Jakhani, S. Samo, S. Kamboh, J. Labadin, and A. Rigit, "An improved mathematical model for computing power output of solar photovoltaic modules," *International Journal of Photoenergy*, vol. 2014, no. ID 346704, p. 9, 2014.
- [31] W. Durisch, D. Tille, A. Worz, and W. Plapp, "Characterisation of photovoltaic generators," *Applied Energy*, vol. 65, no. 1–4, pp. 273–284, 2000.
- [32] S. Rajanna and R. Saini, "Modeling of integrated renewable energy system for electrification of a remote area in india," *Renewable Energy*, vol. 90, no. C, pp. 175–187, 2016.
- [33] M. Badejani, M. Masoum, and M. Kalanta, "Power engineering conference," in *AUPEC 2007*, A. Universities, Ed., 2007, pp. 1–6.
- [34] S. Ferrari, M. L. V. Piuri, A. Salman, L. Cristaldi, M. Faifer, and S. Toscani, "Solar panel modelling through computational intelligence techniques," *Measurement: Journal of the International Measurement Confederation*, no. 93, pp. 572–580, 2016.
- [35] E. Saloux, A. Teyssedou, and M. Sorin, "Explicit model of photovoltaic panels to determine voltages and currents at the maximum power point," *Solar Energy*, vol. 85, no. 5, pp. 713–722, 2011.
- [36] M. Hasan and S. Parida, "An overview of solar photovoltaic panel modeling based on analytical and experimental viewpoint," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 75–83, 2016.
- [37] D. King, W. Boyson, and J. Kratochvill, *Photovoltaic Array Performance Model*. Albuquerque, New Mexico: Sandia National Laboratories, 2004.
- [38] A. Mellit, M. Benghanem, and S. Kalogirou, "Modeling and simulation of a stand-alone photovoltaic system using an adaptive artificial neural network: Proposition for a new sizing procedure," *Renewable Energy*, vol. 32, no. 2, pp. 285–313, 2007.
- [39] J. Cubas, S. Pindado, and F. Sorribes-Palmer, "Analytical calculation of photovoltaic systems maximum power point (mpp) based on the operation point," *Applied Sciences*, vol. 7, no. 9, pp. 870–884, 2017. [Online]. Available: <http://www.mdpi.com/2076-3417/7/9/870>
- [40] V. Brano, A. Orioli, G. Ciulla, and A. Gangi, "Modeling , design and simulation of stand-alone photovoltaic power systems with battery storage," *Solar Energy Materials & Solar Cells*, vol. 94, pp. 1358–1370, 2010.
- [41] E. Shenawy, O. Esmail, A. Elbaset, and H. Hamed, "Practical identification of photovoltaic module parameters," *ISESCO Journal of Science and Technology*, vol. 11, no. 19, pp. 66–71, 2015.
- [42] H. Tian, F. Mancilla-David, K. Ellis, E. Muljadic, and J. Jenkins, "A detailed performance model for photovoltaic systems," National Renewable Energy Laboratory, Colorado, Tech. Rep., 2012.
- [43] M. Villalva, J. Gazoli, and E. Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 24, pp. 1198–1208, 2009.
- [44] R. Ross, "Flat-plate photovoltaic array design optimization," in *14th IEEE Photovoltaic Specialists Conference*, C. San Diego, Ed., 1980, pp. 1126–1132.
- [45] J. Copetti, E. Lorenzo, and F. Chenlo, "A general battery model for pv system simulation," *Progress in Photovoltaics: Research and Applications*, vol. 1, no. 4, pp. 283–292, 1993.
- [46] Y. Sukamongkol, S. Chungpaibulpatana, and W. Ongsakul, "A simulation model for predicting the performance of a solar photovoltaic system with alternating current loads," *Renewable Energy*, vol. 27, no. 2, pp. 237–258, 2002.
- [47] J. Manwell and J. McGowan, "Lead acid battery storage model for hybrid energy systems," *Solar Energy*, vol. 50, no. 5, pp. 399–405, 1993.
- [48] —, "Extension of the kinetic battery model for wind/hybrid power systems," in *5th European Wind Energy Association Conference*, 1994, pp. 284–289.
- [49] S. Wenham, M. Green, and M. Watt, *Applied Photovoltaics*. Australia: Center for Photovoltaic Devices and Systems, 1994.
- [50] J. Abdulateef, "Simulation of solar off- grid photovoltaic system for residential unit," *International Journal of Sustainable and Green Energy. Special Issue: Engineering Solution for High Performance of Solar Energy System*, vol. 4, no. 3–1, pp. 29–33, 2014.
- [51] P. Mahanta, K. Debnath, and M. Rahman, "Modeling and simulation of a pv module based power system using matlab/simulink," *Dhaka University Journal of Science*, vol. 62, no. 2, pp. 127–132, 2014.
- [52] A. Dhanowa and V. Garg, "Modeling and simulation of an off grid pv system for with battery backup for remote and rural area network," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 06, pp. 915–918, 2015.
- [53] D. Catherine and K. Bhaskar, "Simulation of a solar mppt charger using cuk converter for standalone application," *International Journal of Electrical, Electronics and Computer Systems (IJECS)*, vol. 1, no. 1, 2013.
- [54] A. Haque, "Maximum power point tracking (mppt) scheme for solar photovoltaic system," *Energy Technology & Policy*, vol. 1, no. 1, pp. 115–122, 2014.
- [55] E. Clarke, *25 Years of Model Checking*. Springer, 2008, ch. The birth of model checking, pp. 1–26.
- [56] F. Forejt, M. Kwiatkowska, G. Norman, and D. Parker, *Automated Verification Techniques for Probabilistic Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 53–113.
- [57] D. Kroening and M. Tautschnig, "Cbmc c bounded model checker (competition contribution)," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol. LNCS 8413, 2014, pp. 389–391.
- [58] M. Gadelha, L. Cordeiro, and D. Nicole, "Encoding floating-points using the smt theory in esbmc: An empirical evaluation over the sv-comp benchmarks," in *20th Brazilian Symposium on Formal Methods (SBMF)*, vol. LNCS 10623, 2017, pp. 91–106.
- [59] M. Ramalho, M. Freitas, F. Sousa, H. Marques, L. Cordeiro, and B. Fischer, "Smt-based bounded model checking of c++ programs," in *20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, 2013, pp. 147–156.
- [60] R. Brummayer and A. Biere, "Boolector: An efficient smt solver for bit-vectors and arrays," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol. LNCS 5505, 2009, pp. 174–177.
- [61] L. D. Moura and N. Björner, "Z3: An efficient smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol. LNCS 4963, 2008, pp. 337–340.
- [62] A. Cimatti, A. Griggio, B. Schaafsma, and R. Sebastiani, "The mathsat5 smt solver," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, vol. LNCS 7795, 2013, pp. 93–107.
- [63] B. Dutertre, "Yices 2.2," in *Computer Aided Verification (CAV)*, A. Biere and R. Bloem, Eds., vol. LNCS 8559, 2014, pp. 734–744.
- [64] C. Barrett, C. Conway, M. Deters, L. Hadarean, D. Jovanovic, T. King, A. Reynolds, and C. Tinelli, "Cvc4," in *Computer Aided Verification*, vol. LNCS 6806, 2011, pp. 171–177.
- [65] UC, "Cyber-physical systems," <http://cyberphysicalsystems.org/>, accessed 1st October, 2016.
- [66] R. Alur, R. Bodik, G. Juniwal, M. Martin, M. Raghothaman, S. Seshia, R. Singh, A. Solar-Lezama, E. Torlak, and A. Udupa, in *2013 Formal Methods in Computer Aided Design*, 2013, pp. 1–17.
- [67] P. Schrammel, D. Kroening, M. Brain, R. Martins, T. Teige, and T. Biemiller, "Incremental bounded model checking for embedded software," *Formal Aspects of Computing*, vol. 29, no. 5, pp. 911–931, 2017. [Online]. Available: <https://doi.org/10.1007/s00165-017-0419-1>
- [68] L. Cordeiro, B. Fischer, and J. Marques-Silva, "Smt-based bounded model checking for embedded ansi-c software," *IEEE Transactions on Software Engineering*, vol. 38, no. 4, pp. 957–974, 2012.
- [69] M. Gadelha, H. Ismail, and L. Cordeiro, "Handling loops in bounded model checking of c programs via k-induction," *International Journal*

