SysML methodology for HIL implementation of PV models

A. Gutierrez, H. Chamorro, Luiz Fernando Lavado Villa, J. Jimenez, Corinne Alonso

SysML Methodology for HIL Implementation of PV Models

A. Gutierrez[1−3], H.R. Chamorro[2], L.F.L. Villa[3−4], J.F. Jimenez[1], C. Alonso[3−4]

[1]Universidad de los Andes, Department of Electrical and Electronic Engineering, Bogotá, Colombia

[2]KTH Royal Institute of Technology, Stockholm, Sweden

[3]Univ. de Toulouse, Univ. Paul Sabatier, LAAS, F-31400, Toulouse, France

[4]CNRS; LAAS; 7 Avenue du Colonel Roche, F-31077, Toulouse, France

{a.gutierrez75; fjimenez}@uniandes.edu.co, hr.chamo@ieee.org, {luiz.villa; alonsoc}@laas.fr

Keywords

<<Photovoltaic panel>>, <<Hardware-in-the-Loop>>, <<FPGA>>, <<SysML>>, <<Petri net>>.

Abstract

This paper describes a methodology for implementing in FPGA models of photovoltaic panels for Hardware-in-the-Loop (HIL) and real-time simulations. The proposed methodology integrates numerical solutions, SysML diagrams and Petri nets for structural design and formal validation. In this study, photovoltaic cells have been modeled using the single diode circuit. The photovoltaic panel model is solved by the Newton-Raphson method, and the Lagrange remainder is employed to limit the iteration number. Results show suitable accuracy and performance of the proposed methodology.

Introduction

Photovoltaic technologies are growing continuously around the global electricity demand. The increasing interest in photovoltaic (PV) systems with large number of PV panels has developed a need of devices for designing and testing these systems [1]. As a result, PV panel emulators are needed to test photovoltaic systems and devices associated to them [2]. However, several concerns to develop these PV emulators are related to the environmental conditions and the nonlinear loads [3]. This situation encourages the growth of research in PV cell modeling and its hardware implementation for real-time simulations [4].

Hardware implementation of PV systems needs devices with high level of processing and performance. Consequently, applications based on FPGA are taken into account because their characteristics of parallel processing and hardware reconfiguration according to the calculation requirements. However, the high level of complexity in PV systems requires design strategies that allow efficiently implementing these models in FPGA [5]. For these reasons, HiLeS (High Level Specification of Embedded System) is presented as a design methodology for implementing PV models on FPGA [6]. HiLeS provides a hybrid design methodology based on SysML and Petri nets for structural design and implementation.

In this study, a single diode approach has been used to model PV cell under variable temperature conditions. The single diode model is implemented on FPGA employing numerical solutions; additionally, a method has been developed to limit the number of iterations and achieve an efficient use of hardware resources. Moreover, this study employs the HiLeS methodology based on SysML and Petri nets to design and implement PV panel models.

Section II of this document describes the mathematical modeling of PV modules. Section III discusses the numerical method to solve the PV panel model. Section IV shows the methodology to implement the PV panel model on FPGA. Section V describes the model to compensate changes in temperature, and Section VI enumerates the simulation results.

Mathematical model of PV panel

Photovoltaic cells can be modeled using the single diode circuit [7] illustrated in Fig.1. Let I_{ph} be the light generated current, under constant temperature conditions, the I_{ph} is proportional to the solar irradiance S_R thus $I_{ph} = GS_R$. From circuit in Fig.1, the characteristics of the PV cell with temperature constant are given by:

$$G\,S_R - I_0\,(e^{\frac{V_D}{V_t}} - 1) - \frac{V_D}{R_p} - I_{pv} = 0 \tag{1}$$

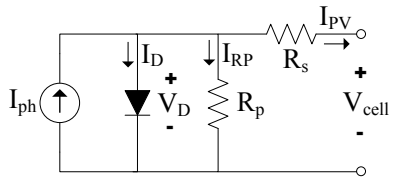


Figure 1: Equivalent circuit of a PV cell.

