Model Checking Cyber-Physical Energy Systems

Youssef Driouich
Dip.to di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata (DIEM)
Università degli Studi di Salerno, Italy
ydriouich@unisa.it

Mimmo Parente
Dip.to Scienze Statististiche & Innovation of
Systems (DISA-MIS)
Università degli Studi di Salerno, Italy
parente@unisa.it

Enrico Tronci
Dip.to di Informatica
Università degli Studi di Roma,
"La Sapienza", Italy
tronci@di.uniroma1.it

Abstract— **Cyber-Physical Systems (CPS) integrates computing and physical processing: their behaviour is dictated by both computation and physical components. In this paper, we present a Model Checking approach for the automatic verification of the correctness of the system at hand. The strengths of our approach are the followings. First, we have implemented the CPS using the modelling and simulation based software JModelica to exploit its modular and parametrizables components. Second, we have verified the system's properties with a model-checking technique to exploit the automaticity of this approach and the completeness of the exploration of all the states of the CPS. In fact, with very few basic notions of the field to investigate, it is possible to design a complete model of a CPS. The system we model, for our proof of concept, is a Distributed Maximum Power Point Tracking (DMPPT) for photovoltaic systems, which relies on a control software to manage the system in every condition imposed by the environment (e.g. irradiation, temperature). We have parameterized the main physical inputs to carry out an automatic verification of a thorough simulation of the circuit under a large number of operational conditions, to deeply investigate the critical aspects of the system. The final goal of our study has been to design a tool to enable automatic verification using a model-checking approach in order to ensure the system correctness with respect to given requirements provided by the designer.**

Keywords-component; Cyber-Physical Systems; Model Checking; Simulation; Automatic Formal Verification; System Analysis and Design; Distributed Maximum Power Point Tracking; Photovoltaic circuit; Automatic Verification

I. INTRODUCTION

The term Cyber-Physical System nowadays indicates a form of integration, or better a melting, of physical and computational processes [1]. The CPSs knew a very fast evolution in different fields like automobiles, smart houses, robots and many others.

Today's CPS are heterogeneous entities that span the cyber and physical worlds, hardware and software, sensors and actuators, etc. They must operate in highly dynamic environments and for dynamically changing objectives, and therefore, must be adaptive [2]. The energy field knew a significant progress especially on the smart grids or energy installation for smart houses but the photovoltaic technology did not get a lot of interest from the CPSs designers.

In this paper, we use an adaptive hybrid CPS modelling approach that combines discrete and continuous dynamics of a Distributed Maximum Power Point Tracking (DMPPT) for photovoltaic systems, which emphasizes the variabilities of the physical side and control aspects of the system together with the automatic verification of the system requirements. This allows the application of parameterizable physical models and automatic control algorithm parameters for the proposed system.

We have implemented the DMPPT and the system with parameterizable models by using a modelling and simulation-based software, JModelica [8] that allows to modularize and to parameterize the components. In such a way our (concurrent) control algorithm identifies the state the Photovoltaic System (PS) is in, during its working period, under different natural conditions (e.g. sun irradiance and ambient temperature). The modelling of the proposed system is constituted by a set of panels; a Perturb & Observe (P&O) based controllers, to adjust and to control the flow of the power produced by the panel, and by a discrete controller to detect the state the converter is into, in real time.

JModelica offers, the interoperability and reusability of the implemented components. In fact, the modeling language, Modelica, is platform independent and thus can be run for simulation on many different tools both open source, like OpenModelica and JModelica, and commercial like Dymola. JModelica consists of modules, implemented by files with extension *.mo.* Each module models a part of the dynamics of the system. The modules run concurrently, thus, their computation steps interleave with each other. This generally causes some problems, and much caution has to be taken during the design. The components and subsystem models we have designed enrich the Modelica library of photovoltaic components using DMPPT architecture and their associated control algorithms. Our system can be used within tools like Dymola or OpenModelica to create models of PV systems in DMPPT. Moreover, these same tools can be used in conjunction with others supporting the *Functional Mock-up Interface* standard for model exchange and co-simulation. For

978-1-5386-2847-8/17/$31.00 ©2017 European Union

example, a PV system model developed in OpenModelica using these models could then be used to validate a control algorithm developed in MATLAB/Simulink or LabVIEW.