- on *Software Tools for Technology Transfer STTT*, vol. 19, no. 1, pp. 97–114, 2017.
- [70] I. Bessa, H. Ismail, R. Palhares, L. Cordeiro, and J. C. Filho, “Formal nonfragile stability verification of digital control systems with uncertainty,” *In IEEE Transactions on Computers*, vol. 66, no. 3, pp. 545–552, 2017.
  - [71] A. Abate, I. Bessa, D. Cattaruzza, L. Cordeiro, C. David, P. Kesseli, D. Kroening, and E. Polgreen, “Automated formal synthesis of digital controllers for state-space physical plants,” in *Computer Aided Verification*, vol. LNCS 10426, 2017, pp. 462–482.
  - [72] E. Kremers, *Modelling and Simulation of Electrical Systems through a Complex Systems Approach using Agent-Based Models*. Karlsruhe - Germany: KIT Scientific Publishing, Karlsruhe, 2013.
  - [73] A. Abate, M. Frnzle, I. Hiskens, and M. Strelec, “Modeling, verification, and control of complex systems for energy networks,” *Dagstuhl Reports*, vol. 4, no. 10, pp. 69–97, 2014.
  - [74] A. Abate, “Verification of networks of smart energy systems over the cloud,” in *Numerical Software Verification*, S. Bogomolov, M. Martel, and P. Prabhakar, Eds., vol. LNCS 10152, 2017, pp. 1–14.

**Alessandro Trindade** Alessandro Bezerra Trindade received his BSc and MSc in Electrical Engineering from the Federal University of Amazonas (UFAM) in 1995 and 2015, respectively. Currently, he is pursuing his PhD in Informatics at the UFAM, and hold a position as Assistant Professor at the Electricity Department from the Federal University of Amazonas.

**Lucas Cordeiro** Lucas Carvalho Cordeiro received his PhD degree from the University of Southampton in 2011. From 2009 to 2016, he was an Adjunct Professor at the UFAM (he is currently in unpaid leave). Since 2018, he is a Senior Lecturer at the University of Manchester, working with the Formal Methods Group.