A PV panel is built by N photovoltaic cells connected in series; therefore, the PV panel voltage is given by:

$$V_{panel} = N\,(\,V_D - R_s I_{pv}\,) \tag{2}$$

From equation (1), V_D can be expressed as:

$$e^{\frac{V_D}{V_t}} = \frac{1}{I_0\,R_p}\,(G\,S_R\,R_p - I_{pv}\,R_p + I_0\,R_p - V_D) \tag{3}$$

Equation (3) is solved in terms of V_D using the Lambert $\boldsymbol{W}$ function [8]; thus, the PV panel voltage is described by (4):

$$V_{panel} = N\,(\,G\,S_R\,R_p - I_{pv}\,R_p + I_0\,R_p - V_t\,\mathbf{W}\left(\frac{I_0\,R_p}{V_t}\,e^{\frac{G\,S_R\,R_p - I_{pv}\,R_p + I_0\,R_p}{V_t}}\right) - R_s\,I_{pv}\,) \tag{4}$$

Numerical solution for the PV panel model

Hardware implementation of analytical PV panel model from (4) is not suitable; as a consequence, a numerical method should be used to solve the equations of the PV panel model. Using the Newton-Raphson method [8], equation (3) can be solved by iteration:

$$V_{D_{n+1}} = V_{D_n} - \frac{f(V_D)}{f'(V_D)} \tag{5}$$

where,

$$f(V_D) = e^{\frac{V_D}{V_t}} - \frac{1}{I_0\,R_p}\,(G\,S_R\,R_p - I_{pv}\,R_p + I_0\,R_p - V_D) \tag{6}$$

thus,

$$V_{D_{n+1}} = V_{D_n} - \frac{V_t\,\left(I_{pv}\,R_p - G\,S_R\,R_p + I_0\,R_p\,e^{\frac{V_{D_n}}{V_t}} - I_0\,R_p + V_{D_n}\right)}{I_0\,R_p\,e^{\frac{V_{D_n}}{V_t}} + V_t} \tag{7}$$

Selection of initial point to quadratic convergence of Newton-Raphson method

Newton-Raphson method shows quadratic convergence given by:

$$|e_{n+1}| = C\,|e_n|^2 \quad , \textit{where}\;\; e_n = \alpha - V_{D_n} \;\; \textit{and} \;\; f(\alpha) = 0 \tag{8}$$

However, a large error in the initial estimate can contribute to non-convergence of the algorithm [9]. In order to avoid the non-convergence of Newton-Raphson method, a sufficiently close initial point should be chosen. We propose to select this initial estimation eq.(9); based on the statistical study of several commercial PV panels. This statistical study showed that relation between the convergence value and the irradiance for typical PV panels can be approximately given by a polynomial relation:

$$V_{d_0} = 0.5 + \left(\frac{0.1\,S_R}{1000}\right) - \left(2.25\,\left(\frac{0.1\,S_R}{1000}\right)^2\right) \tag{9}$$

Polynomial approach of exponential function

Iteration method of equation (7) is suitable for implementation in hardware; however, the implementation of exponential function on FPGA is restricted by concerns of hardware space and performance time. As a result, in this section we propose a method to calculate and implement efficiently the exponential function for a PV panel model. The Taylor polynomial approach of a function $f(x) = e^x$ for $x_0 = 0$ is given by:

$$P_n(x) = \sum_{k=0}^{n} \left(\frac{f^{(k)}(x_0)}{k!} (x - x_0)^k \right) = \sum_{k=0}^{n} \left(\frac{x^k}{k!} \right) \tag{10}$$

Given the Taylor polynomial approach in (10), the error $R_n(x)$ after n terms is written from the Lagrange remainder as:

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x - x_0)^{n+1} \tag{11}$$

where c is a point between x and x_0. Given the interval $0 < c < 1$, the Taylor polynomial approach error is then:

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (1-0)^{n+1} = \frac{e^c}{(n+1)!}, \ \textit{where } c \in (0,1) \tag{12}$$

