**Instituto Tecnológico de Costa Rica**

**Maestría en Ciencia y Tecnología para la Sostenibilidad**

DOCINADE
Maestría en Ciencia y Tecnología para la Sostenibilidad

# Implementation of a model for isolated microgrids with photovoltaic energy sources through emulation in an application-specific specific instruction-set processor (ASIP)

Eng. Mario Alberto Araya Carrillo

Tutor: Dr. Carlos Meza Benavidez

**2023**

DOCINADE
Maestría en Ciencia y Tecnología para la Sostenibilidad

Eje curricular de Energía renovable

Tesis de Maestría

# Implementation of a model for isolated microgrids with photovoltaic energy sources through emulation in an application-specific specific instruction-set processor (ASIP)

Candidato

Eng. Mario Alberto Araya Carrillo

Asesor

Dr. Alfonso Chacón Rodríguez

Tutor

Dr. Carlos Meza Benavidez

Setiembre 2023

Maestría en Ciencia y Tecnología para la Sostenibilidad

Eje curricular de Energía Renovable

# Implementation of a model for isolated microgrids with photovoltaic energy sources through emulation in an application-specific specific instruction-set processor (ASIP)

Work submitted to the Evaluation Tribunal as a requirement for applying for the degree of Master in Science and Technology for Sustainability, under the thematic axis of Renewable Energy

Dra. Ingrid Varela Benavides,

Master's Program Manager, who presides

Firmado digitalmente por
INGRID LORENA VARELA
BENAVIDES (FIRMA)
Fecha: 2023.09.22
08:07:24 -06'00'

Dr. Rolando Madriz Vargas,

Academic Area Coordinating Representative

Dr. Víctor Granados Fernández,

Thematic area representative

Dr. Carlos Meza Benavidez

Thesis Tutor

Dr. Alfonso Chacón Rodríguez

Thesis Advisor

# Table of contents

## Declaration of authenticity

I, the undersigned, Mario Alberto Araya Carrillo, student of the Maestría en Ciencia y Tecnología para la Sostenibilidad, declare that the content of this thesis is my original work and that all the assistance received in preparing this thesis and sources have been acknowledged.

This thesis has not been submitted for any degree or other purposes and objectives of this study and has not been previously submitted to any other university for a higher degree.

Furthermore, I confirm that, I have referenced all sources used in the work and that all data and findings in the work have not been falsified or modified.

## Acknowledgments

Firstly, I would like to thank my tutor Carlos Meza Benavides for all the support and accompaniment provided in the process of developing this thesis project, as well as his guidance throughout the development of the master's program.

I thank my advisor Alfonso Chacón Rodríguez for making available the required DCILab equipment. I am also grateful for his advice during the implementation stage of the project.

I would also like to thank Edgar Brenes Gutiérrez for providing help in managing the hardware used.

Finally, my sincere thanks to Carlos Salazar García, part of the DCILab team. Thanks to his help and expertise the project could be implemented with successfully.

## Dedication

To my father, my grandmother, and my uncle. As a proof that those you left behind will keep moving forward.

# Abbreviations

| Abbreviation | Description |
|---|---|
| AC | Alternating Current |
| DC | Direct Current |
| MG | Microgrid |
| ASIP | Application Specific Instruction-Set Processor |
| PV | Photovoltaic |
| DER | Distributed Energy Resources |
| DG | Distributed Generators |
| SS | Storage System |
| MOSFET | Metal-Oxide-Semiconductor Field-Effect Transistor |
| FPGA | Field Programmable Gate Array |
| RTL | Register Transfer Level |
| HDL | Hardware Description Language |
| HLS | High-Level Synthesis |

# Introduction.

## 1. **Justification**

Worldwide, the distribution of electrical power has traditionally been centralized in a location far from the point of use. This reality is the result of an assortment of technical, economic, and political factors from the era of the War of Currents at the end of the 19th century, where alternating current systems were victorious and the generation paradigm that dominates to date was established (McNichol, 2011). Power generation has been mostly dependent on the use of hydrocarbons as a source of energy and on a global scale oil, coal and natural gas represent the largest share of energy production, representing 81% of the total energy consumed and producing approximately 31575 Mt (megatons) of carbon dioxide, during 2012 (IEA, 2014). In Costa Rica the emissions for that same year were 11.8 Mt to CO2eq according to (MINAE, 2019).

According to the Millennium Development Goals Report 2015, global emissions of carbon dioxide have increased by over 50% since 1990, contributing to the impacts of climate change, such as an increase in extreme weather conditions, altered ecosystems and all the socioeconomic effects derived from these factors, such as mass migrations (United Nations, 2016).

Costa Rica has committed to become a decarbonized economy with net-zero emissions by 2050 in its "National Decarbonization Plan 2018-2050" following the Paris Agreement. This includes a goal of an energy mix with a 100% renewable sources (MINAE 2018), considering actions such as increasing the market penetration of distributed generation.

Microgrids emerge as an interesting solution that can help meet the country's emissions targets. Microgrids are systems that have attracted interest in recent years due to their potential advantages compared to centralized power generation, distribution, and consumption systems. Among the benefits of microgrids are their flexibility to operate as direct current (DC), alternating current (AC), or hybrid systems (AC/DC), thus facilitating the coupling of all types of energy sources with emphasis on clean and renewable sources (Zhang et al., 2013; Planas et al., 2015).

Microgrids represent a paradigm shift towards distributed generation in which renewable energies play a major role. They are considered to have the capacity to improve the reliability, robustness, quality, and efficiency of energy use, in addition to reducing dependence on fossil fuels (Jung & Villaran, 2017).

Additionally, microgrids are of interest for their implementation in rural areas and areas far from generation plants. They allow the reduction of power losses in transmission lines, can improve the quality of energy, and allow a more heterogeneous availability of energy sources (Phurailatpam, 2015). However, there are still problems to be solved before they can be widely used (Unamuno & Barrena, 2015). Microgrids are highly dynamic systems on different time scales and given the different energy management and control tasks, and multiple electrical and environmental variables, research is still required for them to be adopted on a larger scale (Olivares et al., 2014). This is especially true for DC microgrids. Due to the hegemony of centralized AC systems, more research is still required in the subject of isolated microgrids, especially those focused on DC, regarding their modeling and standardization. Questions such as what the optimal voltage levels for power distribution buses are, what type of control strategy (centralized or distributed, short- or long-term prediction) facilitates a more efficient use and higher availability of energy among others, still require more investigation.

However, finding answers to these questions by implementing and testing physical microgrids can be expensive and time consuming due to the iterative nature of systems design and optimization. A cost-effective alternative to analyze and optimize microgrid design and operation is through numerical simulations of microgrid models. Previous works (Hu et al., 2017; Khadepaun et al., 2020) use simulations to optimize the design process of stand-alone microgrids focusing on their steady-state operation.

This approach makes more notable another set of issues regarding microgrid data handling and processing, both during system simulation and operation. Microgrids produce a great amount of data which can require a Big Data approach for the design of stability improvement control algorithms, asset management, and renewable energy prediction (Moharm, 2019). The great amount of data throughput of microgrids also increases the time it takes to simulate these systems, requiring for specialized equipment if time is a constraining factor in the design process.

Considering the previously described context, this research was divided in two areas, where the first one proposes the design of a microgrid simulation program that can be utilized as a tool for testing different architectures, control, and energy management algorithms to aid in the process of microgrid design and optimization. This research is encapsulated within the scope of stand-alone DC microgrids with photovoltaic energy sources as a first approach to the simulation of this kind of systems, and due to the association to the Laboratory of Electronic Systems for Sustainability (SESLab) of the School of Electronic Engineering at the Instituto Tecnológico de Costa Rica (ITCR).

The second area covered by this research is the processing of the large amount of data that microgrids generate, specifically in simulation. For this purpose, the project is associated with a larger project entitled "Design of multicore architectures for massive data processing (Big Data)" developed in the Integrated Circuit Design Laboratory (DCILab) at ITCR and proposes the implementation of an application specific instruction-set processor (ASIP) for the optimization of simulation program to accelerate simulation time, with an ideal goal of achieving real-time simulations.

# 2. Theoretical framework

## 2.1. New energy systems for sustainable development

After the release of the 1987 report of the World Commission on Environment and Development, titled “Our Common Future”, the concept of “sustainable development” began to gather attention on a global scale. It is defined as “development that meets the needs of the present without compromising the ability of future generations to meet their own needs”, considering the limitations imposed by the state of technology and social organization on the environment’s ability to meet present and future needs (United Nations, 1987).

Since then, the concept has expanded to cover multiple areas of human, societal and technological development, culminating in the establishment of the “2030 Agenda for Sustainable Development” adopted by all United Nations member states in 2015. The agenda proposed 17 goals, out of which, Goal 7 looks to 1) ensure universal access to affordable, reliable, and modern energy services, 2) increase substantially the share of renewable energy in the global energy mix, and 3) expand infrastructure and upgrade technology for supplying modern and sustainable energy services (United Nations, 2018).

Renewable technologies have advanced considerably since the 1987 report, as well as cost have decreased, making their adoption more attractive both at governmental institutions and private enterprises. For example, the global energy share of renewables went from 6.4% in 1990 to 11.4% in 2019 (Ritchie & Roser, 2020). More specifically, photovoltaic generation systems have become a viable and promising source. During the period of 2010-2019 the cost of photovoltaics has declined 82% and residential PV systems are as much as two-thirds cheaper (IRENA, 2020, p.11). Due to the reduction in cost and government incentives solar photovoltaics account for 24.3% of the global renewable energy generation or about 710 TW of installed capacity.

Different approaches to energy production, distribution and consumption have also been developed since. Improvements in computing systems have allowed for more complex and efficient energy systems such as virtual power plants which enhance the management of the aggregation of distributed resources, both energy generation and demand (Naval & Yusta, 2021), and smart grids defined as electricity networks that can integrate the behavior and actions of the users (generators, loads, storage, etc.) connected to it intelligently in order to supply sustainable, affordable energy and secure its availability; i.e. they present characteristics such as self-healing from power disturbance events, allow for the active participation by users in demand response, accommodates all types of generation and storage options, enables new products, services, and markets, and perform asset optimization to improve efficiency (Clastres, 2011). On the same vein as smart-grids, smaller scale systems that have a growing interest and research demand are microgrids which are the topic of this thesis.

## 2.2. **Microgrid Characterization**

Microgrids are defined as “a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island mode” (Ton & Smith, 2012).

Microgrids are complex systems with multiple elements or subsystems interacting on different levels that cover areas going from electrical, physical, environmental, all the way up to legal and regulatory elements. Table 1 shows a basic list of the elements that conform a microgrid.

Table 1. Elements of a microgrid. Elaborated with data from (Planas et al., 2015).

| ELEMENT | FUNCTION | EXAMPLE |
|---|---|---|
| **TRANSMISSION** | In the case of grid-tied microgrids, the high voltage transmission systems can also consider an integral part of the whole system. | |
| **POINTS OF COMMON COUPLING (PCC)** | Interphase elements to connect and disconnect from the mains grid. | • Power converters<br>• Switchgears |
| **DISTRIBUTION** | Distribution lines that interconnect distributed energy resources (DERs) to loads | • Single phase or three phase (AC).<br>• Monopolar, homopolar or bipolar (DC). |
| **PROTECTIONS** | Elements that guarantee the safe operation of the microgrid. Designed with different parameters of sensibility, selectivity, response time and safety level. | • Adaptive protection schemes, digital relays, voltage, harmonic content, current. |
| **MONITORING** | Analysis of system status and all its elements. | • Framework based on service-oriented architecture (SOA). |

| | | |
|---|---|---|
| | | • Installation of universal monitoring, protection, and control units (UMPCUs). |
| **POWER CONVERTERS** | Power conditioning to interconnect distributed generators (DG) with storage systems (SS) and loads. | • Regulators<br>• Inverters<br>• Rectifiers<br>• Transformers |
| **CONTROL** | Units to control the operation and optimization of the previously described elements. Can be implemented with a hierarchical, distributed, or global methodology. | • Grid level: Operation within the energy market. Control of energy availability and demand.<br>• Management level: Microgrid energy production optimization.<br>• Field level: Local controller for each element (DGs, SSs, or loads). |
| **REGULATORY ISSUES** | Set of principles, rules, and incentives to address both technical and economic issues. | • Environmental regulations<br>• Technical codes and standards (IEC)<br>• Load connection restrictions. |
| **ECONOMIC ANALYSIS** | Cost/Benefit study of microgrid production | |

All the elements described will widely vary based on various factors such as the type of microgrid (grid-tied, islanded, stand-alone, or hybrid), its size, and the architecture; that is to say, the distribution of the interconnected elements of generation, power conversion, storage, and loads.

The control systems and algorithms are important as they have different requirements and strategies to perform energy balancing functions depending on whether they are optimizing for maximum energy production, economic benefits, safety factors, or system stability. Both, centralized or decentralized control systems, will have to schedule and dispatch DERs to manage energy import and export between the microgrid and the main grid (Hirsch et al., 2018), and balance the energy demand of all the loads.

This puts forward the question of which control strategy is the best for a given architecture? And underlying is the questions of which is the best microgrid architecture for a given combination of energy sources, power converters and loads?

To answer this, the element of microgrid modelling is introduced, as well as the simulation of said models to determine the optimal parameters for a given architecture and the best control strategy. This is the focus of the present thesis document. The following sections present a mathematical model to describe microgrids, the development of a simulation program for stand-alone microgrids as a first approach to this topic, and finally, the implementation of said program within an embedded system for hardware accelerated simulation.

### 2.3. **Microgrid modeling**

There exist multiple ways to describe the behavior of microgrids which depend on the objective desired for the model. According to (Sen & Kumar, 2018), microgrid modeling is broadly divided in four mayor categories: 1) component-wise modeling were the components of the microgrid are modeled individually and these models are later used to form an aggregated mode, 2) lumped or single entity modeling where the complete microgrid model is obtained, usually in its state-space form, 3) stochastic modeling which can be used as forecasting tools that can be applied to areas of operation, control and planning, and performance optimization, and 4) dynamic equivalence modeling in where the detailed model of a microgrid is substituted with a simplified model having similar dynamic characteristics.

For this research, a component-wise modeling approach was selected as this allows to decompose a microgrid into multiple models of its components which are comparatively simpler than modelling a whole microgrid, as well as facilitating the scalability of the microgrid size by reutilizing the same models for each component with minor modification.

### 2.4. **Microgrid general model**

This thesis focuses on the simulation environment for a microgrid with a bus topology/architecture, which can in turn be characterized as a network. This configuration is the most widely used because it allows simple integration of the different components that make up a microgrid, e.g., power generators, storage, load. The elements of the microgrid are interconnected with each other through buses that can transmit direct current or alternating current as shown in the Figure 1, where, for simplicity, the power processing units, which are located between the sources and the bus and between the loads and the bus, are omitted.

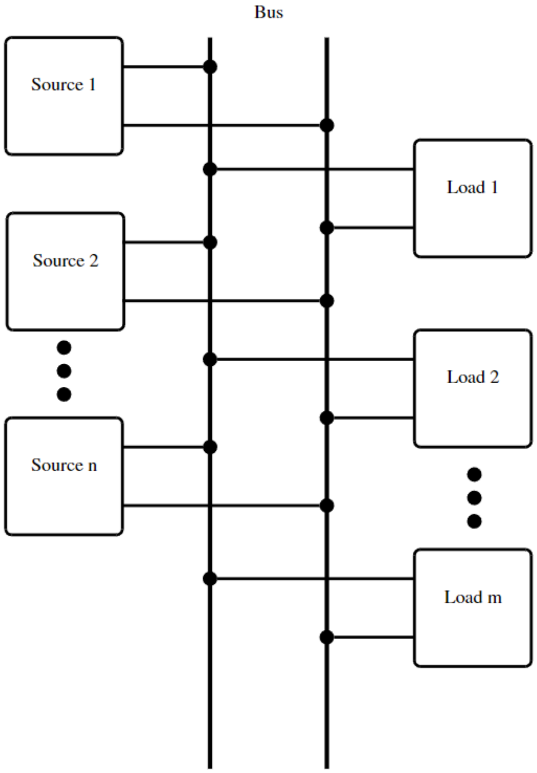

Figure 1. General microgrid architecture.