Besides the modeling approach, our goal has been the verification of the correctness and the reliability of the system automatically, without human supervision. For instance the works described in [5, 6, 7] to formalise system requirements and like those in [10, 12, 13] to define admissible operating scenarios. We presents a model checking [18] approach based on trace-based technique that allows to drive the simulations and to save the system's state in case of an abnormal behaviour or an error.

The paper is organized as follows: The section II presents the CPS and underlines the major steps to realise the modeling of our system. Section III describes the environment used to achieve the simulations together with the verification flow of the system's properties. The last section gives a summary of the realised work and the forthcoming enhancement that can be accomplished.

II. Modeling of the CPS

The models we study are primarily about dynamics, the evolution of the DMPPT system state in time. Our purpose is to verify if our system is able to minimize the loss of produced power when the irradiation of the panels is changing frequently in one hand, in the other hand, to check if the system converges to a desired behaviour under the actions of the controller.

A. Physical modeling

Physical modeling is a way of modeling and simulating systems that consist of real physical components. Physical processes are compositions of many parallel processes [11]. Measuring and controlling the dynamics of these processes by managing actions that influence the processes are the main goal of the CPS. In our system, solar cells, photovoltaic panels, dc/dc converters and dc/ac inverters *software* models are the main components of the physical system (fig. 1). To model the continuous dynamics of those elements, we have adopted an equation-based approach. For example, the PV cell model is described by the non-linear equation given in (1), where Ipv and Vpv are current and voltage of the panel, Rs and Rh are the series and parallel resistances of the model, *Vd = Vpv + Ipv·Rs* is the voltage of the diode model and Io its inverse saturation current. n is the ideal factor of the p-n junction, k is the Boltzmann constant, q is electron charge, and Tpv is the panel temperature in kelvins. Finally, Iph is the current produced by the photovoltaic effect.

$$I = Iph - Io \cdot q \cdot exp((Vpv + Ipv \cdot Rs)/\eta \cdot k \cdot Tpv) - 1 - (Vd / Rh) \qquad (1)$$

The architecture based on a DC-DC converter for each PV module also allows Maximum Power Point (MPP) tracking in each panel, removing the local power maximums and leaving a single one.

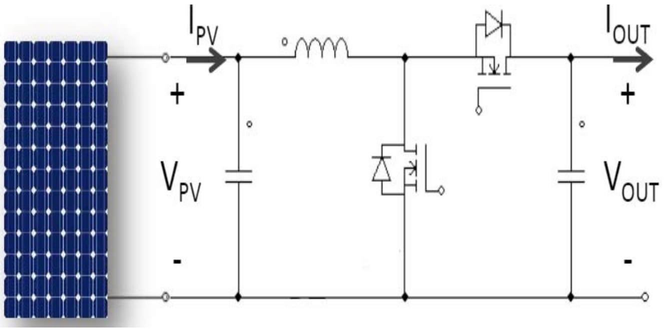

Figure 1: Physical model of the circuit

B. Control algorithm

Two controls are adopted: the first control is the Maximum Power Point Tracking (MPPT) based on the P&O method. This operates by periodically incrementing or decrementing the output terminal voltage of the photovoltaic cell and comparing the power obtained in the current cycle with the power in the previous cycle (in [*3*] the MPPT control is discussed in details). However, the main control of our design consists in adapting the switching mode (duty cycle) of the dc/dc converter according to the current of the connected panel (*I_pan*) and the output current of the converter (*I_out*), in order to allow to the system the tracking of the MPPT. We have modelled each plant in such a way that the software modules of the converter communicates the *I_out* value to the other modules. In this way, each software module of the converter can detect the state in which the others are into and adapt its state accordingly. More precisely, this control depends on the computation of *I_out* and of *I_pan* and depending on the entered values, the switching mode is determined. The output voltage of the converter (*V_out)* is computed which modifies the switching frequency of the converter by calculating a new value of the MPPT duty cycle.