From (12), the number of Taylor polynomial terms for a given error can be defined before execute the operations. This is an advantage because allows limiting the hardware requirements in the FPGA. For instance, the number of terms for a maximum error of 0.001 is given by:

$$\frac{e^c}{(n+1)!} < \frac{e}{(n+1)!} < 0.001 \quad \textit{where } c \in (0,1), \ \textit{thus } n = 6 \tag{13}$$

Normalizing the exponential function in a PV panel model

Equation (12) allows limiting the amount of terms for the Taylor polynomial representation of an exponential function; however, this assumption is limited to the interval $0 < c < 1$. For the specific case of a PV panel model, the exponential function is given by $e^{\frac{V_{D_n}}{Vt}}$. Where the maximum diode voltage is around $V_{D_n max} = 0.7V$ and the thermal voltage is $V_t = kT/q = 0.025V$; q is the electron charge $[1.602.10^{-19}C]$, k is the Boltzmann constant $[1.38.10^{-23}J/K]$ and the cell temperature T for standard conditions is $[T = 298.15K]$. Thus, $\frac{V_{D_n max}}{Vt} = 28$. For practical implementation of the normalized exponential function, the following procedure is taken:

$$e^{\frac{V_D}{V_t}} = \left(e^{\frac{V_D}{32 V_t}} \right)^{32} \tag{14}$$

In equation (14), the exponent is normalized dividing by 32; in this case the terms is divided by 32 because this is the power of two closer to 28. Using power of two decrease the hardware requirements and the performance time $[\, x.x = x^2 \rightarrow x^2.x^2 = x^4 \rightarrow x^4.x^4 = x^8 \rightarrow x^8.x^8 = x^{16} \rightarrow x^{16}.x^{16} = x^{32} \,]$.

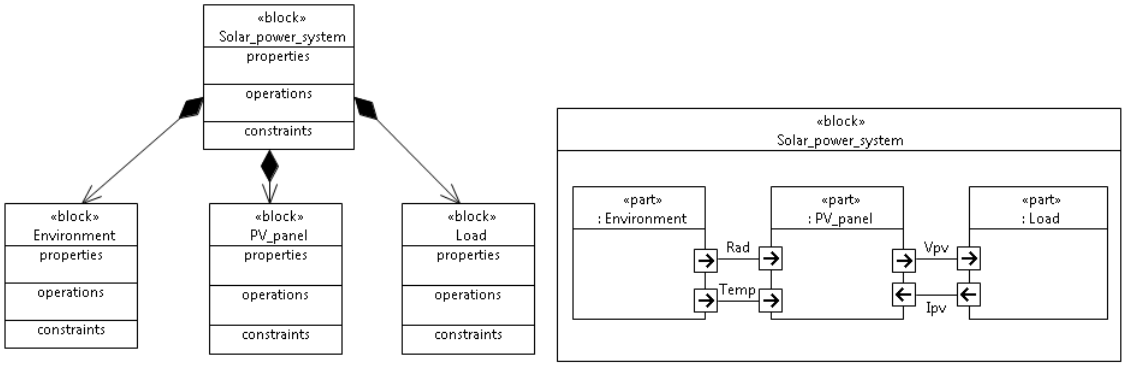
Model compensation for changes in cell operation temperature

The PV cells in an illuminated module operate hotter than the environment temperature. Then, the NOCT(Nominal Operating Cell Temperature), temperature coefficient of current CT_i, and temperature coefficient of voltage CT_v are correction factors given by manufactures to compensate the changes due to cell temperature T_c. Therefore, PV current and voltage can be recalculated by (15). In addition, experimental results show that serial resistance R_s of PV model also changes with cell temperature. From experimental results, we propose the compensated serial resistance given by $R_{sT} = 1.1\, R_s \left(\frac{T_c}{25^oC}\right)$.