It is also possible to have configurations of more than one bus such as the one shown in Figure 2.

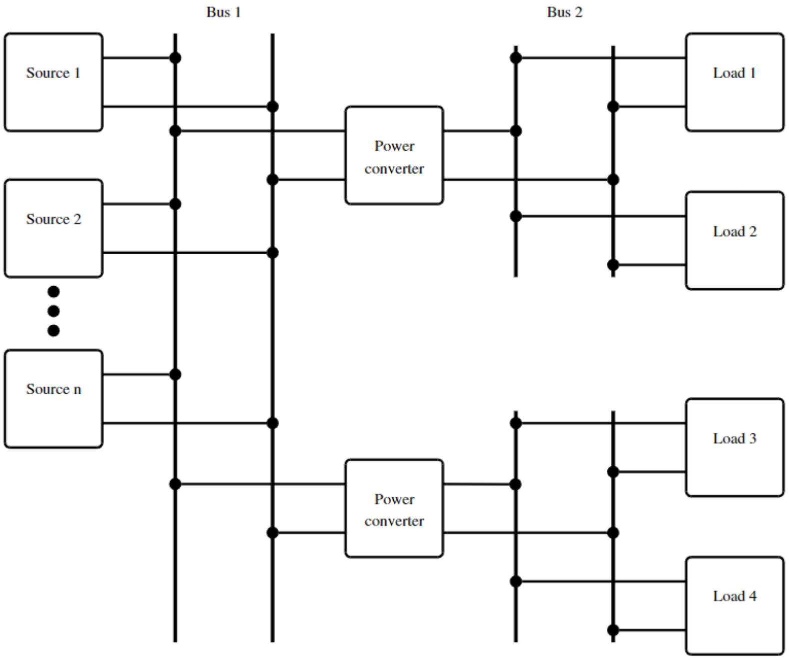

Figure 2. Multi-bus microgrid architecture.

From a mathematical modeling point of view, the incorporation of the buses can be represented by the following set of equations:

$$v_{b_q} = v_{1_{bq}} = v_{2_{bq}} = \dots \tag{1}$$

$$\sum i_{j_{bq}} = 0 \tag{2}$$

Where $v_{b_q}$ represents the voltage on the $\mathrm{q}$ -th bus and $\mathrm{v}_{1_{\mathrm{bj}}}$, $\mathrm{v}_{1_{\mathrm{bj}}}$ , … are the voltages of the devices connected to that bus. Likewise, $\mathrm{i}_{\mathrm{j}_{\mathrm{bq}}}$ represents the current contributed by the $j$-th device connected to the $\mathrm{q}$ -th bus.

In addition to the buses, the model intrinsically considers the equations that describe the components connected to them. However, these are discussed in further sections of this document.

Generally speaking, the modeling was carried out by means of algebraic differential equations described as follows:

$$\frac{dv_i}{dt} = f_i(v_1, v_2, \dots, v_n, i_1, i_2, \dots, i_m, u_k, \rho_i) \tag{3}$$

$$\frac{di_j}{dt} = f_j(v_1, v_2, \dots, v_n, i_1, i_2, \dots, i_m, u_k, \rho_j) \tag{4}$$

Where $\mathrm{v}_\mathrm{i}$and $\mathrm{i}_\mathrm{j}$ represent the system voltages and currents associated with those elements that can store energy, i.e., capacitors and inductors, respectively. Thus, the open-loop microgrid can be represented with $n + m$ first-order differential equations where $n$ is the number of capacitive elements and $m$ is the number of inductive elements. The rates of change of these voltages and currents are determined by a set of functions, $f_i(\cdot)$ and $f_j(\cdot)$, which are related to the network topology and the characteristics of the sources and loads connected to the network. These functions depend on the electrical variables of the system and on a set of parameters $\rho_i = \{\rho_{i_1}, \rho_{i_2}, \cdots\}$ and $\rho_j = \{\rho_{j_1}, \rho_{j_2}, \cdots\}$ specific to the elements that make up the microgrid. It is important to clarify that these functions are discontinuous and for systems using photovoltaic energy they are also nonlinear. For example, the function that determines the current of a photovoltaic generator is shown in the equation 5, where $\mathrm{v}_{\mathrm{pv}}$ is the voltage at the terminals of the generator, and $\Lambda$, $\Psi$ and $\alpha$ are parameters.

$$f_{pv} = \Lambda - \Psi \mathrm{e}^{\alpha \mathrm{v}_{\mathrm{pv}}} \tag{5}$$

The subsystems that make up the microgrid must be managed and controlled in such a way as to ensure energy balance and correct operation. Therefore, in addition to (3) and (4) there is also a set of equations that describe the dynamic performance of the control subsystems, i.e.

$$u_k = g_k\left(\frac{d^{n_u}u_k}{dt}, \frac{d^{n_u-1}u_k}{dt}, \cdots, \frac{du_k}{dt}, \frac{d^{n_v}v_i}{dt}, \frac{d^{n_v-1}v_i}{dt}, \cdots, \frac{dv_i}{dt}, v_i, \frac{d^{n_j}i_j}{dt}, \frac{d^{n_i-1}i_j}{dt}, \cdots, \frac{di_j}{dt}, i_j, \rho_k\right) \qquad (6)$$

Where $\mathrm{u_k}$ represents the $\mathrm{k}$-th control signal for one of the elements of the microgrid that is generated by the control subsystem $\mathrm{k}$ and where $\rho_\mathrm{k}\{\rho_{\mathrm{k}_1}, \rho_{\mathrm{k}_2}, \ldots\}$ are its parameters. Note that, unlike the expressions in $(3)$ and (4), the function $g_k(\cdot)$ can depend on the derivatives of the input voltages, currents, and signals, which means that the expression (6) can be described as a system of $n_u$ first-order differential equations.

The sources, loads and elements interconnecting the buses are managed by a power processing unit which, given their high efficiency, consist of power MOSFET circuits that are switched in such a way that the desired conversion process is achieved. The control signals $u_k$ control these power units and represent duty cycles that are converted to discontinuous signals by means of a pulse width modulator. The discontinuous signals, which have a switching frequency of between 100 kHz up to 1 MHz, are responsible for turning the transistors on and off. This activation process generates a ripple in the currents and voltages of the system that can provide useful information for its management, as indicated in (Meza & Ortega, 2013). If these equations are solved in real-time, then it is possible to visualize and analyze ripple signals at all points of the microgrid.

Nevertheless, the development of digital models of the electrical microgrids described by means of the equations (1), (2), (3), (4) and (6) presents important challenges, which are detailed below:

- The mathematical model of the microgrid can be described by a set of differential equations of order $n + m + n_u$, where even for a system with few generators and loads one can have a high order system. For example, a microgrid with two power sources and two independent loads can be described, in its simplest configuration, by a 6th order differential equation.
- The emulation time of the digital microgrid will depend on the characteristics of the connected generators and loads. For example, for a microgrid based on photovoltaic generators with resistive loads, the emulation time should be at least 10 minutes, to see all the transient processes of the load balance. It is also desirable to perform a 24-to-48-hour analysis to evaluate the load balance on predetermined days.
- Microgrids based on photovoltaic generators can be described by means of a set of first order difference equations and a set of algebraic equations. The number of differential equations depends on the number of elements, i.e., generators and loads, present in the grid. The numerical solution of this set of differential equations must focus on the parallelization of the processes.

- The storage and display of storage system data can represent a bottleneck for the emulation system for the following reasons:
    - There is a difference of several orders of magnitude between the period of the signals that could be displayed and the emulation time. For example, the ripple of the system currents and voltages is between 10 microseconds and 0.01 microseconds, while the emulation time could be from 10 minutes to 48 hours.
    - There are many possible measurement points in the microgrid. For example, each generator and power processing unit set has at least 2 voltage and 2 current values that could be useful to visualize. The same amount can be considered for the loads.

### 2.5. **FPGA based hardware acceleration**

When it comes to simulation of any system, execution time can be a mayor consideration. For microgrids, this is an important resource during the design stages because it can be an iterative process to define and optimize an architecture and control strategies. It is especially important because microgrids operate in multiple time domains which can be time consuming to simulate simultaneously. An improvement in simulation time can facilitate a better and more expedited understanding of the behavior of a microgrid under specific conditions or a range of conditions.

A method by which microgrid simulations have been successfully accelerated is by implementing running the models within a hardware designed specifically for this application. The use of FPGAs has permitted to achieve real-time simulation in a variety of scenarios, methodologies, and objectives such as load behavior studies, network decoupling algorithms and control strategy optimization (Zang et al., 2017; Sheikh et al., 2019; Xu et al., 2020).

Hardware synthesis for an FPGA can be done through two methods. With a Hardware Description language (HDL) such as Verilog or with High Level Synthesis (HLS). HLS is a functional description of a hardware design that is automatically compiled into an RTL implementation within user specified constraints (Oshana, 2012). It allows for a higher level of abstraction from languages such as C, thus reducing implementation time and complexity compared to an HDL implementation. One of the objectives of this thesis is to make a general comparison between a microgrid simulation environment implemented in a general use processor against an application specific processor. Therefore, HLS was selected to minimize development time and to establish a base comparison without delving too deeply into hardware optimization.

# 3. Objectives

## 3.1. General objective

To implement a mathematical model for isolated DC microgrids with photovoltaic energy sources in an emulation system with an application-specific processor.

## 3.2. Specific objectives

1. To model the generation, storage, and load elements of a microgrid for rural electrification and their interactions.
2. To develop an algorithm for the emulation of an isolated microgrid for rural electrification in a high-level programming language.
3. To validate the developed emulation algorithm trough a comparison against a pre-validated circuit simulation software.
4. To compare the efficiency of the algorithm implemented on a general-purpose processor versus an application-specific processor.

## 4. **Methodological framework**

The project consists of experimental research with a quantitative approach. More specifically, this master's thesis is centered around a simulation program delimited within the scope of stand-alone microgrids and further into the research, the implementation of said program in a hardware accelerated processor to measure its performance gains. Given the nature of this project, the research was conducted as a case study, which according to Hernández Sampieri (2014) and Wohlin (2012):

a) can be used in complex, unpredictable, and dynamic environments, under conditions where not all variables are under the control of the researcher,

b) facilitates the comparison of methodologies between works, and

c) can generate qualitative or quantitative conclusions from multiple sources of information.

Wohlin (2012) establishes five systematic stages for the development of a case study which can be defined as such:

1) Design and planning (objective 1): The theoretical framework, data collection methods, and experiments are designed and delimited based on the research objectives and the object of study.
2) Protocols (objective 2): Procedures for the modeling of a microgrid's elements an defining the level of level of complexity for said models.
3) Preparation and data collection (objective 3 & 4): An experiment was applied under simulated conditions with a controlled set of input parameters.
4) Data analysis (objective 3 & 4): Quantitative analysis of the information collected and comparison with benchmark data.
5) Results report (objective 3 & 4): Generation of research conclusions.

Hereinafter the stages of the research are described:

### 4.1. **Design and planning (objective 1)**

A) Literature Review

The project commenced with a literature review of microgrid simulation papers, ranging from general simulation concepts to specific case studies. The review was started with a search of scientific articles in both ScienceDirect and IEEE Xplore databases where logical AND/OR operators were combined with the search keywords 1) microgrid, 2) simulation 3) standalone, 4) photovoltaic and 5) FPGA. The results were limited to the last 5 years, yielding 10,759 and 10,712 articles respectively, before including the FPGA keyword, and

215 and 46 articles after. After sorting by relevance and number of citations, 5 articles were identified as the most relevant.

Older articles, books, and other related documents were included to expand the conceptual framework of the thesis.

B) Data collection methods

With microgrid simulation established as the object of study, it was required to compare the resulting simulation program with an existing microgrid architecture. The benchmark architecture was selected with 3 criteria. It had to be a standalone microgrid, its main source of energy must be photovoltaic generators, and scientific articles referencing said architecture had to be widely cited.

Originally this project was poised to implement a physical microgrid with the selected architecture. However, due to the Covid-19 pandemic, the methodology was reframed such that research results were compared against a pre-validated simulation software as a benchmark.

### 4.2. **Protocols (objective 2)**

The protocols for the development of the simulation program consist of the following considerations:

a) Modeling approach

As discussed previously, there are different modeling approaches when it comes to microgrids. A modeling approach was selected based on the main objective of the research to determine both the complexity and main variables for the simulation program.

b) Model complexity

A bibliographic review was made that established the complexity. The available equipment for simulation and hardware implementation was considered as an additional factor for this determination.

### 4.3. **Preparation and data collection (Objective 3 & 4)**

First-degree data collection was utilized, that is to say, data collected by the researcher with direct contact with the object of study (Wohlin et al., 2012), considering that tests were executed on the developed simulation program with direct control of the input variables and parameters of said tests.

The data generated during tests was digitally archived to achieve dependence on the reliability of the research (Hernández Sampieri et al., 2014). Additionally, the test’s datasets and source code can be accessed through the link given in Araya et al (2022).

### 4.4. **Description of the instrumentation used during the experiment (objective 3 & 4)**

From the protocols, the requirements of the equipment for simulation and data collection were identified.

### 4.5. **Design of the experiment (objective 3 & 4)**

The equipment and nature of the protocols allowed for a controlled experiment. The simulation program that was developed had a controlled input vector which was also fed into the benchmark simulation to generate an output vector for comparison and validation of the simulation model.

This same logic was followed for the comparison between the simulation software implemented in hardware compared to its general-purpose processor counterpart. This section focused on comparing the performance of both simulation platforms.

### 4.6. **Data analysis (objective 3 & 4)**

The data was interpreted with a statistical analysis comparing the simulation output, first between the benchmark software and the simulation program with a general-purpose processor, and secondly between the simulation program implemented in a general-purpose processor and the same program implemented in an FPGA. Data analysis was performed with Python’s Pandas module to extract descriptive statistics and error value comparisons, and Matplotlib to generate voltage curve graphs for visual comparison between the programs’ results.

### 4.7. **Report of results (objective 3 & 4)**

Conclusions were produced and reported in two scientific articles and the present master’s thesis.

# Synthesis.

## 1. **Summary of the thesis**

The main objective of the present research was to develop a mathematical model for microgrid simulation and implement it in an application-specific processor to measure the performance gains obtained with this approach against the performance of a general use processor. Given that this project was a first approach the topic of microgrid simulation, its scope was limited to stand-alone microgrids with photovoltaic generators as the main source of energy.

A mathematical model was presented and implemented in the simulation program. An Intel Core i7-8750H was utilized as the general-purpose processor. The model was programmed in Python 3.6 language with the use of C optimized libraries: Numpy for arithmetic operations and Pandas for data handling tasks. Differential equations were solved with Euler's explicit integration method that was selected for its simplicity which could be leveraged during the hardware implementation stage of the project. The validation of the simulation program was done through the comparison of results against PowerSim which was selected as a benchmark simulation software. A specific microgrid architecture was selected and simulated with open-loop control, for which the approximate absolute error was 1% for the steady state regime.

The same simulation program was then ported to C and implemented in a Xilinx Zedboard FPGA development board. High-Level Synthesis (HLS) was used to convert the program to its hardware counterpart. The previous microgrid architecture was programmed in the board for comparison against the general-purpose CPU. The performance gains were measured based on execution time. For the selected architecture, and with varying number of elements in the microgrid, the FPGA based program was 14.2 to 87.1 times faster. Real-time simulation was also achieved with up to 2.7 times greater magnitude of signal oscillations during the system transient.