Control	Mode

V_pan = Vmpp
I_pan = I_mpp
	MPPT

V_pan > V_mpp I_pan < I_mpp	CUT-OFF
V_pan < V_mpp I_pan > I_mpp	PASS-THROUGH
V_pan <= 0 I_pan = I_SC	BY-PASS

C. Cyber System

According to [3] there are four possible modes of the DMPPT system. First the MPPT mode, which is the desired working way of the dc/dc converter. In this case, the output voltage is in the range between (*V_max*) and (*V_min*). This protection measure is realized by modeling a blocking diode. The converter in this mode switches according to the computed value of the duty cycle, the output voltage(*V_out*) is computed as follows:

$$V\_out = (Vpv * Ipv) / I\_out \qquad (2)$$

The second mode is the CUT-OFF: this typically occurs when the converters are connected to unshaded panels and generates much more power than shaded ones; the converter still switches and the output voltage is computed as follows:

$$V\_out = V\_max \qquad (3)$$

The third state is the PASS-THROUGH: to boost converters that are connected to shaded panels that generates much less power than un-shaded ones; the converter in this mode stops switching and the panel is directly connected in series to the output. The output voltage is computed as follows:

$$V\_out = Vpv - Rs * I\_out \qquad (4)$$

The fourth mode is the BY-PASS: when the panel is heavily shaded and it cannot be connected to the string because it would sink rather than source power. In this mode, the converter bypasses the panel and here too stops switching. The output voltage equation describing this mode is:

$$V\ \_out = - Rs * I\_out \qquad (5)$$

In particular, once the control algorithm implemented in detects the mode, according to the values of the currents, the formula for computing *V_out* is communicated to the mode controller, which modifies the value of the duty cycle, this way determining the switching mode. If the control does not detect a change in the values of the currents, then the output voltage is unmodified.

III. Simulation and Verification Flow:

We have used the JModelica software for the simulation of the CPS, which is an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems. The fact that Modelica language is an object-oriented and equation based language makes the modelling of the software components effortless; the models are available in [15]. The proposed verification technique allows the exploration of various states of the CPS under different values of irradiation. To perturb the system in order to verify the correctness of the properties, we are using the following technique. Let n be the number of panels, k the number of *perturbations* of the irradiation during the simulation process, Z the set of the possible values assigned to the irradiation, and p the cardinality of Z. The number of traces clearly is $(p^n)^k$. However, by ignoring w.l.o.g. symmetric values and assuming that the irradiation of the *i-th* panel is always less than or equal to the irradiation of the *(i+1)-th* panel for i in [$1, n-1$], we get fewer traces to check: $(C(p-1, p+n-1))^k$.

A. Simulation:

In our experiment, we modelled two photovoltaic panels *(n=2)*, twelve cells per panel, with different levels of irradiation, with 0.01 seconds as sampling time of the MPPT control and 0.05 seconds as sampling time for the converter state control.

Table 1 shows the parameters used to execute the experimental simulations.

Using the time events functionality of Modelica language, we created an event that changes the value of the irradiation four times *(k = 4)* during one simulation. The irradiation of a single panel can take one of the following values: 0%, 30%, 70%, and 100% *(p =4)*. Using the calculation described before, the traces to check are hence $C(3,5)^4$ = 10,000.

The compiled code of the simulation is executed together with a model-checking module in order to explore the 10,000 states of the system. This module detects whether the system is working abnormally. If a bug or an error arises, the system's state is recorded into a log file.