$$I_{pvT} = I_{pv} + (CT_i\,(T_c - 25^oC)) \ \ \textit{and} \ \ V_{panelT} = V_{panel} + (CT_v\,(T_c - 25^oC)) \tag{15}$$

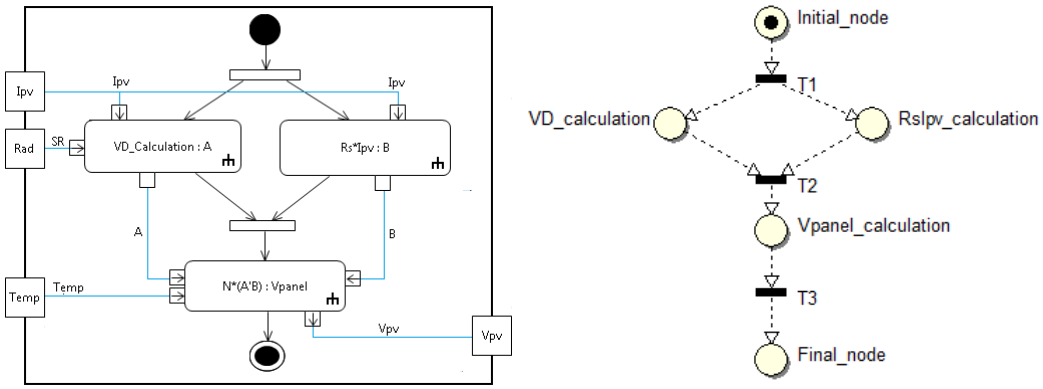
HiLeS-RCP for SysML modeling and Petri net representation of PV panel models

The HiLeS-RCP platform allows representing structurally models using Systems Modeling Language (SysML). In addition, the HiLeS-RCP is used to formal validation of SysML models by means of Petri nets for hardware implementation [10]. Fig.2a, Fig.2b and Fig.3a depict the PV system model using SysML diagrams. Fig.2a represents the system components, and Fig.2b describes the interaction among components. Fig.3a shows the activity diagram, which represents the behavior of the PV panel model. Finally, Fig.3b describes the control Petri net associated to this activity diagram; which is appropriate for implementation on FPGA. The proposed methodology for calculating the exponential function is modeled by SysML diagrams (Fig.4 and Fig.5a). The interaction among elements to calculate this function is described in Fig.4; also, this graph enumerates the hardware requirements to implement the proposed method. Fig.5a represents the calculation sequence, and Fig.5b shows the Petri net associated.



(a) Partial *Block Definition Diagram* of PV system model (b) Partial *Internal Block Diagram* of PV system model

Figure 2: Structural SysML diagrams of PV system model



(a) Partial *Activity Diagram* of PV panel model (b) Partial control Petri net of PV panel model