## 2. **Articles**

The thesis was developed from a compendium of two articles, as indicated below:

1. Araya, M. and Meza, C. (2022). Wide-range time-domain simulation environment for stand-alone microgrids. In Escamilla-Ambrosio, P.J., Hernández-Callejo, L., Nesmachnow, S., Moreno, P. & Rossit, D. (Eds.), *Proceedings of the IV Ibero-American Conference on Smart Cities*. CITIES., Cancún, México, November 29 – December 1, 2021, 856–870.
2. Araya-Carrillo, M., Meza, C., Salazar-Garcia, C., & Chacón-Rodríguez, A. (2022). A stand-alone photovoltaic microgrid simulation environment using FPGA hardware acceleration. 13th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2022), 2022, 386–394. doi:10.1049/icp.2023.0024

Each one of the articles has a contribution to the specific objectives of the thesis, allowing this, to reach the general objective of the research. The contribution of each one is indicated below.

Article 1 presents a microgrid description model. Subsequently a simulation program is developed with said model and validated by comparing against a benchmark simulation software. The article covers the **objectives 1, 2 and 3** of this thesis.

Article 2 covers **objective 4** of this thesis. The simulation program developed in the first article was implemented in a hardware with an FPGA board and validated. Furthermore, a performance comparison between the hardware accelerated and general-purpose processor version is done to measure the performance benefits of the hardware implementation.

## Article 1.

# Wide-range time-domain simulation environment for stand-alone microgrids

Journal: Proceedings of the IV Ibero-American Conference on Smart Cities

Authors: Araya, M. and Meza, C.

Year: 2022

ISBN: 978-607-99960-0-0

# Wide-range time-domain simulation environment for stand-alone microgrids

Mario Araya-Carrillo[1[0000−0001−9626−186x]] and
Carlos Meza[1,2[0000−0002−7374−505x]]

[1] Costa Rica Institute of Technology,
30101 Cartago, Costa Rica

[2] Anhalt University of Applied Sciences
06366 Köthen, Germany

**Abstract.** Microgrids represent a growing paradigm shift from centralized energy generation to a distributed model. However, given the relative novelty compared to traditional grids, there are still many unknown factors regarding the optimal design of this kind of system. In this paper, a mathematical model to describe microgrids is presented and a stand-alone microgrid simulation program for a wide range of time domains was proposed as a tool to facilitate the design process and study of these microgrids. The program was validated employing simulation of a stand-alone microgrid with photovoltaic generation and electrochemical energy storage. The program limitations are discussed, and further improvements are proposed to increase execution time performance.

**Keywords:** microgrid · photovoltaics · stand-alone

## 1 Introduction

Microgrids are systems that have attracted interest in recent years due to their potential advantages compared to centralized power generation, distribution, and consumption systems. Among the benefits of microgrids are their flexibility to operate as direct current (DC), alternating current (AC), or hybrid systems (AC and DC), thus facilitating the coupling of all types of energy sources with emphasis on clean and renewable sources [14], [19]. They also present lower energy losses during transmission since they follow an *in situ* generation model or are located at short distances from loads and can reduce the number of conversion stages along distribution lines [7], [10].

Microgrids can operate connected to the main power grid or in isolation. The latter is a viable option to achieve the electrification of rural areas due to their capacity for self-sufficiency. Microgrids represent an alternative in those places where the power grid does not meet the appropriate levels of reliability, stability, and quality of energy required [15]. Given these characteristics, isolated microgrids are an area of study that could have a high impact at a global level on the quality of life of people in regions far from metropolitan centers.

Due to the hegemony of centralized AC systems, more research is still required in the area of isolated microgrids, especially those focused on DC, regarding their modeling and standardization [16, 17]. Research topics of interest in microgrids are what is the ideal microgrid architecture for any specific application, what are the optimal voltage levels for power distribution buses, what type of control strategy (centralized or distributed, short or long term prediction) facilitates a more efficient use and higher availability of energy as assessed in [3, 13], among others.

A cost-effective alternative to analyze and optimize microgrid design and operation is through numerical simulations of microgrid models. Research works such as [9], [8] and, [2] use simulations to optimize the design process of stand-alone microgrids. Those works provide important insight into the operation of microgrids over their steady-state operation. Nevertheless, they do not provide information about the transient response of the different elements connected to the microgrid. Information about transients is useful because there might be overshoots that may damage equipment or affect the operation of the whole system.

Given these factors, the design of a simulation program for electrical microgrids, in a wide range of time domains, is presented to evaluate and eventually optimize different architectures, loading conditions, energy balance, and control strategies. The novelty of the proposed work is that the obtained code for the microgrid mathematical model captures the system's behavior in a wide range of time domains. In other words, short-time transient events, as well as longer dynamic evolution are visible. In this way, it is possible to optimize the microgrid design and operation process before implementing it. As a first approach, the program is bounded within the context of isolated DC microgrids. A generalized mathematical model for microgrids is proposed and a validation process of the program is performed employing a comparison with a prevalidated circuit simulation software. The aim is to create a tool to speed up the microgrid simulation, analysis, and design process. Therefore, the execution time of the program was defined as a variable of interest, and the limitations of the program, as well as the possible solutions to improve its performance such as the hardware acceleration proposed in [5] are discussed. In this regard, the results obtained in this paper will serve to develop hardware-specific systems (digital twins) that allow to significantly reduce the simulation time of microgrids.

## 2 Microgrid mathematical model

This paper deals with the development of a wide-range time-domain simulation environment for a microgrid with a bus topology. This network configuration is the most widely used because it allows simple integration of the different components that make up a microgrid, e.g., power generators, storage, load. The elements of the microgrid are interconnected with each other through buses that can be direct current or alternating current as shown in the Figure 1, where, for simplicity, the power processing units, which are located between the sources

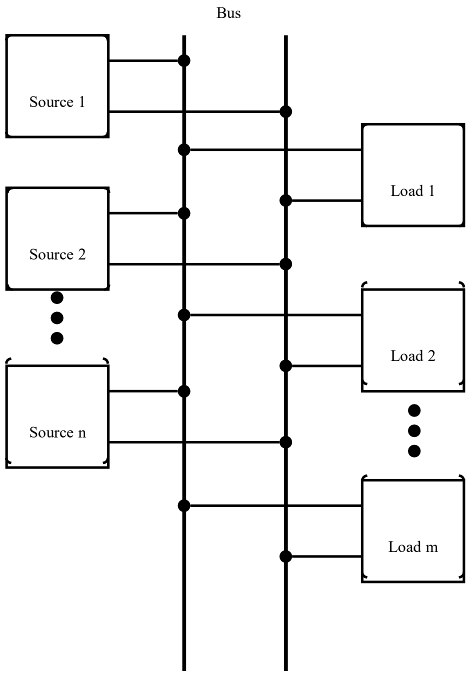

**Fig. 1.** Microgrid architecture

and the bus and between the loads and the bus, are omitted. It is also possibleto have configurations of more than one bus such as the one shown in Figure 2.

From a mathematical modeling point of view, the incorporation of the buses can be represented by the following set of equations

$$v_{b_q} = v_{1_{bq}} = v_{2_{bq}} = \dots \quad (1)$$

$$\sum i_{j_{bq}} = 0 \quad (2)$$

Where $v_{b_q}$ represents the voltage on the $q$-th bus and $v_{1_{bj}}$, $v_{1_{bj}}$, . . ., are the voltages of the devices connected to that bus. Likewise, $i_{j_{bq}}$ represents the current contributed by the $j$-th device connected to the $q$-th bus.

In addition to the buses, the model considers the following microgrid components:

- power generators,
- energy storage systems (e.g. electrochemical batteries),
- power converters,
- electrical loads,
- control subsystems.

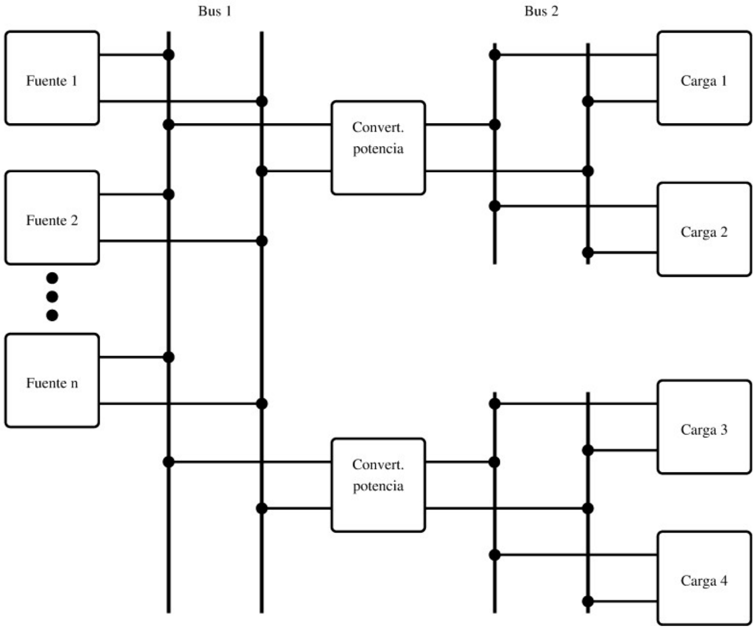

**Fig. 2.** Multi-bus microgrid architecture

The modeling was carried out by means of algebraic differential equations described as follows

$$\frac{dv_i}{dt} = f_i(v_1, v_2, \ldots, v_n, i_1, i_2, \ldots, i_m, u_k, \rho_i) \quad (3)$$

$$\frac{di_j}{dt} = f_j(v_1, v_2, \ldots, v_n, i_1, i_2, \ldots, i_m, u_k, \rho_j) \quad (4)$$

Where $v_i$and $i_j$ represent the system voltages and currents associated with those elements that can store energy, i.e., capacitors and inductors, respectively. Thus, the open-loop microgrid can be represented with $n+m$ first-order differential equations where $n$ is the number of capacitive elements and $m$ is the number of inductive elements. The rates of change of these voltages and currents are determined by a set of functions, $f_i(\cdot)$ and $f_j(\cdot)$, which are related to the network topology and the characteristics of the sources and loads connected to the network. These functions depend on the electrical variables of the system and on a set of parameters $\rho_i = \{\rho_{i_1}, \rho_{i_2}, \cdots\}$ and $\rho_j = \{\rho_{j_1}, \rho_{j_2}, \cdots\}$ specific to the elements that make up the microgrid. It is important to clarify that these functions are discontinuous and for systems using photovoltaic energy they are also nonlinear. For example, the function that determines the current of a photovoltaic generator is shown in the equation 5, where $\mathrm{v}_{\mathrm{pv}}$ is the voltage at the terminals of the generator, and $\Lambda$, $\Psi$ and $\alpha$ are parameters.

$$V_{pv} = \Lambda - \Psi \mathrm{e}^{\alpha \mathrm{v}_{\mathrm{pv}}} \quad (5)$$

The subsystems that make up the microgrid must be managed and controlled in such a way as to ensure energy balance and correct operation. Therefore, in addition to (3) and (4) there is also a set of equations that describe the dynamic performance of the control subsystems, i.e.

$$u_k = g_k\left(\frac{d^{n_u}u_k}{dt}, \frac{d^{n_u-1}u_k}{dt}, \dots, \frac{du_k}{dt}, \frac{d^{n_v}v_i}{dt}, \frac{d^{n_v-1}v_i}{dt}, \dots, \frac{dv_i}{dt}, v_i, \frac{d^{n_j}i_j}{dt}, \frac{d^{n_i-1}i_j}{dt}, \dots, \frac{di_j}{dt}, i_j, \rho_k\right) \tag{6}$$

Where $\mathrm{u_k}$ represents the $\mathrm{k}$-th control signal for one of the elements of the microgrid that is generated by the control subsystem $\mathrm{k}$ and where $\rho_k\{\rho_{k_1}, \rho_{k_2}, \dots\}$ are its parameters. Note that, unlike the expressions in $(3)$ and (4), the function $g_k(\cdot)$ can depend on the derivatives of the input voltages, currents, and signals, which means that the expression (6) can be described as a system of $n_u$ first-order differential equations.

The sources, loads and elements interconnecting the buses are managed by a power processing unit which, given their high efficiency, consist of power MOSFET circuits that are switched in such a way that the desired conversion process is achieved. The control signals $u_k$ control these power units and represent duty cycles that are converted to discontinuous signals by means of a pulse width modulator. The discontinuous signals, which have a switching frequency of between 100~kHz up to 1~MHz, are responsible for turning the transistors on and off. This activation process generates a ripple in the currents and voltages of the system that can provide useful information for its management, as indicated in (Meza, 2013). If these equations are solved in real-time, then it is possible to visualize and analyze ripple signals at all points of the microgrid.

Nevertheless, the development of digital models of the electrical microgrids described by means of the equations (1), (2), (3), (4) and (6) presents important challenges, which are detailed below:

- The mathematical model of the microgrid can be described by a set of differential equations of order $n + m + n_u$, where even for a system with few generators and loads one can have a high order system. For example, a microgrid with two power sources and two independent loads can be described, in its simplest configuration, by a 6th order differential equation.
- The emulation time of the digital microgrid will depend on the characteristics of the connected generators and loads. For example, for a microgrid based on photovoltaic generators with resistive loads, the emulation time should be at least 10 minutes, in order to see all the transient processes of the load balance. It is also desirable to perform a 24-to-48-hour analysis to evaluate the load balance on predetermined days.
- Microgrids based on photovoltaic generators can be described by means of a set of first order difference equations and a set of algebraic equations. The number of differential equations depends on the number of elements, i.e., generators and

- loads, present in the grid. The numerical solution of this set of differential equations must focus on the parallelization of the processes.
- The storage and display of storage system data can represent a bottleneck for the emulation system for the following reasons:
  - There is a difference of several orders of magnitude between the period of the signals that could be displayed and the emulation time. For example, the ripple of the system currents and voltages is between 10 microseconds and 0.01 microseconds, while the emulation time could be from 10 minutes to 48 hours.
  - There are a large number of possible measurement points in the microgrid. For example, each generator and power processing unit set has at least 2 voltage and 2 current values that could be useful to visualize. The same amount can be considered for the loads.

## 3 Simulation approach

As stated previously, we are looking for a microgrid simulation program that can resolve the values of the system's electrical variables over a wide range oftime domains; from the small microseconds and milliseconds scale to capture the ripple in currents and voltages and system transients, and on the other hand, alarger scale at the minutes, hours and days level to be able to observe characteristics of the system energy balance, the state of charge of storage elements and the response to different input vectors of the photovoltaic generation system and loads, i.e., to solve equations (1), (2), (3), (4) and (6).

The microgrid architecture proposed in [11] was taken as a reference as a basis for evaluating the performance of the simulation program as it was developedfor scalable stand-alone microgrids for rural electrification with the use of photovoltaic generators. The architecture, shown in Figure 3, consists of *n* number of PV arrays as a power source, with a respective boost converter each. All boost converters feed a single high voltage distribution bus $V_H$ with voltages between 360 V and 400 V. Connected to this bus are a *m* number of load branches eachfeeding a buck converter, called distribution nodes, with an intermediate voltage $V_M$ between 45 V and 50 V, and these, in turn, feed a *l* number of buck converters, where each converter is a power management unit (PMU) connected to the load and storage system in the form of 12 V electrochemical batteries througha low voltage bus $V_L$.