Parameters	Values
Nominal Irradiation of the panels	1000 W/m2
Rs (series resistance)	0.11 Ω
Rp (parallel resistance)	148 Ω
T_amb	20 °C
I_ph_STC	7.7 A
T_STC	25 °C
T_NOCT	46 °C
Fs (Switching frequency if the converter)	5^4
Ta(Sampling time of the MPPT control)	0.01 s
Td(Sampling time of the mode control)	0.05 s
V_max	(0.8 * cell's number) V
V_min	(0.05 * cell's number) V

TABLE1: PARAMETERS USED TO EXECUTE THE SIMULATIONS.

B. Verification flow: Model checking

We created the executable model of our circuit in Modelica instantiating the photovoltaic panel equations, the dynamic irradiation of each cell of the panels, the feedback control of the MPPT and the discrete control of the converter states. Once the model has been compiled, we obtain the simulation executable code. The input to this code is the number of panels, the number of cells for each panel and the time-map of the irradiation for all the photovoltaic cells. The outputs is the total power produced by the overall system (*P_out*) and the efficiency factor, which is the ratio between the input power (panel power) and *P_out*.

To make the verification process automatic, we adopt a trace-based verification approach that consists on conducting the simulations via a trace file, that allows us the exploration of all the states in which the systems operates on (in our case all the possible combinations of the irradiation range 0%-100%, with a step of 30%). This verification avoids the complexity of traditional verification techniques, by analysing the traces and by working directly with the system under verification. Each trace line starts with the index number ranging from 0000 to 9999 then the event (irradiation change) followed by the simulation time point (every 0.25 sec) in which this event occurs.

Every digit of the index number represents a perturbation of the irradiation, since we have four digits it means that the simulation time is divided on four time quantum's, one quantum for each perturbation. For example, if the index is 2375, for the first time quantum (<0.25 second) the irradiation is 0% for the first panel and 70% for the second panel coded with the digit value 2. In the second time quantum (>=0.25 and < 0.5) the irradiation is 0% for the first panel and 100% for the second panel, coded with the digit 3. In the third time quantum (>=0.5 and < 0.75) the irradiation's value is 70% for both panels. In the last time quantum (>= 0.75) the irradiation takes the values 30% and 70% for the first and second panels respectively. Table 2 show the indexes coding the irradiation values.

Index	Irradiation first panel	Irradiation second panel
0	0%	0%
1	0%	30%
2	0%	70%
3	0%	100%
4	30%	30%
5	30%	70%
6	30%	100%
7	70%	70%
8	70%	100%
9	100%	100%

TABLE 2: ASSOCIATIVE TABLE BETWEEN THE INDEX AND THE IRRADIATION

This trace file is given as input to the Python script that launches many simulations in parallel, our hardware allows to run 15 simulations simultaneously. Within this script, three functions are implemented; the first function is responsible of the configuration of the simulations where the simulation time, the time-map irradiation and the number of panels are transmitted. The second function reads the trace file and compiles the models before launching the simulations. The last function creates a log file in case of a malfunction of the mode controller and saves the systems state for which the abnormal behaviour occurred, and verifies the *quality* of the outputs at the end of each simulation; this is provided as a *counter-example* of the specifications. More precisely, it checks whether the system satisfies the intended specification which, in our case, is whether the efficiency factor is close to 1. Once the simulation is completed, the last index is saved and the log file is written. In the log file, the irradiations of the panels, the modes of the converters and the time point in which the anomaly occurred are backed up. Using this automatic verification flow, we are checking the behaviour of the models and the controls in one hand, and the specifications of the system design in the other hand.

C. Hardware configuration:

The pre-compiled simulation permits us to finish the simulation within 5600 seconds on a 32-core Intel Xeon@2.7GHz machine with 64-Gigabyte available memory, for 1 seconds of the simulation time. The results size of a single simulation is 1.5 Gigabyte, since we need to simulate 10,000 scenarios, we used a file transfer function that upload the results files in a cloud storage when the hard disk of the machine is 90% full, by automatizing this process we are not interrupting the ongoing simulations.

D. Results:

We briefly present a case of the results from the simulations. It consists of plotting the power of both panels, the modes of the converters, the time-map irradiation for each panel, the efficiency factor, and finally the output power. The index is equal to {2578} in this case.