Figure 3: Behavioral representation of PV panel model

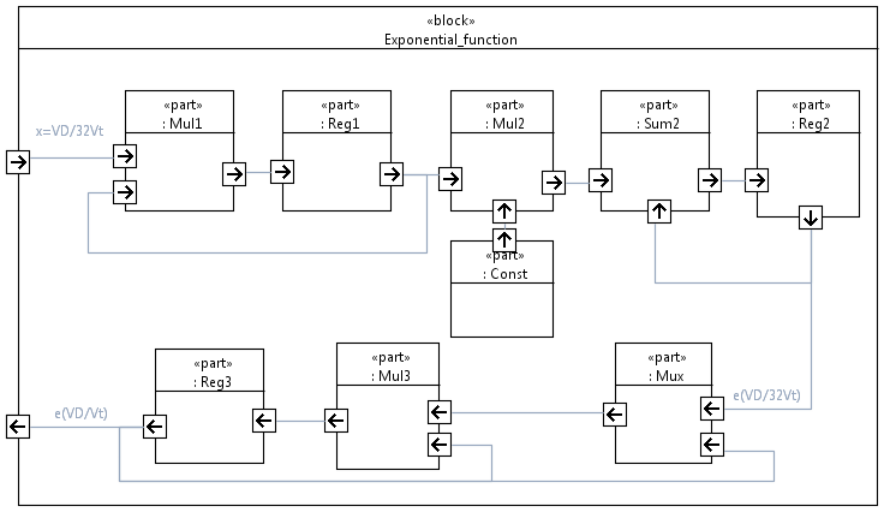
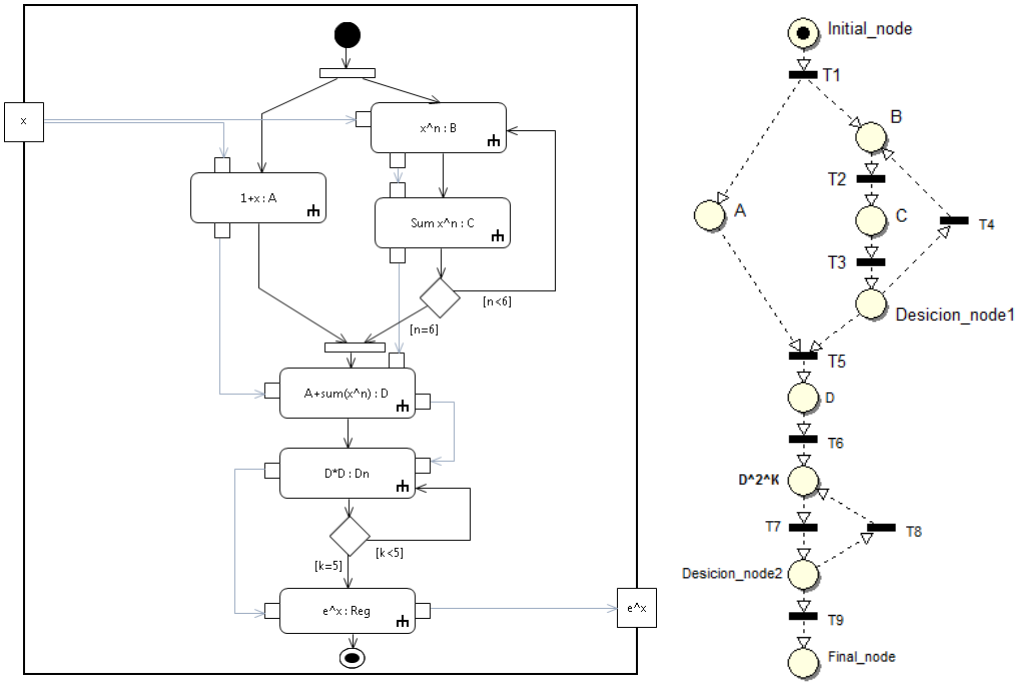


Figure 4: Partial *Internal Block Diagram* for computing exponential function for PV model.

Results

In order to validate the proposed methodology, commercial PV panels are taken as reference [11][12][13]. The parameters of these PV panels are the input of the model. Fig.6 depicts the model results from parameters of PV Module KC200GT [11]. This results use the proposed methodology and the Lambert function (analytical solution from eq.(4)). The I-V characteristic results confirm the accuracy of the proposed method in comparison with PV



(a) *Activity Diagram* (b) Control Petri net

Figure 5: Behavioral representation for computing exponential function of PV model

panel technical data [11] and the analytical model. Additionally, table I shows the hardware requirements and performance of the simplified method to evaluate the exponential function in a PV model. Results in this table describe the important decreasing of hardware requirements and performance time in contrast with conventional approach. This reduction of hardware requirements allows implementing the proposed methodology on FPGA for Hardware-in-the-Loop applications in real-time emulations. Furthermore, table II shows that appropriate initial value is fundamental for the quadratic convergence of Newton-Raphson method. In addition, experimental and simulation results in Fig.7 and Fig.8 show the suitable model accuracy.