Analyzing the most simple case of the chosen architecture, i.e., a single power source and a single loading branch as shown in Figure 4, it is relatively trivialto obtain the dynamic equations that describe the system. For this case, we obtain a system with the six first-order differential equations (7) to (13); without accounting for the equations of the switching control algorithms which may beof higher order. However, this is the smallest possible case, in reality, one can have multiple load branches and each of these can in turn have several PMUs with their respective loads. Thus, the number of elements that make up the

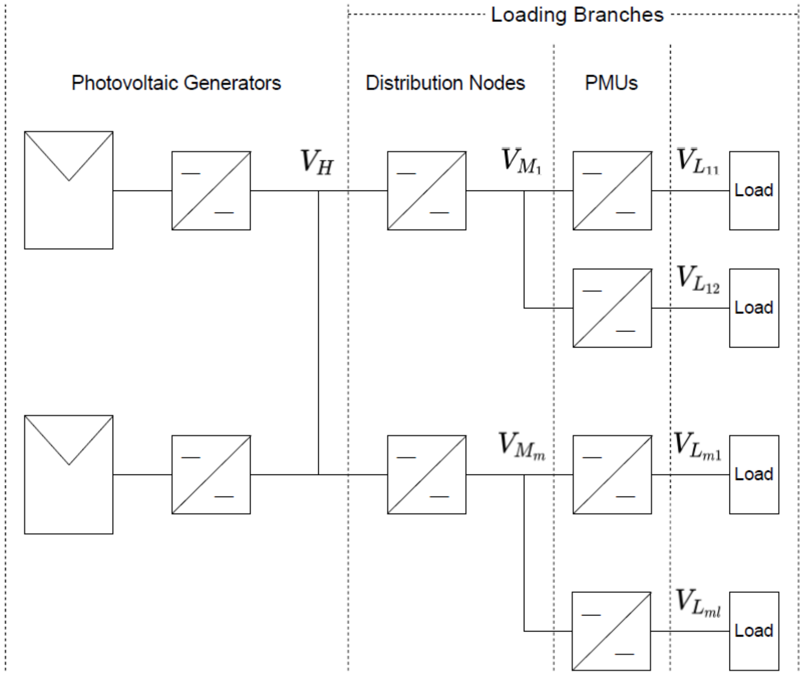

**Fig. 3.** Multi-bus microgrid architecture based on Madduri et al [11].

microgrid can grow by one or more orders of magnitude, along with the number of equations of the system. The task of obtaining all the dynamic equations could quickly become impractical.

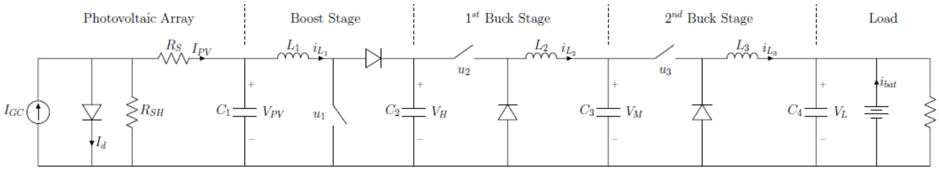

**Fig. 4.** Microgrid equivalent circuit with one photovoltaic source and one loading branch. Based on Madduri et al [11].

$$\frac{dV_{PV}}{dt} = \frac{i_{PV} - i_{L_1}}{C_1} \tag{7}$$

$$\frac{di_{L_1}}{dt} = \frac{V_{PV} - V_H(1 - u_1)}{L_1} \tag{8}$$

$$\frac{dV_H}{dt} = \frac{(1 - u_1)i_{L_1} - u_2 i_{L_2}}{C_2} \tag{9}$$

$$\frac{di_{L_2}}{dt}=\frac{u_2V_H+V_M}{L_2} \tag{10}$$

$$\frac{dV_M}{dt}=\frac{i_{L_2}-u_3i_{L_3}}{C_3} \tag{11}$$

$$\frac{di_{L_3}}{dt}=\frac{u_3V_M-V_L}{L_3} \tag{12}$$

In order to reduce the complexity of larger systems, it was deemed convenient to decompose and describe the microgrid through a series of blocks that represent the different elements that compose it (energy sources, power converters, loads,and energy storage). Code-wise, each block is represented as an object or a class composed of its electrical parameters and the dynamic equations that describe its behavior. In such a way that each block has a single dynamic equation and is interconnected with adjacent blocks according to the microgrid architecture. For each calculation iteration, the equations of all the blocks are solved separately and then the resulting currents and voltages are propagated to the neighboring blocks.

Therefore, the dynamic equations of the boost and buck power converters can be modeled with equations (14) and (15) respectively, and the voltage buses with equation (16) in accordance with the voltage equation of a capacitor. For the photovoltaic generators, the equivalent circuit equations of the 1D2R model [4] were used.

$$\frac{di_L}{dt}=\frac{V_{in}-(1-u)V_{out}}{L} \tag{13}$$

$$\frac{di_L}{dt}=\frac{uV_{in}-V_{out}}{L} \tag{14}$$

$$\frac{dV_C}{dt}=\frac{i_{in}-i_{out}}{C} \tag{15}$$

### 3.1 Program sequence

The simulation program developed was implemented using Python language and structured in a linear way as shown in Figure 5. So that first the architecture of the microgrid to be simulated is defined, where all the objects/classes representing the different elements of generation, power converters, storage, and loads arecreated. Next, the simulation parameters corresponding to time variables and a constant integration step are defined. Then the initial state of the microgrid is established, in other words, the voltages and currents in the system, as well as the state of charge of the batteries.

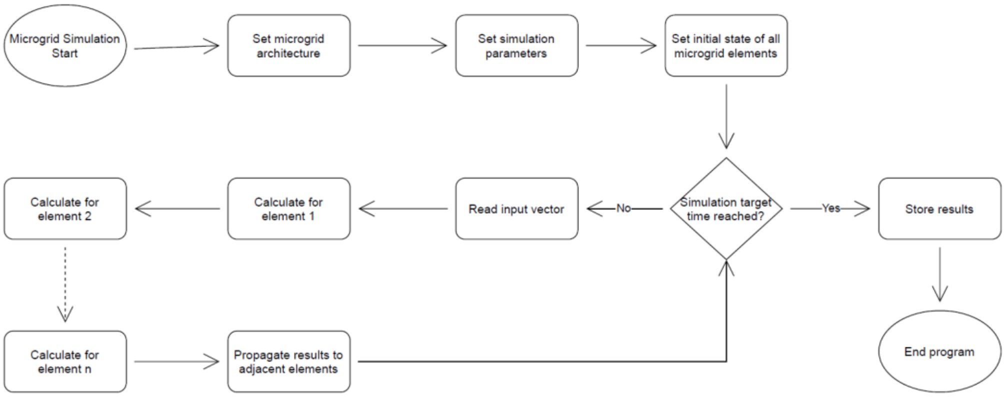
Microgrid Simulation Start
Set microgrid architecture
Set simulation parameters
Set initial state of all microgrid elements
Simulation target time reached?
No
Yes
Read input vector
Store results
Calculate for element 1
Calculate for element 2
Calculate for element n
Propagate results to adjacent elements
End program

**Fig. 5.** Stand-alone DC microgrid simulation program flowchart.

The next stage starts the simulation process in a loop that stops when the final simulation time is reached. During the simulation, the program reads an input vector that includes the solar irradiance intensity and magnitude of the loads. It then proceeds to sequentially solve the dynamic equations of all the elements of the microgrid. For the solution of the equations, the explicit Euler integration method is used [6], which was chosen for its ease of calculation and also because it allows solving for each element of the microgrid individually for each iteration. Once all the elements of the microgrid have been solved for the current iteration, the results of each element are propagated throughout the microgrid and the next iteration is initiated.

Lastly, when the final simulation time is reached, the results are stored for further analysis.

## 4 Simulation results and discussion

In order to assess the program’s capability, three different tests were conducted. First, a validation test to confirm that the program returns correct results. Thena demonstration of the ability to simulate signals in a wide range of time domainsand finally a test to measure the time it takes for the program to simulate one second for microgrid configurations with differing number of elements.

### 4.1 Program validation

To validate the performance of the program, a comparison of the results obtained against the software PowerSim (PSIM) was carried out; this software was taken as a reference since it is designed for the simulation of power circuits and it is a prevalidated software for commercial use.

A simulation of the circuit previously shown in Figure 4 was performed for a time of 1 s and a with a step of 0.1 μs to capture the transient period of microgrid start-up and subsequent stabilization.

For the microgrid voltages, a maximum absolute error of 0.98 % was obtained for PSIM and 1.08 % for the Python-based program as shown in Table 1.

**Table 1.** Steady state voltage comparison for PSIM and Python simulations.

| *Parameter* | Theoretical Steady State [V] | PSIM Result [V] | PSIM Error [%] | Python Result [V] | Python Error [%] |
|---|---|---|---|---|---|
| $V_{PV}$ | 179.71 | 181.48 | -0.98 | 181.33 | -0.91 |
| $V_H$ | 359.42 | 362.94 | -0.98 | 362.67 | -0.90 |
| $V_M$ | 44.93 | 45.36 | -0.96 | 45.10 | -0.38 |
| $V_L$ | 11.23 | 11.29 | -0.52 | 11.35 | -1.08 |

In the case of the currents, a maximum absolute error of 0.78 % was obtained for PSIM and 1.12 % for the Python program as can be seen Table 2.

**Table 2.** Steady state current comparison for PSIM and Python simulations.

| *Parameter* | Theoretical Steady State [V] | PSIM Result [V] | PSIM Error [%] | Python Result [V] | Python Error [%] |
|---|---|---|---|---|---|
| $i_{L1}$ | 1.404 | 1.394 | 0.74 | 1.394 | 0.68 |
| $i_{L2}$ | 5.616 | 5.574 | -0.51 | 5.678 | -1.12 |
| $i_{L3}$ | 22.463 | 22.297 | 0.78 | 22.706 | -1.08 |

For both cases, it is considered that the error of approximately 1 % achieved is acceptable since the solution method selected for its simplicity was not affected by the discontinuous equations caused by the high-frequency switching of the MOSFETs. Also, the error can be reduced by using a smaller integration step, although this would have an impact on the simulation's execution time.

Furthermore, by means of the voltage curves shown in Figure 6 it can be seen that the resulting behavior is similar for both programs. The oscillations during the transient in the curve generated by Python are larger in magnitude than those of PSIM. This is because the models simulated in Python are idealand do not consider the internal resistance of the capacitors and inductors ofthe system. PSIM on the other hand, requires the configuration of minimum internal resistance, in the range of the milliohms to facilitate the convergence of the solver.

### 4.2 Time domains

To demonstrate the capability of the simulation program in different time domains, the aim was to study the behavior of the system on a small time scale, given by the switching frequency of the MOSFETs, and one of several orders of magnitude larger given by the state of charge of the battery in the circuit.

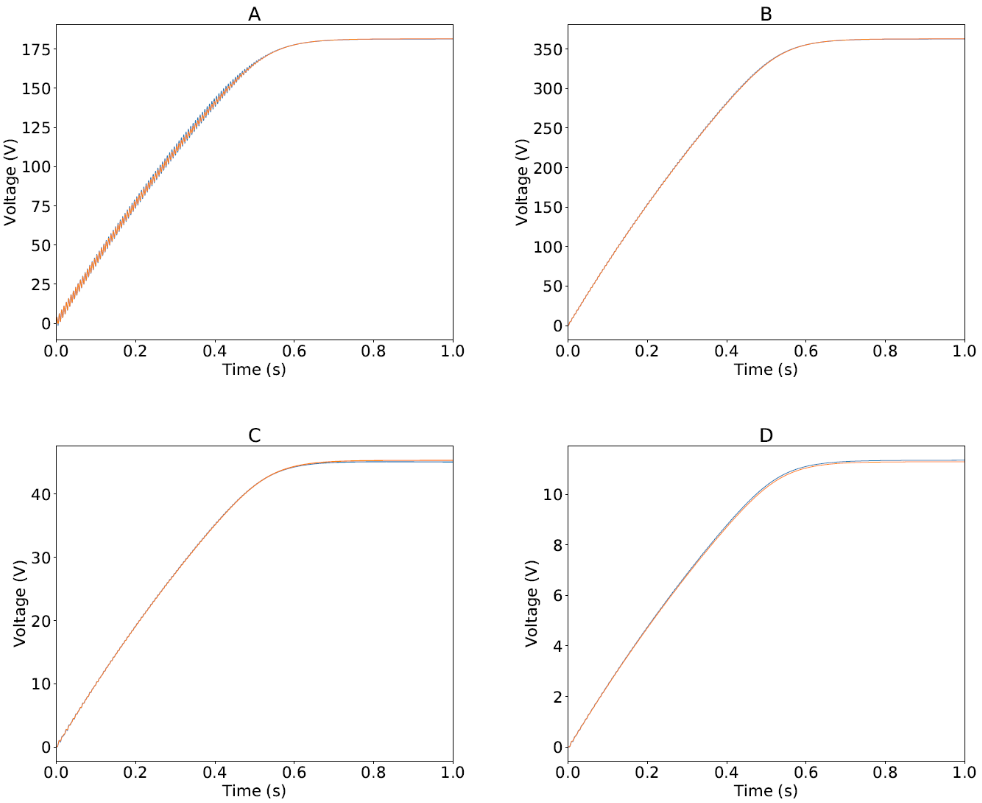

**Fig. 6.** Voltage curve comparison between Python (blue) and PSIM (orange) simulations. A) $V_{PV}$ B) $V_H$ C) $V_M$ D) $V_L$

.

The $V_M$ bus ripple obtained was a sawtooth signal with an oscillation range of 0.04216 V in Python and 0.04208 V in PSIM, which corresponds to a differenceof less than 0.2 %. In Figure 7 the results of both programs were normalized. It can be seen how both signals overlap in both magnitude and phase.

In order to demonstrate the change in the state of charge of the battery, the simulation time was extended to 10 minutes since a period of 1 s producesa negligible variation. The power consumed by the load was varied to see the effect on the charge/discharge rate of the battery. The simulation started witha load of 160 W, then increased to 630 W at $t$ = 200 s, and finally reduced to80 W at $t$ = 400 s.

Figure 8 shows how the state of charge initially increases, indicating that the battery is charging. Then when the load increases at $t$ = 200 s, the PV generation is not sufficient to supply the required power, so the battery discharges to compensate for the missing power. Finally, the battery begins charging again at$t$ = 400 s.

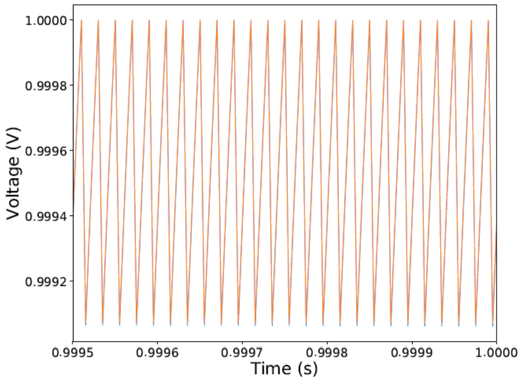

**Fig. 7.** Normalized voltage ripple of $V_M$ bus at 0.0995 s <= $t$ <= 1 s. Python (orange) y PSIM (blue).

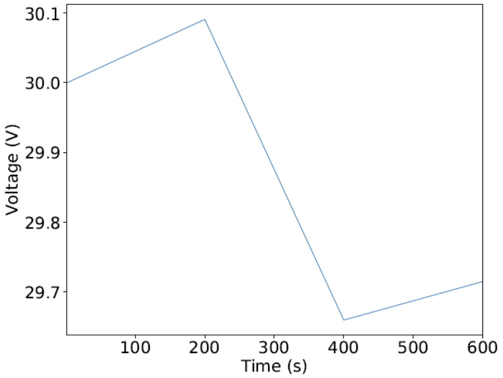

**Fig. 8.** State-of-charge (SOC) variation of the energy storage component. System load defined as 160 W at $t$ = 0 s, 630 W at $t$ = 200 s, and 80 W at $t$ = 400 s.