The power of the first panel (fig.6) is negative in the first time quantum since the panel is completely shaded (fig. 2), it will sink energy rather than produce it. The mode's value of the connected converter is BY-PASS (fig. 4). After the second time quantum, the panel is irradiated at 30% and the mode of the connected converter switches to MPPT, which is not withstanding the fact that the panel is irradiated at the 30%. The time point (0.25 second), the mode, the value of the irradiation and the efficiency factor are then saved into the log file and provided as the counterexample to the specification.

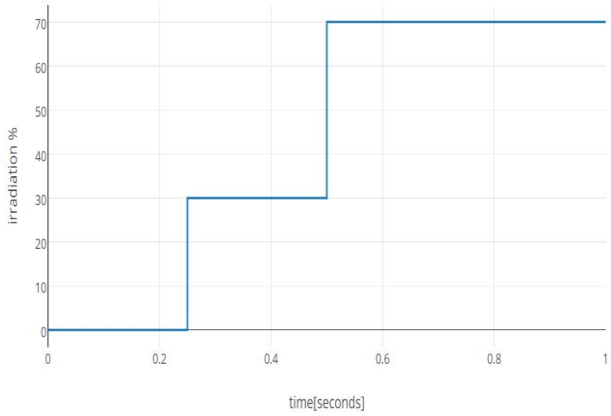

FIGURE 2: THE IRRADIATION OF THE FIRST PANEL

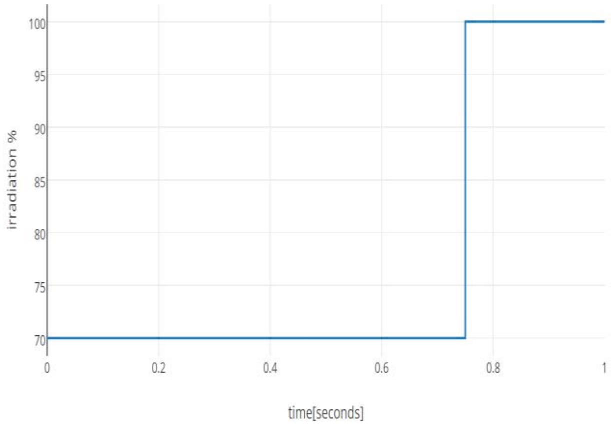

FIGURE 3: THE IRRADIATION OF THE SECOND PANEL

For the third and fourth time quantum, the panel is irradiated 70% and the mode is either MPPT or CUT-OFF, which is the correct and expected behaviour of the system.

The irradiation value of the second panel (fig. 3) is 70% from the beginning of the simulation to the third time quantum, the converter mode is CUT-OFF. After the irradiation increases to 100% and the converter mode switches between CUT-OFF and MPPT. Again, the state of the system is archived on the log file because it does not correspond to the specifications.