Table I: Hardware requirements for implementation of PV exponential function

	Expected error	Polynomial terms	Execution of operations	Expected time performance
Standard	0.001	120	240	19.2μS
Proposed method	0.001	6	12	0.96μS

Table II: Iterations (n) of Newton-Raphson method for PV panel model

Initial value	n (S_R 1000W/m^2)	n (S_R 800W/m^2)	n (S_R 600W/m^2)	n (S_R 400W/m^2)
$V_{D_0} = 0.45$	206	139	72	8
$V_{D_0} = 0.5$	40	27	14	3
$V_{D_0} = 0.6$	3	4	5	7
Proposed initial point	2	3	3	5

Conclusion

This paper showed an alternative method to implement PV panel models. The Newton-Raphson numerical approach, the Taylor polynomial method and the Lagrange remainder were integrated through this research to simplify the required calculus to evaluate PV panel models. Furthermore, SysML diagrams and Petri nets were appropriately used to describe the system structure for implementing the proposed method on FPGA. This simplified method shows low expected error, adequate time execution, and limited hardware utilization in comparison with traditional computing methods. These characteristics are suitable for Hardware-in-the-Loop applications in real-time for standard conditions. However, The proposed methodology disregards other external variable conditions. As a consequence, this study can be extended to other external variables, such as shaded conditions, in order to increase the model accuracy.

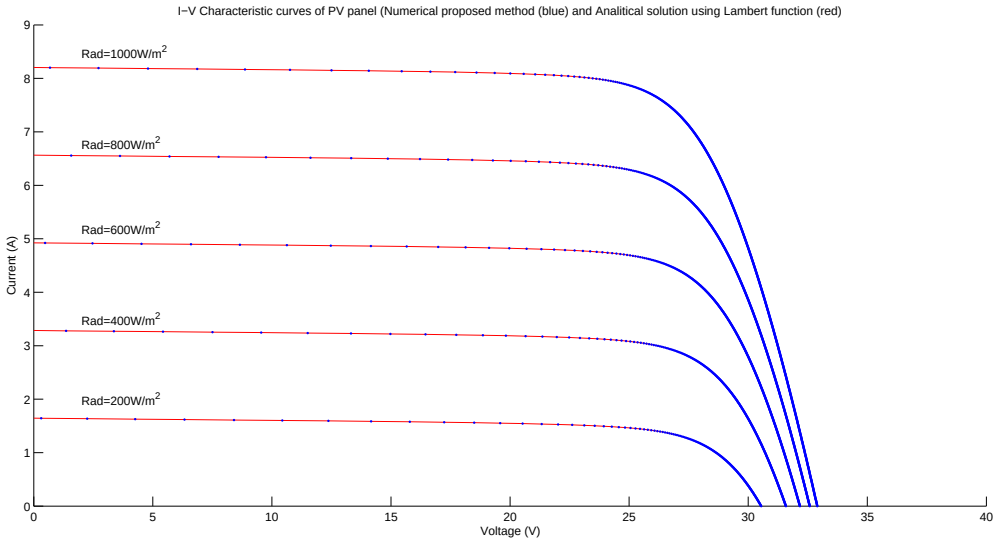


Figure 6: $I - V$ characteristic of panel KC200GT [11]. Proposed method (*blue) and solution with Lambert function (-red).