### 4.3 Simulation time

The performance of the developed program with respect to the execution time of the simulations was measured for three simulations with 1, 5, and 10 loading branches. In all three cases, a single PV array was set as the generator anda simulation time of 1 s was used as a benchmark. It was determined that for the selected microgrid architecture, simulating one second requires executing theprogram for a duration of 516 s for 1 loading branch. This time increases linearly with the number of elements in the microgrid as shown in Table 3, reaching upto 3788 s with 10 loading branches.

The execution time of the program, which is considered a vital factor presents a bottleneck, which is caused by three factors. Firstly, the program was writtenin Python, which is a high-level interpreter that is not optimized for applications that require high performance in a short time. To overcome this limitation, the

**Table 3.** Execution time of microgrid simulations with varying number of elements.

| Number of Load Branches | Number of Elements | Execution Time [s] |
|---|---|---|
| 1 | 7 | 516 |
| 5 | 22 | 2172 |
| 10 | 42 | 3788 |

program can be translated into a lower abstraction level language, such as C, which allows better use of computational resources.

The second factor that contributes to the long execution time of the programis its linearity. The equations of the system are solved one after the other, so a reduction in execution time can be achieved by parallelizing the calculation process. This can be done by adapting the program to make use of multiple processor cores/threads. Hardware acceleration can also be used through the implementation of the program on a stand-alone FPGA board or in combination with specialized emulation equipment as used in [1, 18].

The third factor is a small integration step that forces the program to per- form millions of iterations for each simulated second. The step can be increased, however, its maximum value depends on the switching frequency of the MOSFETs and the transients of the system, so it is not considered an ideal solution.In addition, a larger step size may result in convergence issues in the solution of the dynamic equations of the microgrid due to stiffness since these are discontinuous. For these reasons, it is considered appropriate to address the first two factors.

## 5 Conclusions

A model was proposed to describe different microgrid architectures employing aseries of algebraic differential equations dependent on the voltages and currentsof the microgrid elements, a set of parameters specific to each element, and a setof equations to describe the dynamic behavior of the control subsystems.

A simulation program for isolated microgrids in a wide range of time do- mains was developed. The results of the program were validated by comparison with the results of the PSIM software. An absolute error of approximately 1 % was obtained and the ability to simulate the behavior of the selected microgrid architecture in both a domain in the range of microseconds as the ripple in the voltage curves, as well as in a domain in the range of several minutes in the case of the state of charge of the energy storage system was showcased.

The performance of the program was evaluated with respect to its execution time in simulations of microgrids with different number of elements. Timewas found to increase linearly with the number of elements that make up themicrogrid. Optimization measures proposed for the program included the modification of the sequential simulation of the microgrid elements in favor of a parallel simulation strategy or the acceleration of the computation using a hard-ware implementation.

# References

1. Abrishambaf, O., Faria, P., Gomes, L., Spínola, J., Vale, Z., Corchado Rodríguez, J.: Implementation of a real-time microgrid simulation platform based on centralized and distributed management. Energies 10, 806 (06 2017)
2. Alam, M., Kumar, K., Srivastava, J., Dutta, V.: A study on dc microgrids voltages based on photovoltaic and fuel cell power generators. In: 2018 7th International Conference on Renewable Energy Research and Applications (ICRERA). pp. 643– 648. IEEE (2018)
3. Alzahrani, A., Ferdowsi, M., Shamsi, P., Dagli, C.H.: Modeling and simulation of microgrid. Procedia Computer Science 114, 392 – 400 (2017), complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA
4. Araújo, N., Sousa, F., Costa, F.: Equivalent models for photovoltaic cell–a review. Revista de Engenharia Térmica 19(2), 77–98 (2020)
5. Brenes, E.M., Meza, C.: An application-specific instruction set processor for microgrid simulation. In: 2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX). pp. 1–6 (2019)
6. Chakraverty, S., Mahato, N., Karunakar, P., Rao, T.: Advanced Numerical and Semi-Analytical Methods for Differential Equations. Wiley (2019)
7. Elsayed, A.T., Mohamed, A.A., Mohammed, O.A.: Dc microgrids and distribution systems: An overview. Electric Power Systems Research 119, 407–417 (2 2015)
8. Hu, G., Li, S., Cai, C., Wu, Z., Li, L.: Study on modeling and simulation of photo- voltaic energy storage microgrid. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE). pp. 692–695. IEEE (2017)
9. Khadepaun, N.C., Shah, N., et al.: Operation of solar pv with pem fuel cell for remote hybrid microgrid. In: 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). pp. 1–6. IEEE (2020)
10. Lasseter, R.H.: Microgrids. In: 2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings. vol. 1, pp. 305–308. IEEE (2002)
11. Madduri, P.A., Poon, J., Rosa, J., Podolsky, M., Brewer, E., Sanders, S.: A scalable dc microgrid architecture for rural electrification in emerging regions. In: 2015 IEEE Applied Power Electronics Conference and Exposition (APEC). pp. 703–708 (2015)
12. Meza, C., Ortega, R.: On-line estimation of the temperature dependent parameters of photovoltaic generators. IFAC Proceedings Volumes 46(11), 653–658 (2013)
13. Olivares, D.E., Cañizares, C.A., Kazerani, M.: A centralized energy management system for isolated microgrids. IEEE Transactions on Smart Grid 5(4), 1864–1875 (2014)
14. Planas, E., Andreu, J., Gárate, J.I., de Alegría, I.M., Ibarra, E.: Ac and dc technology in microgrids: A review. Renewable and Sustainable Energy Reviews 43, 726–749 (3 2015)
15. Tank, I., Mali, S.: Renewable based dc microgrid with energy management system. In: 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). pp. 1–5 (2015)
16. Unamuno, E., Barrena, J.A.: Hybrid ac/dc microgrids—part i: Review and classification of topologies. Renewable and Sustainable Energy Reviews 52, 1251–1259 (12 2015)
17. Unamuno, E., Barrena, J.A.: Hybrid ac/dc microgrids—part ii: Review and classification of control strategies. Renewable and Sustainable Energy Reviews 52, 1123–1134 (12 2015)
18. Zhang, B., Fu, S., Jin, Z., Hu, R.: A novel fpga-based real-time simulator for micro- grids. Energies 10(8) (2017)
19. Zhang, Y., Gatsis, N., Giannakis, G.B.: Robust energy management for microgrids with high-penetration renewables. IEEE transactions on sustainable energy 4(4), 944–953 (2013)

## Article 2.

# A stand-alone photovoltaic microgrid simulation environment using FPGA hardware acceleration

State: To be published

Authors: Araya, M., Meza, C., Salazar-García, C., Chacón-Rodríguez, A.

Year: 2022

# A stand-alone photovoltaic microgrid simulation environment using FPGA hardware acceleration

*Mario Araya-Carrillo*[1]*, Carlos Meza*[2]** Carlos Salazar-García*[1]*, Alfonso Chacón-Rodríguez*[1]

[1]*Costa Rica Institute of Technology, 30101 Cartago, Costa Rica*
[2]*Anhalt University of Applied Sciences, 06366 Köthen, Germany*
**E-mail: carlos.meza@hs-anhalt.de*

**Keywords:** photovoltaic energy, microgrid, FPGA

## Abstract

Microgrids represent a growing paradigm shift from centralised energy generation to a distributed model. However, given the relative novelty compared to traditional grids, there are still many unknown factors regarding the optimal design of this kind of system. Given this, the study of microgrids can benefit from developing tools designed specifically for such purposes. This paper presents the results of the hardware acceleration of a previously developed stand-alone microgrid simulation environment through an FPGA. The accuracy of the system was validated through a comparison with benchmark software. Execution time performance is measured against the non-hardware accelerated simulation program for microgrid configurations with varying sizes. Real-time simulations are achieved with a reduction of accuracy during transients. Finally, hardware resource usage and further improvements are discussed.

## 1. Introduction

A microgrid is a set of electrically interconnected power sources and storage units coordinated to supply power to a group of loads. A microgrid can be isolated or connected to other microgrids or the primary electricity grid. The elements interconnected to the microgrid usually operate in a plug- and-play manner, i.e. new elements can be integrated into the microgrid without significantly modifying its functionality.

Microgrids are one of the fundamental systems in the distributed generation paradigm. In this paradigm, the aim is to decentralize the generation and management of energy systems. In this way, the objective is to achieve a more robust system with lower transmission losses, [1], [2]. Stand-alone microgrids have the advantage that they can offer more flexibility and robustness to loads, [3].

More research is still required in the area of stand-alone microgrids, especially those focused on DC, regarding their modelling and standardization [4, 5]. More specifically, according to [6, 7] the following research questions remain without a clear answer:

- What is the ideal microgrid architecture for any specific application?
- What are the optimal voltage levels for power distribution buses?
- What control strategy (centralized or distributed, short or long-term prediction) facilitates a more efficient use and higher energy availability?

A cost-effective alternative to analyze and optimize microgrid design and operation is through numerical simulations of microgrid models. Research works such as [8], [9] and, [10] use simulations to optimize the design process of stand-alone microgrids. Those works provide important insight into the operation of microgrids over their steady-state operation. Nevertheless, they do not provide information about the transient response of the different elements connected to the microgrid. Information about transients is useful because there might be overshoots that may damage equipment or affect the operation of the whole system.

The present research was proposed as a solution to the large execution time of the program, which was determined as its biggest limitation. Therefore, a hardware acceleration approach was selected to significantly reduce the execution time and evaluate the possibility of running real-time simulations with the same algorithms. The previously developed program was implemented in a field-programmable gate array (FPGA). This approach has already been utilised for grid-tied, isolated, and hybrid microgrid simulations in [11–13], where real-time simulation has been achieved for different microgrid architectures and sizes. A similar approach for utilizing FPGAs to emulate grid systems can be found in [14]. Also, in [15], FPGA was used to estimate the parameters of PV modules. In all the cases mentioned above, the non-linear equation of a PV module was emulated and run in real-time, as is done in the present work.

The rest of the paper is structured as follows: Section 2 presents the mathematical model of the microgrid analyzed in this paper. Section 3 describes the simulation approach followed, including the hardware emulation. Section 4 presents and discusses the simulation results. Finally, the conclusions are presented in Section 5.

## 2. Microgrid mathematical model

This paper deals with developing a wide-range time-domain simulation environment for a microgrid with a bus topology.

This network configuration is the most widely used because it allows simple integration of the different components that make up a microgrid, e.g., power generators, storage, and load. The elements of the microgrid are interconnected with each other through buses that can be direct current or alternating current. It is also possible to have configurations of more than one bus.

From a mathematical modelling point of view, the incorporation of the buses can be represented by the following set of equations.

$$v_{b_q} = v_{1_{bq}} = v_{2_{bq}} = \ \dots \ = v_{k_{bq}} \tag{1}$$

$$\sum i_{k_{bq}} = 0 \tag{2}$$

Where $v_{b_q}$ represents the voltage on the q-th bus and $v_{1_{bq}}$, $v_{2_{bq}}$, ..., $v_{k_{bq}}$are the voltages of the devices connected to that bus. Likewise, ikbq represents the current contributed by the k-th device connected to the q-th bus.

In addition to the buses, the model considers the following microgrid components:

- power generators,
- energy storage systems (e.g. electrochemical batteries),
- power converters,
- electrical loads,
- control subsystems.

The modelling is carried out employing algebraic differentia equations described as follows.

$$\frac{dv_i}{dt} = f_i(v_1, v_2, \dots, v_n, i_1, i_2, \dots, i_m, u_k, \rho_i) \tag{3}$$

$$\frac{di_j}{dt} = f_j(v_1, v_2, \dots, v_n, i_1, i_2, \dots, i_m, u_k, \rho_j) \tag{4}$$

where $v_i$ and $i_j$ represent the system voltages and currents associated with those elements that can store energy, i.e., capacitors and inductors, respectively. Thus, the open-loop microgrid can be represented with n + m first-order differential equations where n is the number of capacitive elements, and m is the number of inductive elements. The rates of change of these voltages and currents are determined by a set of functions, $f_i$ (·) and $f_j$ (·), which are related to the network topology and the characteristics of the sources and loads connected to the network. These functions depend on the electrical variables of the system and on a set of parameters $\rho_i$= {$\rho_{i1}$, $\rho_{i2}$,...} and $\rho_j$= {$\rho_{j1}$, $\rho_{j2}$,...} specific to the elements that make up the microgrid. It is important to clarify that these functions are discontinuous, and for systems using photovoltaic energy, they are also nonlinear. For example, the function that determines the current of a photovoltaic generator is shown in the equation (5), where $V_{pv}$ is the voltage at the terminals of the generator, and Λ, Ψ and α are parameters.

$$V_{pv} = \Lambda - \Psi e^{\alpha v_{pv}} \tag{5}$$

The subsystems that make up the microgrid must be managed and controlled in such a way as to ensure energy balance and correct operation. Therefore, in addition, to (3) and (4), there is also a set of equations that describe the dynamic performance of the control subsystems, i.e.

$$\begin{aligned} u_k = g_k(&\frac{d^{n_u}u_k}{dt}, \frac{d^{n_u-1}u_k}{dt}, \cdots \\ &\frac{du_k}{dt}, \frac{d^{n_v}v_i}{dt}, \frac{d^{n_v-1}v_i}{dt}, \dots, \\ &\frac{dv_i}{dt}, v_i, \frac{d^{n_j}i_j}{dt}, \frac{d^{n_i-1}i_j}{dt}, \cdots, \frac{di_j}{dt}, i_j, \rho_k) \end{aligned} \tag{6}$$

where $u_k$ represents the k-th control signal for one of the elements of the microgrid that is generated by the control subsystem k and where $\rho_k \in \{\rho_{k1}, \rho_{k2}, ...\}$ are its parameters. Note that, unlike the expressions in (3) and (4), the function $g_k(\cdot)$ can depend on the derivatives of the input voltages, currents, and signals, which means that the expression (6) can be described as a system of nu first-order differential equations. Madduri in [16] shows an interesting case using the previously described microgrid bus architecture. It was developed for scalable stand-alone microgrids for rural electrification using photovoltaic generators.

The sources, loads and elements interconnecting the buses are managed by a power processing unit which, given their high efficiency, consists of power MOSFET circuits that are switched in such a way that the desired conversion process is achieved. The control signals $u_k$ control these power units and represent duty cycles converted to discontinuous signals employing a pulse width modulator. The discontinuous signals, which have a switching frequency of 100 kHz up to 1 MHz, are responsible for turning the transistors on and off. This activation process generates a ripple in the currents and voltages of the system that can provide helpful information for its management, as indicated in [17]. Furthermore, if these equations are solved in real-time, it is possible to visualize and analyze ripple signals at all points of the microgrid.

Nevertheless, the development of digital models of the electrical microgrids described by means of the equations (1), (2), (3), (4) and (6) presents important challenges. Firstly, the mathematical model of the microgrid can be described by a set of differential equations of order $n + m + n_u$, where even for a system with few generators and loads, one can have a high order system. For example, a microgrid with two power sources and two independent loads can be described, in its simplest configuration, by a 6th-order differential equation. Notice that to obtain accurate results, it is necessary that the integration step for the numerical algorithm that must solve the system of algebraic differential equations must be at least ten times smaller than the period of the carrier signal. That is, it must be equal to or less than 1 microsecond. Additionally, the emulation time of the digital microgrid will depend on the characteristics of the connected generators and loads. For example, for a microgrid based on photovoltaic generators with resistive loads, the emulation time should be at least 10 minutes to see all the transient processes of the load balance. It is also desirable to perform a 24 to 48-hour analysis to evaluate the load balance on predetermined days. Microgrids based on photovoltaic generators can be described using a set of first-order differential equations and a set of algebraic equations. The number of differential equations depends on the number of elements present in the grid, i.e., generators and loads. Therefore, the numerical solution of this set of differential equations must focus on the parallelization of the processes.