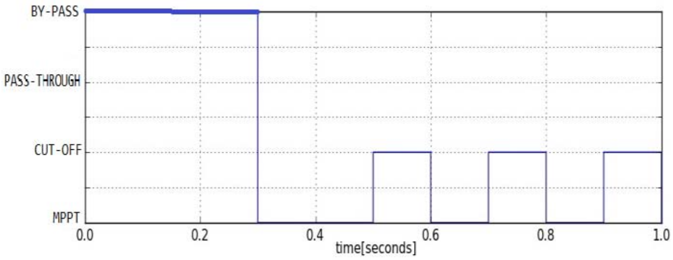

FIGURE 4: THE MODES OF THE CONVERTER CONNECTED TO THE FIRST PANEL

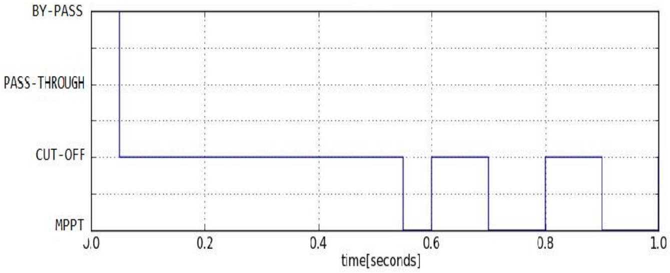

FIGURE 5: THE MODES OF THE CONVERTER CONNECTED TO THE SECOND PANEL

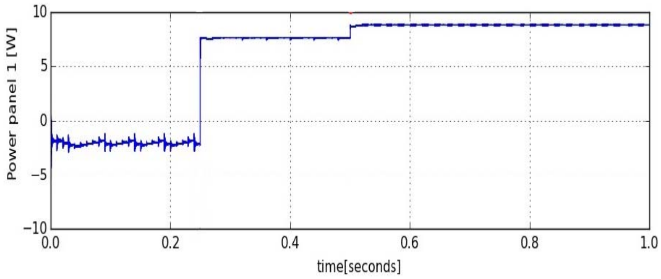

FIGURE 6: POWER OF THE FIRST PANEL

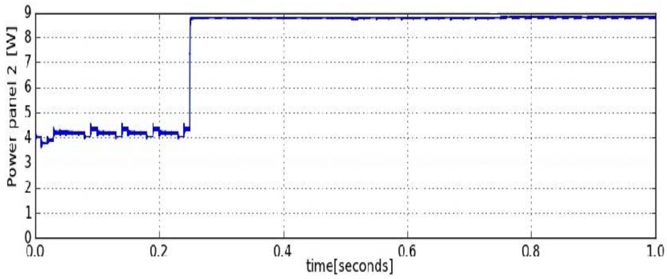

FIGURE 7: POWER OF THE SECOND PANEL

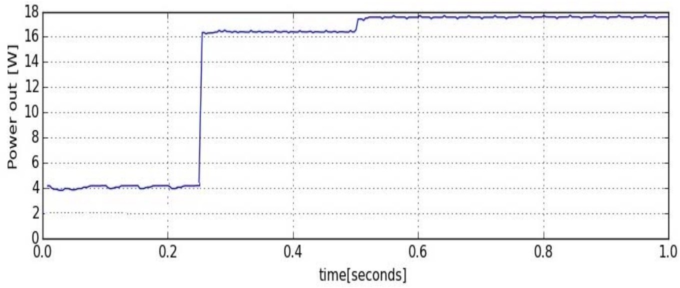

FIGURE 8: POWER OUTSIDE THE SYSTEM

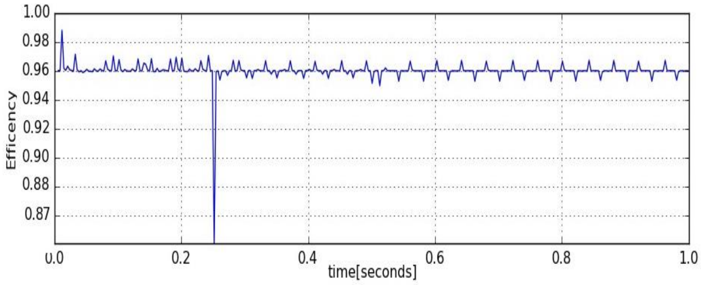

FIGURE 9: THE EFFICIENCY FACTOR DURING THE SIMULATION PROCESS

The power out (fig. 8) satisfies the properties of the system; in the first time quantum, the value of the power is very close to the power of the second panel, which proves that the first panel was by-passed. For the rest of the simulation time, the power outside is approximately the sum of both powers of the panels, demonstrating that the power lost is meaningless. The efficiency factor (fig. 9) is very close to 1 in the major part of the simulation process which proves the good performance of the system. The drop on 0.25 second has been already archived on the log file.