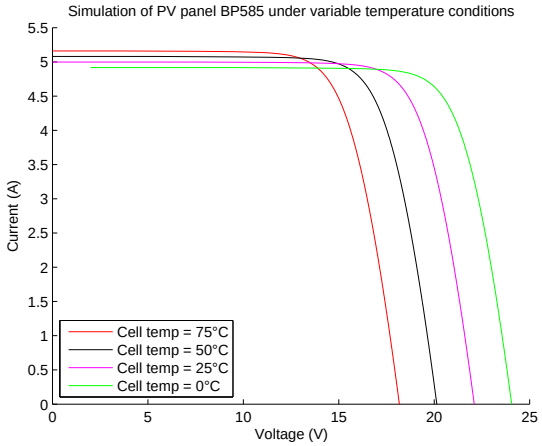


Figure 7: Simulation of PV panel BP585

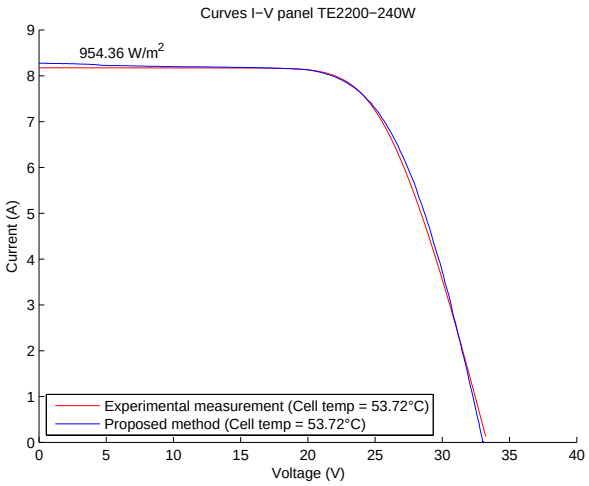


Figure 8: Experimental measurements from PV panel TE-2200 and I-V curve from proposed method

References

[1] R. Stala, "Testing of the grid-connected photovoltaic systems using fpga-based real-time model," in *Power Electronics and Motion Control Conference, 2008. EPE-PEMC 2008. 13th*, pp. 1852–1858, Sept 2008.

[2] J.-H. Jung, M.-H. Ryu, J.-H. Kim, and J.-W. Baek, "Power hardware-in-the-loop simulation of single crystalline photovoltaic panel using real-time simulation techniques," in *Power Electronics and Motion Control Conference (IPEMC), 2012 7th International*, vol. 2, pp. 1418–1422, June 2012.

[3] P. Rajesh, S. Rajasekar, R. Gupta, and P. Samuel, "Solar array system simulation using fpga with hardware co-simulation," in *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pp. 2291–2296, June 2014.

[4] R. Stala, "Testing of the grid-connected photovoltaic systems using fpga-based real-time model," in *Power Electronics and Motion Control Conference, 2008. EPE-PEMC 2008. 13th*, pp. 1852–1858, Sept 2008.

[5] D. Ickilli, H. Can, and K. Parlak, "Development of a fpga-based photovoltaic panel emulator based on a dc/dc converter," in *Photovoltaic Specialists Conference (PVSC), 2012 38th IEEE*, pp. 001417–001421, June 2012.

[6] C. Gomez, J.-C. Pascal, J. Jimenez, and P. Esteban, "Heterogeneous systems verification on hiles designer tool," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 132–137, Nov 2010.

[7] E. Koutroulis, K. Kalaitzakis, and V. Tzitzilonis, "Development of an fpga-based system for real-time simulation of photovoltaic modules," in *Rapid System Prototyping, 2006. Seventeenth IEEE International Workshop on*, pp. 200–208, June 2006.

[8] J.-H. Jung and S. Ahmed, "Model construction of single crystalline photovoltaic panels for real-time simulation," in *Energy Conversion Congress and Exposition (ECCE), 2010 IEEE*, pp. 342–349, Sept 2010.

[9] H. Can, D. Ickilli, and K. Parlak, "A new numerical solution approach for the real-time modeling of photovoltaic panels," in *Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific*, pp. 1–4, March 2012.

[10] A. Gutierrez, H. Chamorro, and J. Jimenez, "Hardware-in-the-loop based sysml for model and control design of interleaved boost converters," in *Control and Modeling for Power Electronics (COMPEL), 2014 IEEE 15th Workshop on*, pp. 1–6, June 2014.

[11] Kyocera, *KC200GT: Datasheet. http://www.kyocerasolar.com/assets/001/5195.pdf.*

[12] BP, *BP585: Datasheet.http://www.comel.gr/pdf/bpsolar/BP585.pdf*.

[13] Tenesol, *TE2200: Datasheet.http://www.mltdrives.com/datasheets/solar/tenesol/TE220-240-60PP.pdf*.