## 3. Simulation approach

In this paper, we develop an FPGA-based simulation system that can simulate a PV-based microgrid over a wide range of time domains. In this way, we can capture the ripple in currents and voltages, the transients due to the capacitor and inductors, and observe the characteristics of the system energy balance, the state of charge of storage elements and the response to different input vectors of the photovoltaic generation system and loads. In other words, we need a system that is able to solve equations (1), (2), (3), (4) and (6).

To construct the proposed simulation system, we use the architecture shown in Figure 1, which consists of n number of PV arrays as a power source, with a respective boost converter each. All boost converters feed a single voltage distribution bus $V_H$ with voltages between 360 V and 400 V. Connected to this bus are a $m$ number of load branches, each feeding a buck converter, called distribution nodes, with an intermediate voltage $V_M$ between 45 V and 50 V. These, in turn, feed a l number of buck converters, where each converter is a power management unit (PMU) connected to the load and storage system in the form of 12 V electrochemical batteries through a low voltage bus $V_L$.

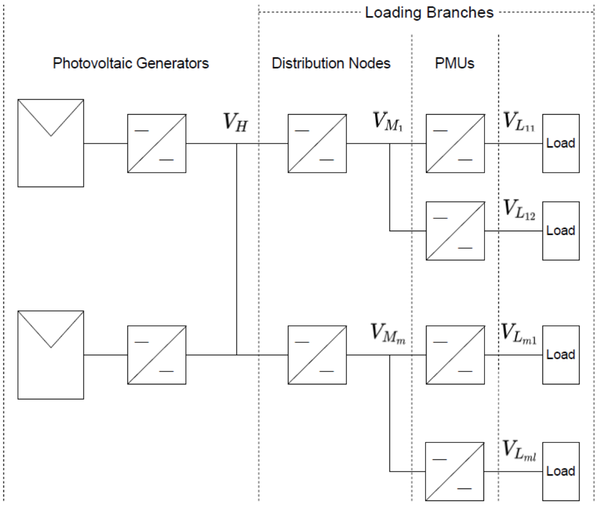

Fig. 1. Multi-bus microgrid architecture based on [16].

Without a loss of generality, we first analyzed the simplest case of the chosen architecture, i.e., a single power source and a single loading branch. For this case, we obtain a system with the seven first-order differential equations (7) to (13); without accounting for the equations of the switching control algorithms, which may be of higher order. Although this is the simplest possible case, one can have multiple load branches, each of which can have several PMUs with their respective loads. Thus, the number of elements that make up the microgrid can grow by one or more orders of magnitude, along with the number of equations of the system. The task of solving all the dynamic equations could quickly become cumbersome.

$$\frac{dV_{PV}}{dt} = \frac{i_{PV} - i_{L_1}}{C_1} \tag{7}$$

$$\frac{di_{L_1}}{dt} = \frac{V_{PV} - V_H(1 - u_1)}{L_1} \tag{8}$$

$$\frac{dV_H}{dt} = \frac{(1 - u_1)i_{L_1} - u_2 i_{L_2}}{C_2} \tag{9}$$

$$\frac{di_{L_2}}{dt} = \frac{u_2 V_H + V_M}{L_2} \tag{10}$$

$$\frac{dV_M}{dt} = \frac{i_{L_2} - u_3 i_{L_3}}{C_3} \tag{11}$$

$$\frac{di_{L_3}}{dt} = \frac{u_3 V_M - V_L}{L_3} \tag{12}$$

$$\frac{dV_L}{dt} = \frac{i_{L_3} + i_{bat} - i_R}{C_4} \tag{13}$$

Decomposing and describing the microgrid through a series of blocks representing the different elements that compose it (energy sources, power converters, loads, and energy storage) allow us to scale up the system later and represent a more extensive complex system. Code-wise, each block is represented as an object or a class composed of its electrical parameters and the dynamic equations that describe its behavior. In such a way, each block has a single dynamic equation and is interconnected with adjacent blocks according to the microgrid architecture. For each calculation iteration, the blocks' equations are solved separately, and then the resulting currents and voltages are propagated to the neighboring blocks. For the photovoltaic generators, the equivalent circuit equations of the 1D2R model [18] were used.

### *3.1 Hardware implementation*

The simulation program was implemented using Python and structured linearly and sequentially. So first, the architecture of the microgrid to be simulated is defined, where all the objects/classes representing the different elements of generation, power converters, storage, and loads are created. Next, the simulation parameters corresponding to time variables and a constant integration step are defined. Then the initial conditions of the capacitor's voltages, the inductor's currents, and the state of charge of the batteries are defined.

The next stage starts the simulation process in a loop that stops when the final simulation time is reached. During the simulation, the program reads an input vector that includes the loads' solar irradiance intensity and magnitude. It then proceeds to sequentially solve the dynamic equations of all the elements of the microgrid. For the solution of the equations, the explicit integration method is used [19], which was chosen for its ease of calculation and also because it allows solving for each element of the microgrid individually for each iteration. Once all the microgrid elements have been solved for the current iteration, the results of each element are propagated throughout the microgrid, and then the next iteration is initiated.

However, this sequential approach was deemed as the cause of the high execution time of the simulation program as it forces each element to be calculated only after the previous one finishes solving its respective dynamic equations. A hardware implementation with an FPGA allows the creation of independent cores for each element, thus allowing to solve of all the equations of the microgrid at the same time. The result propagation stage is preserved such that when all elements finish solving for the respective equations, the resulting values are shared between them.

We use a Zedboard development board from Xilinx with a Zynq®-7000 SoC for the implementation. The FPGA chip XC7Z020-CLG484-1 contains 106400 flip-flops logic, 53200 LUTs, 220 DSP48E blocks and 560 KB of dual-port BRAMs. The communication within the processing system (PS fabric) and the programmable logic (PL fabric) was handled through an AXILITE interface. The communication with an external computer for data extraction utilised a UART serial port limited to a maximum baud rate of 115200 Bd/s by the drivers of said computer. All the system's values were handled as 32-bit floating point variables.

## 4. Simulation results and discussion

Multiple different comparative tests were conducted to assess the hardware accelerated program's capabilities:

- A validation test confirms that the program returns accurate results compared to benchmark software and the Python implemented program running the same simulation algorithms.
- Four simulations of microgrids with different elements were carried out to measure the FPGA's performance based on its execution time; the results are likewise compared to the Python program.
- The possibility of real-time simulation is evaluated, and compromises are required.

Finally, the FPGA hardware resource usage is discussed.

### *4.1 Hardware results validation*

A comparison with the results obtained in the software PowerSim (PSIM) and the results obtained with the program previously developed for Python was carried out to validate the accuracy of the accelerated hardware system. PSIM was used as a benchmark since it is designed to simulate power circuits and is a prevalidated software for commercial use.

Initially, a simulation of the microgrid circuit was performed for a time of 1 s and with a step of 0.1 µs to capture the transient period of microgrid start-up and subsequent stabilization.

However, the FPGA results presented an unnatural oscillation, as shown in Figure 2, which was caused by the accumulation of errors in the integration method used.

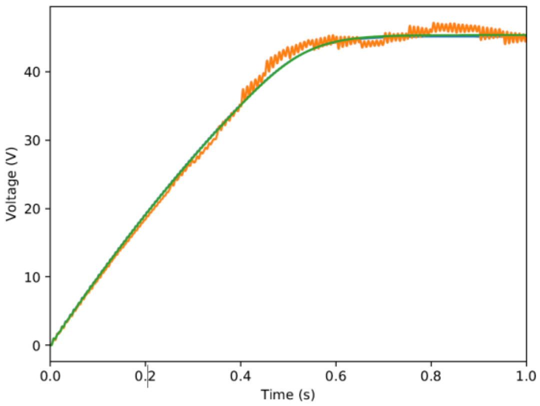

Fig. 2 Oscillation in bus VM due to integrator error. Python (blue), FPGA (orange), and PSIM (green) simulations.

The simulation step was halved to 0.05 µs to reduce the integration error and eliminate the oscillation. This proved a relatively effective solution, with some caveats discussed further down the line.

The average values during the steady state regime were used to determine the accuracy of the simulation programs. For the microgrid voltages, the maximum absolute errors for PSIM, the Python-based program and the hardware accelerated program was -0.62 %, -0.89 %, and -0.82 % respectively, as shown in Table 1.

In the case of the currents, a maximum absolute error of -1.00 % was obtained for PSIM, -1.28 % for the Python program, and -2.14 % for the FPGA as can be seen Table 2.

For the resulting values, in all but three cases, the absolute error is less than 1 %, two of which correspond to the currents calculated by the FPGA with a maximum error of -2.14 %. Part of this can be attributed to the integration method utilised in combination with the discontinuous equations caused by the high-frequency switching of the power converting elements' MOSFETs.

The voltage curves determined that the resulting behaviour is similar for all three simulation programs, as shown in Figure 3 for VM. However, the oscillations during the transient curve generated by Python and the FPGA are larger in magnitude than those of PSIM. This is because the models simulated in Python are ideal and do not consider the internal resistance of the capacitors and inductors of the system. PSIM, on the other hand, requires the configuration of minimum internal resistance in the milliohms range to facilitate the solver's convergence.

Table 1 Steady state voltage comparison for the simulations.

| Parameter | Theoretical Steady State [V] | PSIM Result [V] | PSIM Error [%] | Python Result [V] | Python Error [%] | HW Result [V] | HW Error [%] |
|---|---|---|---|---|---|---|---|
| VPV | 179.71 | 180.55 | -0.47 | 181.27 | -0.87 | 181.18 | -0.82 |
| VH | 359.42 | 361.10 | -0.47 | 362.54 | -0.87 | 362.37 | -0.82 |
| VM | 44.93 | 45.14 | -0.47 | 45.32 | -0.87 | 45.29 | -0.80 |
| VL | 11.23 | 11.30 | -0.62 | 11.33 | -0.89 | 11.32 | -0.80 |

Table 2 Steady state current comparison for the simulations.

| Parameter | Theoretical Steady State [A] | PSIM Result [A] | PSIM Error [%] | Python Result [A] | Python Error [%] | HW Result [A] | HW Error [%] |
|---|---|---|---|---|---|---|---|
| iL1 | 1.404 | 1.418 | -1.00 | 1.422 | -1.28 | 1.434 | -2.14 |
| iL2 | 5.616 | 5.658 | -0.75 | 5.663 | -0.84 | 5.682 | -1.18 |
| iL3 | 22.463 | 22.601 | -0.61 | 22.659 | -0.87 | 22.640 | -0.79 |

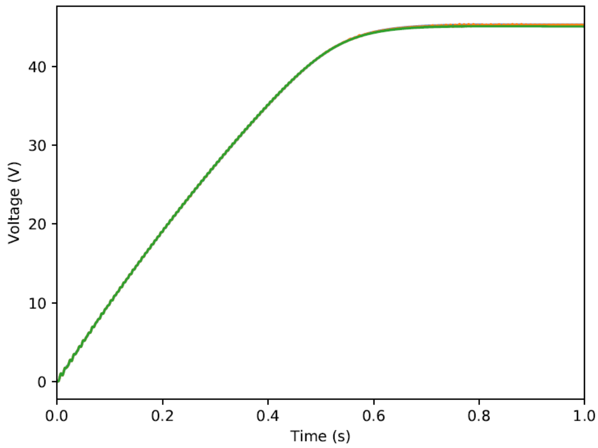

Fig. 3 VM curve comparison for between Python (blue), FPGA (orange), and PSIM (green) simulations

A second reason for the slightly higher error values for the FPGA's results is that unlike PSIM and the Python program, which run on a 64-bit system, this hardware runs on a 32bit system. Even though the FPGA runs the same algorithms as its Python counterpart, the reduced precision can increase the accumulated integration error. This becomes notable when zooming into the voltage curves. Although reducing the integration step was able to eliminate the oscillation shown in Figure 2, the error persists on a smaller scale, as shown in Figure 4.

The oscillation could be further reduced by once again decreasing the step. However, this also increases the execution time of the simulations in a proportionally inverse fashion. A different approach would be changing the integration method like Euler's implicit method or a higher order Runge–Kutta method, which presents better convergence for stiff systems [20].

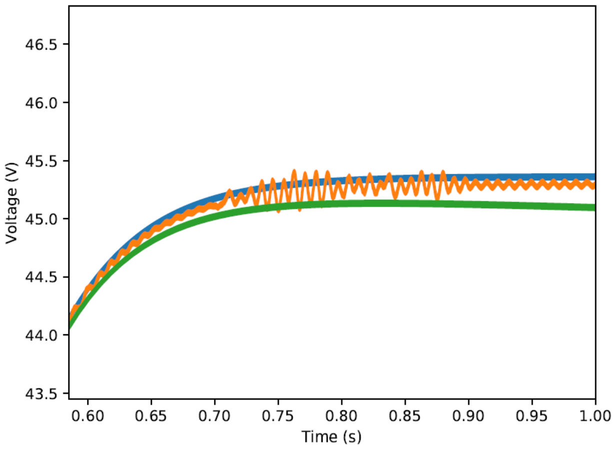

Fig. 4 Oscillation in bus VM due to integrator error. Python (blue), FPGA (orange), and PSIM (green) simulations.

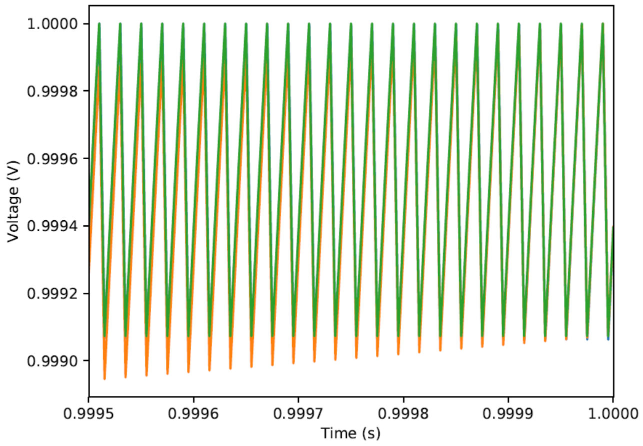

Fig. 5 $V_M$ normalised voltage ripple comparison for 0.9995 s < t < 1 s. Python (blue), FPGA (orange), and PSIM (green) simulations.

Nonetheless, this approach will in turn, increase the hardware resource usage in the FPGA and execution time as they require multiple evaluations of the integration function.

It is worth noting that the FPGA is still capable of simulating relatively closely the system behavior in its smallest time domain. That is to say, the ripple caused by the commutation of the MOSFETs, as is depicted in the normalized curves of Figure 5.

### 4.2 Simulation performance

To determine the performance gains of the simulation program provided by the hardware acceleration, multiple simulation runs were performed utilizing the same microgrid architecture of the validation section but varying the number of loading branches in the system. Simulations for 1, 2, 5, and 10 branches, including the photovoltaic generator, power converters, buses, storage system and loads, were conducted with simulation times of 1 s to 5 s, and a 0.05 μs step. For the cases of the FPGA simulations with 5 and 10 branches, the results are theoretical values calculated based on the number of clock cycles predicted by the Xilinx software, as these configurations could not be implemented due to hardware resource constraints which are discussed in the next section.

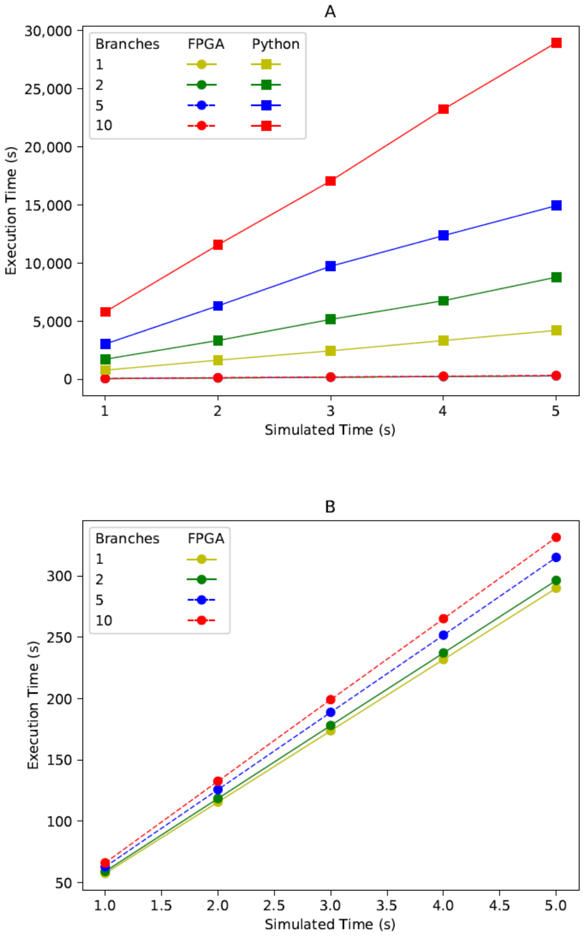

Fig. 6 Execution time comparison of microgrid simulations for a different number of load branches. A) Python vs FPGA B) FPGA zoom.

For both the Python and FPGA programs, the execution time grows linearly with the number of microgrid elements. However, both the rate at which this time increases, which can be observed the graph A of Figure 6, and its causes differ. For the Python program, the increase in execution time is caused by the sequential nature of the program. The dynamic equations for all microgrid elements are solved one after the other. Thus, a proportional time increase is expected as the microgrid grows in size. The FPGA, on the other hand, solves the equations simultaneously, which should mean that execution time is independent of the number of elements in the microgrid, and it is bounded by the microgrid element, which equations take the most time to solve for. However, in graph B of Figure 6, the FPGA execution time does vary with the size of the microgrid. This is caused by the communication between the PS and PL fabrics when updating the input vectors and retrieving the results after every iteration. Nonetheless, the rate at which the FPGA's execution time increases is negligible compared to the Python program. Furthermore, this communication overhead could be reduced by utilizing other communication methods, such as direct memory access (DMA) to the shared PS-PL block ram.

Due to the difference in the execution time increase rate, the average performance gain of the FPGA results over the Python program grows with the microgrid's size, as depicted in Table 3, where the FPGA program can be up to 87.1 times faster than Python for the microgrid with ten load branches. Although as the microgrid grows in size, the FPGA program runs the risk of hitting the hardware resource limitations.

Table 3 Average performance gains of hardware accelerated program vs Python program.

| Number of Load Branches | Performance Gain<br>(Python Time / FPGA Time) [s / s] |
|---|---|
| 1 | 14.2 |
| 2 | 28.9 |
| 5 | 49.3 |
| 10 | 87.1 |

### 4.3 Real-time simulation

To achieve real-time simulation capacity, the FPGA program required a modification to the power converter dynamic equations. Up to this point, the control signals uk for the MOSFETs had been modelled as PWM signals, that is, a square wave with defined oscillation frequency, duty cycle and possible discrete values of 0 or 1. However, this discontinuous signal, in conjunction with the high frequency of its commutation, makes for the system's stiffness, which requires the simulation program to use very small simulation steps for the integration method used to converge, and by consequence, increase the execution time of the program.

The PWM signal was substituted for a continuous range of values, where $u_k \epsilon [0,1]$, represents an equivalent duty cycle. This effectively gets rid of the commutation frequency parameter and reduces the stiffness of the system, allowing a bigger simulation step to be used and, in turn, decreasing the execution time.

With a step of 5.86 µs, real time-simulation was achieved for the microgrid configuration with one load branch. The resulting curves after were very similar to the ones obtained in the validation section, where the absolute errors of the steady state values were within 0.05 % of the results obtained previously with a 0.05 µs step and the PWM signal. However, there was a notable difference during the transient. The natural oscillations in the real-time simulation during this period have a greater magnitude. They grow with time, reaching a size 2.7 times larger and subsequently decrease as the stability point is reached, as shown in Figure 7. This means that the system still is affected by stiffness issues, even without the PWM, but is capable of adequately stabilizing after a transient.

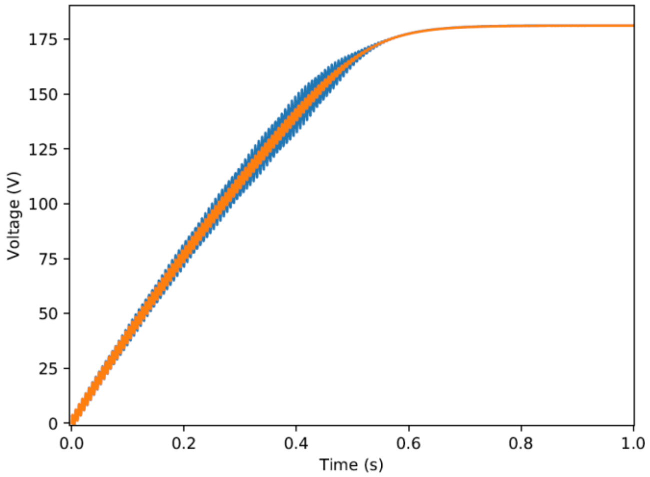

Fig. 7 Comparison of FPGA results for VPV . With PWM and 0.05 µs step (orange), without PWM and 5.86 µs step (blue)

These results reveal that it is possible to achieve real-time simulations with two compromises that could be accepted or rejected depending on which electrical behaviors are being studied, those being the loss of accurate curves with the ripple produced by the MOSFETs commutation and a reduction in accuracy during transients. Further research into this area is required to determine if the compromises of this system are admissible for the application of microgrid monitoring and diagnostics through the implementation of a digital twin, as presented in [21].

### *4.4 Resource usage*

As it was previously mentioned, it was not possible to implement the 5 and 10-load branch microgrid configurations within the resource constraints of FPGA available for the experiments. In Table 4, resource usage percentage is presented, where the DSP blocks and LUTs are the limiting factors in the five branch case and the FF elements are also included for the ten branch microgrid.

Table 4 Hardware resource usage percentage

| Load branches | BRAM | DSP48E | FF | LUT |
|---|---|---|---|---|
| 1 | 1 | 22 | 11 | 33 |
| 2 | 4 | 61 | 26 | 79 |
| 5 | 8 | 146 | 62 | 194 |
| 10 | 15 | 280 | 118 | 361 |

The easiest solution to the limited hardware resources would be to use a bigger FPGA, as long as budget constraints allow. A solution within the scope of the current equipment is to have a more stringent resource allocation depending on the precision required by the system. Currently, all variables and operations are handled as 32-bit floating points. The integer part of all currents, voltages and other system variables can be limited to only the number of digits necessary to represent the expected values. The same can be applied to the decimal part of all values. Conversely, this could allow for greater precision when required by allocating more bits to the decimal part, and execution time could be reduced by having fixed point variables instead of floating-point ones.

Another possibility is the implementation of the simulation program divided between multiple smaller FPGAs, like the system described in [22]. This, however, introduces variables such as FPGA synchronization and cross-communication.

Lastly, in the case of real-time simulation, specifically for monitoring applications, FPGAs could be deployed as digital twins for critical sections or elements of a microgrid instead of simulating the whole system.

## 5. Conclusions

A simulation program for isolated microgrids in a wide range of time domains was implemented in an FPGA to reduce calculation time. The results of the hardware accelerated program were validated by comparison with the PSIM and the previously developed Python simulation program. The FPGA required a simulation step of 0.05 µs, half of what was used initially in its Python counterpart, obtaining a maximum absolute error of 2.14 %.

A comparison between Python and the FPGA implemented a program of execution time when simulating microgrids of different sizes was performed. The FPGA was 14.2 to 87.1 times faster than Python, where the performance gains increased as the microgrid grew in a number of elements.

A real-time simulation was achieved by substituting the PWM of the control signals for the MOSFETs with a continuous value equivalent to the duty cycle and by increasing the simulation step to 5.86 µs. The absolute errors of the steady state values were within 0.05 % of the FPGA values when using the PWM, and a decrease in accuracy was perceived during the transient, where oscillations were up to 2.7 times larger. Further research is proposed to determine if this system can be used for real-time monitoring and diagnostics of microgrids using a digital twin.

## References

[1] A. T. Elsayed, A. A. Mohamed, and O. A. Mohammed, "Dc microgrids and distribution systems: An overview," *Electric Power Systems Research*, vol. 119, pp. 407–417, 2 2015.

[2] R. H. Lasseter, "Microgrids," in *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings*, vol. 1. IEEE, 2002, pp. 305–308.

[3] I. Tank and S. Mali, "Renewable based dc microgrid with energy management system," in *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2015, pp. 1–5.

[4] E. Unamuno and J. A. Barrena, "Hybrid ac/dc microgrids—part i: Review and classification of topologies," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1251–1259, 12 2015.

[5] ——, "Hybrid ac/dc microgrids—part ii: Review and classification of control strategies," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1123–1134, 12 2015.

[6] D. E. Olivares, C. A. Cañizares, and M. Kazerani, "A centralized energy management system for isolated microgrids," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1864–1875, 2014.

[7] A. Alzahrani, M. Ferdowsi, P. Shamsi, and C. H. Dagli, "Modeling and simulation of microgrid," *Procedia Computer Science*, vol. 114, pp. 392 – 400, 2017, complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.

[8] N. C. Khadepaun, N. Shah *et al.*, "Operation of solar pv with pem fuel cell for remote hybrid microgrid," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE, 2020, pp. 1–6.

[9] G. Hu, S. Li, C. Cai, Z. Wu, and L. Li, "Study on modeling and simulation of photovoltaic energy storage microgrid," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2017, pp. 692–695.

[10] M. Alam, K. Kumar, J. Srivastava, and V. Dutta, "A study on dc microgrids voltages based on photovoltaic and fuel cell power generators," in *2018 7th International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE, 2018, pp. 643–648.

[11] S. S. Sheikh, S. Iqbal, M. Kazim, and A. Ulasyar, "Realtime simulation of microgrid and load behavior analysis using fpga," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–5.

[12] B. Zhang, S. Fu, Z. Jin, and R. Hu, "A novel fpga-based real-time simulator for micro-grids," *Energies*, vol. 10, no. 8, 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/8/1239

[13] J. Xu, K. Wang, P. Wu, and G. Li, "Fpga-based submicrosecond-level real-time simulation for microgrids with a network-decoupled algorithm," *IEEE Transactions on Power Delivery*, vol. 35, no. 2, pp. 987–998, 2020.

[14] E. M. Brenes and C. Meza, "An application-specific instruction set processor for microgrid simulation," in *2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*, 2019, pp. 1–6.

[15] R. Zárate-Ferreto, K. Alfaro-Badilla, C. Meza-Benavides, C. Salazar-García, and A. Chacón-Rodríguez, "Soc-fpga implementation of a temperature-dependent parameters estimator for photovoltaic generators," in *2020 IEEE 11th Latin American Symposium on Circuits Systems (LASCAS)*, 2020, pp. 1–4.

[16] P. A. Madduri, J. Poon, J. Rosa, M. Podolsky, E. Brewer, and S. Sanders, "A scalable dc microgrid architecture for rural electrification in emerging regions," in *2015 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2015, pp. 703–708.

[17] C. Meza and R. Ortega, "On-line estimation of the temperature dependent parameters of photovoltaic generators," *IFAC Proceedings Volumes*, vol. 46, no. 11, pp. 653–658, 2013.

[18] N. Araújo, F. Sousa, and F. Costa, "Equivalent models for photovoltaic cell–a review," *Revista de Engenharia Térmica*, vol. 19, no. 2, pp. 77–98, 2020.

[19] S. Chakraverty, N. Mahato, P. Karunakar, and T. Rao, *Advanced Numerical and Semi-Analytical Methods for Differential Equations*. Wiley, 2019.

[20] "Chapter 3 - modeling formalisms and their simulators," in *Theory of Modeling and Simulation*, 3rd ed., B. P. Zeigler, A. Muzy, and E. Kofman, Eds. Academic Press, 2019, pp. 43–91.

[21] J. Xiong, H. Ye, W. Pei, K. Li, and Y. Han, "Realtime fpga-digital twin monitoring and diagnostics for pet applications," in *2021 6th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2021, pp. 531–536.

[22] P. Li, Z. Wang, C. Wang, X. Fu, H. Yu, and L. Wang, "Synchronisation mechanism and interfaces design of multi-fpga-based real-time simulator for microgrids," *IET Generation, Transmission & Distribution*, vol. 11, no. 12, pp. 3088–3096, 2017.

# General discussion

To validate the proposed component-wise model for microgrid simulation and determine the performance of the simulation program in a general-purpose processor, a comparison was proposed against a reference program pre-validated for power electronics simulation (PSIM), and two other tests to establish the capacity to simulate multiple time domains and measure the execution time of the developed program. In line with this objective, three tests were conducted using the microgrid architecture proposed by Madduri et al. (2015). The parameters used for the PV array, power converters and loads are shown in annex A.

First, both programs were fed with the same architecture, input vectors and time step of 0.1 µs. The simulation ran for 1 s to capture the transient of the microgrid startup and the settling of the system. Comparing the resulting voltage curves, both programs returned almost identical results. During the system transient, PSIM's oscillations are more attenuated as the software required the inclusion of internal resistance of the capacitive and inductive components for the solver to converge. Python's program utilized ideal component models, resulting in greater magnitude oscillations. However, these oscillations settle down once the steady state is reached where the curves of both programs are practically superposed with a 0.2 % steady state difference between each other. Using PSIM's results as a benchmark, Python presents a maximum absolute error of 0.65 % for the voltage curves and 0.63 % for the settling time during the transient.

Additionally, when comparing the average steady state values expected from the system, which are obtained by equaling the left side of equations (3) and (4) to zero and solving the resulting system of equations, both Python and PSIM had approximately a 1% error from the theoretical steady state. Python's error was within the same range as the benchmark software, which was considered an acceptable result that called for further tests.

For the second test, the simulation time was extended from 1 s to 600 s to capture not only the voltage ripple in the microsecond time domain, but the change in the state of charge of the battery in the scale of multiple minutes simultaneously. By varying the load at the 200 s and 400 s marks, the battery responded by discharging energy into the microgrid when the load increased to 630 W, and charging when the load was subsequently reduced to 80 W. It is notable that this simulation had an execution time of about 3.6 days. This leads into the third test which was to measure the performance of the simulation program.

By changing the amount of microgrid load branches, based on the same architecture, the execution time for 1 second of simulation was determined to be 516 s with 1 load branch, up to 3788 s with 10 branches. The growth in simulation time is not quite a one-to-one relation, however this test demonstrated that the developed program's performance is not ideal if a quick results turnout is required. There are multiple factors that contribute to the high simulation time. First is the programing language itself, Python is an interpreter suited for prototype programs where low development time and programing flexibility are required, but the same cannot be said for high performance and code optimization. Secondly, the program was set to solve the microgrid's differential equations in a sequential fashion. There is wasted potential by not solving them as parallel calculations. The third reason is based on Euler's integration method which requires a small integration step to converge into a solution. This also brings forth one of the weaknesses of the simulation approach, by trying to simulate multiple time domains orders of magnitude apart at the same time, the program is forced to solve the equations millions of times for each simulated second.

A hardware implementation was selected as an option to improve the execution time of the simulation program without sacrificing the capacity to simulate multiple time domains. After converting the program's code to C language and running it through a High-Level Synthesis software, the program was revalidated with the same methodology as previously described. An increase in error was perceived, as well as oscillations caused by integration error due to system stiffness. This was likely caused by the loss of precision when going from a 64-bit system in the general-purpose processor to a 32-bit system in the FPGA implementation. By reducing the simulation step to 0.05 μs the error oscillations were attenuated. The maximum error during the transient when comparing with PSIM was 1.46 % for the voltage curves and 0.86 % for the settling time. And the maximum absolute steady state error was 2.14 %.

After the validation, a performance comparison between the Python and FPGA programs resulted in an execution time 14.2 to 87.1 times faster on the FPGA system. Figure 3 shows that the FPGA is at least one order of magnitude faster than Python, and the rate which execution time increases with the number of load branches is greater also lower for the FPGA.

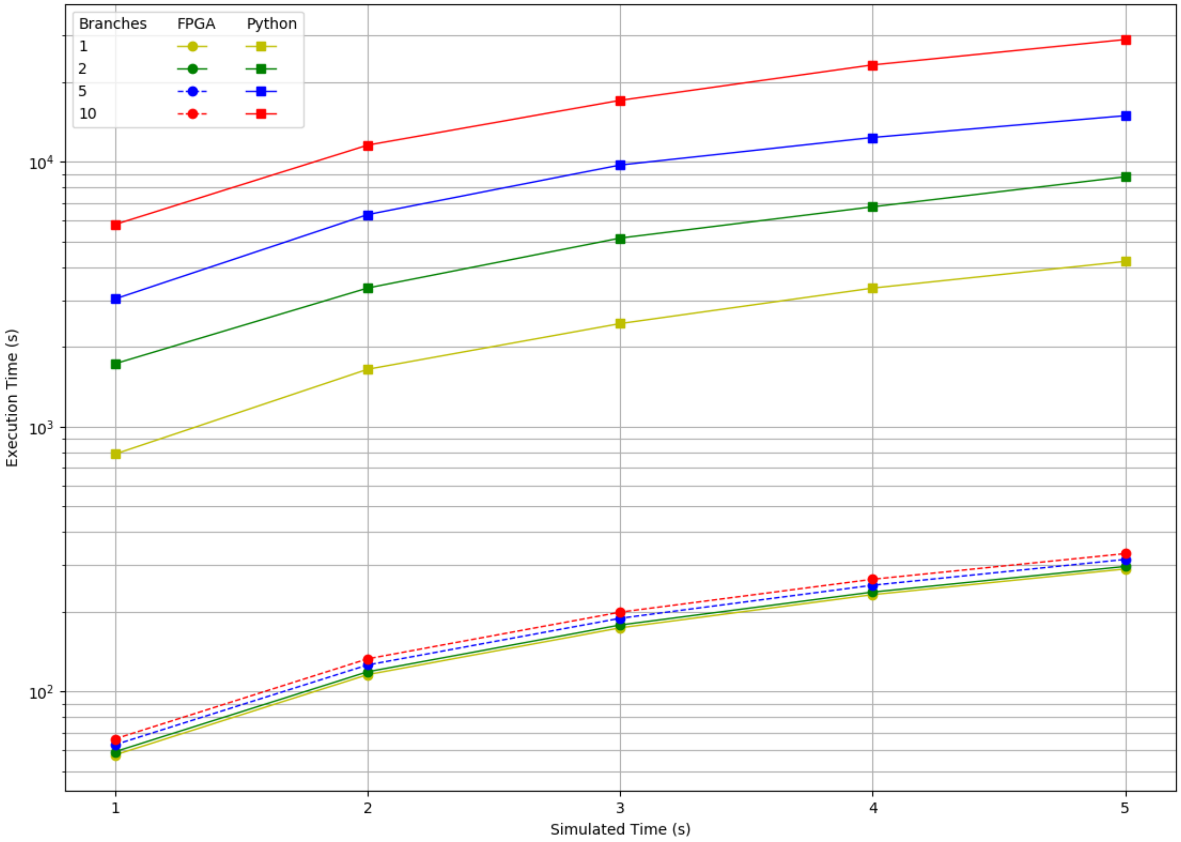

Figure 3. Execution time vs. simulated time by number of load branches.

Furthermore, by modifying the system's equations to utilize a constant value instead of a PWM for the control signals, the integration step could be increased to 5.86 μs to achieve real-time simulation. This however means that the model, and the resulting values by consequence, are not as close to real values as with the model utilizing PWM signals. However, real-time simulation can be an enticing characteristic worth the compromise of a less accurate model some scenarios such as system monitoring and diagnosis via digital twin as described in Xiong (2021).

A new issue that became apparent was the FPGA's resource usage. It was not possible to implement the microgrid with 5 and 10 load branches at their footprint demanded more hardware resources than those available. This issue can be mitigated by optimizing the resource allocation, using a multi-FPGA system, or simply using a board with more hardware resources. However, this topic requires further research.

As a final notes, all the experiments were performed as a case study of a general model, as described in the methodological framework of this thesis, to validate the simulation program through a comparison of its output data against a benchmark. The same methodology can be used for any isolated DC microgrid and its elements as long as they can be modeled as a system of equations described in equations 3, 4, and 5 of the theoretical framework.

## Global conclusions

Based on the presented research, and the objectives initially established the following conclusions were derived:

1. The simulation program was developed with a component-wise approach, and then validated through a comparison with the benchmark software PSIM. By simulating a predefined microgrid architecture and comparing both the transient and steady state results, an absolute error of approximately 1% was obtained and the ability to simultaneously simulate the time domain in microseconds range and several minutes range was demonstrated. The execution time was confirmed as a bottleneck for the simulation approach due to the nature of the programming language and the sequential approach to solving the microgrid's equations.
2. The simulation program was ported to C language and later implemented in an FPGA board with the use of HLS. The hardware implementation was validated with the same methodology as the Python counterpart and a maximum absolute error of 2.14% was obtained.
3. The performance gain of the hardware accelerated simulation program was measured to range from 14.2 to 87.1 times faster than Python depending on the size of the microgrid, where the relative execution time reduction became greater as the microgrid increased in number of elements.
4. Real-time simulation was achieved by increasing the simulation step and substituting the PWM signal for MOSFET commutation with a continuous value of the equivalent duty cycle. Steady state values were within 0.05% of the PFGA's results obtained previously and an increase of the transient oscillations was measured up to 2.7 times larger.

## Recommendations

Given the nature of this research as a first incursion into the subject of microgrid simulation, there is still several issues to resolve and improve upon. Firstly, the simulation program that was developed currently works for open-loop systems. It is necessary to implement additional functions for control algorithms that will result in the simulation of closed-loop systems. This will allow the program to double as an assisting tool for control algorithm design. The same programming methodology utilized in this research can be reproduced first implementing with Python, then converting the program to C language and subsequently synthesizing a hardware equivalent.

Secondly, the catalogue of microgrid elements that can be simulated should be expanded to include a wider range of energy sources, not just photovoltaics, as well as different loads and power conversion elements. This improvement should also include elements capable of working with AC current like transformers, rectifiers, and inverters, to bring the program closer to a generalized microgrid simulation tool that can work both with grid-tied and isolated systems. Inductive and capacitive loads should be included, and further tests must be performed; current spikes, generated by loads such as motors during startup are of special interest, as these must be accounted for during a microgrid's design stages.

Next, when the previous recommendations are implemented, the program validation must be repeated. Additionally, the current program was validated by comparing simulation results to the PowerSim software, which was considered as a valid reference given its wide commercial use as a dynamic system simulation tool, and specifically by its capability to simulate the transient behavior of a system. It would be a valuable to compare against other specialized software like HOMER which is used globally for microgrid design optimization. However, a different comparison approach should be applied as this software can reach a 1-minute time step at best; in contrast to the developed program which can go down to the microsecond scale. This difference will most likely require limiting the comparison to the power balance of a determined microgrid, however, this could also help to show the benefits of the developed program which can simulate a wide range of time domains, down to the microseconds range.

Additionally, hardware optimization is required to improve the performance from a time-wise point of view, and a resource usage perspective. The current hardware implementation can perform real-time simulation. However, the accuracy of the results had to be compromised; especially during the transient. Pipelining the program sequence might result in lower overall calculation time. Another possible optimization is changing the way the output values of the photovoltaic generators and energy storage units are calculated. Currently these values are calculated with arithmetic functions, but these could be substituted for lookup tables containing a limited number of discrete output values associated to discrete inputs; intermediate values can be calculated through interpolation. This method could save time, as these operations are significantly faster, and thus would allow for smaller simulation steps sizes which in turn will improve accuracy without losing the real-time simulation capability.

It would be valuable to try different mathematical methods to solve the differential equations for a microgrid. Euler's Explicit method was utilized due to its simplicity, but further testing is required to determine if it is indeed the most efficient method in every scenario, and which specific cases could benefit from a more sophisticated solver.

Finally, the ideal use case for the hardware implementation of the simulation program is as a real-time digital twin of an actual microgrid. A comparative test against a physical microgrid should be designed to test if the developed system can indeed function as a monitoring and diagnostics tool for microgrids.

# References

Araya, M., Meza, C., Salazar-García, C., & Chacón-Rodríguez, A. (2022). Microgrid Simulation Datasets (Version v1) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.6342357

Clastres, C. (2011). Smart grids: Another step towards competition, energy security and climate change objectives. Energy Policy, 39(9):5399–5408.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación (6a. ed. --.). México D.F.: McGraw-Hill.

Hirsch, A., Parag, Y., and Guerrero, J. (2018). Microgrids: A review of technologies, key drivers, and outstanding issues. Renewable and Sustainable Energy Reviews, 90:402–411.

Hu, G., Li, S., Cai, C., Wu, Z., and Li, L. (2017). Study on modeling and simulation of photovoltaic energy storage microgrid. In 2017 4th International Conference on Information Science and Control Engineering (ICISCE), pages 692–695. IEEE.

IEA (2014), Key World Energy Statistics 2014, OECD Publishing, Paris, https://doi.org/10.1787/key_energ_stat-2014-en.

IRENA (2020). Renewable Power Generation Costs in 2019. International Renewable Energy Agency, Abu Dhabi.

Jung, J. and Villaran, M. (2017). Optimal planning and design of hybrid renewable energy systems for microgrids. Renewable and Sustainable Energy Reviews, 75:180–191.

Khadepaun, N. C., Shah, N., et al. (2020). Operation of solar pv with pem fuel cell for remote hybrid microgrid. In 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pages 1–6. IEEE.

Madduri, P. A., Poon, J., Rosa, J., Podolsky, M., Brewer, E., and Sanders, S. (2015). A scalable dc microgrid architecture for rural electrification in emerging regions. In 2015 IEEE Applied Power Electronics Conference and Exposition (APEC), pages 703–708.

McNichol, T. (2011). AC/DC: The Savage Tale of the First Standards War. Wiley.

Meza, C. and Ortega, R. (2013). On-line estimation of the temperature dependent parameters of photovoltaic generators. IFAC Proceedings Volumes, 46(11):653–658.

MINAE (2018). National decarbonization plan 2018-2050. https://unfccc.int/sites/default/files/resource/NationalDecarbonizationPlan.pdf.

MINAE (2019). II informe bienal de actualización. Technical report, MINAE, http://cglobal.imn.ac.cr/documentos/publicaciones/BUR2019/offline/IBA-2019.pdf.

Mitsubishi Electric. (2012). Photovoltaic Modules. MLT Series, Mitsubishi Electric Corporation, https://www.mitsubishi-pv.de/datasheets/mlthc-datasheet.pdf

Moharm, K. (2019). State of the art in big data applications in microgrid: A review. Advanced Engineering Informatics, 42:100945.

Naval, N. and Yusta, J. M. (2021). Virtual power plant models and electricity markets - a review. Renewable and Sustainable Energy Reviews, 149:111393.

Olivares, D. E., Cañizares, C. A., and Kazerani, M. (2014). A centralized energy management system for isolated microgrids. IEEE Transactions on Smart Grid, 5(4):1864–1875.

Oshana, R. (2012). DSP for Embedded and Real-Time Systems. Expert guide. Elsevier Science.

Phurailatpam, C. (2015). Embracing microgrids: Applications for rural and urban india. 10th Nationa Conference on ’Synergy with Energy’, pages 49–54.

Planas, E., Andreu, J., Gárate, J. I., de Alegría, I. M., and Ibarra, E. (2015). Ac and dc technology in microgrids: A review. Renewable and Sustainable Energy Reviews, 43:726–749.

Ritchie, H. and Roser, M. (2020). Energy. Our World in Data. https://ourworldindata.org/energy.

Sen, S. and Kumar, V. (2018). Microgrid modelling: A comprehensive survey. Annual Reviews in Control, 46:216–250.

Sheikh, S. S., Iqbal, S., Kazim, M., and Ulasyar, A. (2019). Real-time simulation of microgrid and load behavior analysis using fpga. In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pages 1–5.

Ton, D. T. and Smith, M. A. (2012). The U.S. department of energy’s microgrid initiative. The Electricity Journal, 25(8):84–94.

United Nations (1987). Report of the World Commission on Environment and Development: Our common future. Oxford University Press.

United Nations (2016). The Millennium Development Goals Report 2015. United Nations.

United Nations (2018). The 2030 Agenda and the Sustainable Development Goals: An opportunity for Latin America and the Caribbean. United Nations.

Unamuno, E. and Barrena, J. A. (2015a). Hybrid ac/dc microgrids—part i: Review and classification of topologies. Renewable and Sustainable Energy Reviews, 52:1251–1259.

Unamuno, E. and Barrena, J. A. (2015b). Hybrid ac/dc microgrids—part ii: Review and classification of control strategies. Renewable and Sustainable Energy Reviews, 52:1123–1134.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., and Wesslén, A. (2012). Experimentation in Software Engineering. Computer Science. Springer Berlin Heidelberg.

Xiong, J., Ye, H., Pei, W., Li, K., and Han, Y. (2021). Real-time fpga-digital twin monitoring and diagnostics for pet applications. In 2021 6th Asia Conference on Power and Electrical Engineering (ACPEE), pages 531–536.

Xu, J., Wang, K., Wu, P., and Li, G. (2020). Fpga-based sub-microsecond-level real-time simulation for microgrids with a network-decoupled algorithm. IEEE Transactions on Power Delivery, 35(2):987– 998.

Zhang, B., Fu, S., Jin, Z., and Hu, R. (2017). A novel fpga-based real-time simulator for micro grids. Energies, 10(8).

Zhang, Y., Gatsis, N., and Giannakis, G. B. (2013). Robust energy management for microgrids with high-penetration renewables. IEEE transactions on sustainable energy, 4(4):944–953.

# Annexes

## A. Simulation Parameters

Table 2. Solar irradiance parameters for San Carlos, Costa Rica at coordinates (10.566122°, -084.517136). Taken from Global Solar Atlas (https://globalsolaratlas.info/map).

| Parameter | Value | Unit |
|---|---|---|
| Latitude | 10.56 | ° |
| α | 0.05 | |
| $S_m$ | 1140 | $\mathrm{W/m^2}$ |

Table 3. Photovoltaic array parameters from Mitsubishi Electric (2012).

| Parameter | Value | Unit |
|---|---|---|
| $I_{SC}$ | 8.89 | $\mathrm{A}$ |
| $V_{OC}$ | 37.8 | $\mathrm{V}$ |
| $\alpha_T$ | 0.00056 | $\mathrm{V/°C}$ |
| T | 30 | $\mathrm{°C}$ |
| μ | 1.2 | |
| Cell N° | 120 | |
| $R_S$ | 0 | Ω |
| $R_{SH}$ | ∞ | Ω |
| $S_{STC}$ | 1000 | $\mathrm{W/m^2}$ |
| $T_{STC}$ | 25 | $\mathrm{°C}$ |
| Panels per string | 5 | |
| Strings | 1 | |

Table 4. Microgrid power converter and load parameters.

| Parameter | Value | Unit |
|---|---|---|
| $C_1 - C_3$ | 1 | mF |
| $L_1 - L_3$ | 1 | mH |
| $R$ | 0.5 | Ω |
| $u_1$ | 0.5 | |
| $u_2$ | 0.125 | |
| $u_3$ | 0.25 | |
| $f$ | 50 | kHz |