IV. CONCLUSION AND FUTURE WORKS

Using as case study, the DMPPT from [4], we have shown how Modelica can be effectively used to model CPS stemming from PESs along with the model checking techniques to ensure the correctness of the system requirements. This is a fundamental step to enable usage of Modelica for model-based verification of complex PES. Our simulation results show that CPSs stemming from PESs can be easily modelled, validated using Modelica, and finally verified using the proposed verification approach. For the future, we are planning to reduce the perturbation step of the irradiation to 1% and enlarge the time quantum, which will require a huge amount of resources (memory, storage space, processors) and a significant time. Besides shortening the time quantum and the irradiation step, we are planning to increase also the number of panels and scale it for a modular version. We are projecting also to include the Temporal Logic in the expression of our system's properties, as instance FORM-L[17], which is a formal requirement modeling language, enabling the visual modeling of system properties as well as their verification through simulation.

REFERENCES

[1] E. A. Lee and S. A. Seshia, "Introduction to Embedded Systems: A Cyber-Physical Systems Approach", Second Edition, MIT Press, 2017.

[2] S. A. Seshia, S. Hu, W. Li and Q. Zhu, "Design Automation of Cyber-Physical Systems: Challenges, Advances, and Opportunities". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, n. 99, 2016 DOI: 10.1109/TCAD.2016.2633961

[3] M. De Cristofaro; N. Femia; M. Migliaro; G. Petrone "Minimum Computing Adaptive MPPT Control" ISIE 2014, Pages: 1384 - 1389, DOI: 10.1109/ISIE.2014.6864816.

[4] M. De Cristofaro; G. Di Capua; N. Femia; G. Petrone; G. Spagnuolo; D. Toledo "Models and Methods for Energy Productivity Analysis of PV Systems" INDIN'15, Pages: 1153 – 1158, DOI: 10.1109/INDIN.2015.7281898

[5] A. Murano, M. Napoli, M. Parente. "Program Complexity in Hierarchical Module Checking", LPAR'08, ISBN: 978-3-540-89438-4.

[6] A. Ferrante, M. Napoli, M. Parente, "Model-Checking for Graded CTL", Fundamenta Informaticae, 96(3), 323-339, 2009.

[7] A. Ferrante, M. Napoli, M. Parente, "Graded-CTL: Satisfiability and Symbolic Model Checking", Int.l Conf. on Formal Engineering Methods, (ICFEM) 2009: 306-325, Lecture Notes in Computer Science 5885, Springer 2009, ISBN 978-3-642-10372-8.

[8] http://www.jmodelica.org/

[9] N. Femia, G. Lisi, G. Petrone, G. Spagnuolo, and M. Vitelli "Distributed Maximum Power Point Tracking of Photovoltaic Arrays: Novel Approach and System Analysis" IEEE Transactions on Industrial Electronics (Volume: 55, Issue: 7, July 2008), DOI: 10.1109/TIE.2008.924035

[10] T. Mancini; F. Mari; A. Massini; I. Melatti; E. Tronci "SyLVaaS: System Level Formal Verification as a Service", PDP 2015 Pages: 476 - 483, DOI: 10.1109/PDP.2015.119

[11] H. M. Buini, S. Peter, T. Givargis "Including Variability of Physical Models into the Design Automation of Cyber-Physical Systems" DAC 2015, Pages: 1 - 6, DOI: 10.1145/2744769.2744857

[12] T. Mancini, F. Mari, A. Massini, I. Melatti, F. Merli, and E. Tronci "System level Formal Verification via Model Checking Driven Simulation", CAV 2013, DOI: 10.1007/978-3-642-39799-8_21

[13] T. Mancini, F. Mari, A. Massini, I. Melatti, E. Tronci, "System Level Formal Verification via Distributed Multi-Core Hardware in the Loop Simulation", PDP 2014, DOI: 10.1109/PDP.2014.32

[14] J. C. Jensen; D. H. Chang; E. A. Lee, "A Model-Based Design Methodology for Cyber-Physical Systems" (IWCMC), 2011, DOI: 10.1109/IWCMC.2011.5982785

[15] https://github.com/driysf/DMPPT-CPS

[16] http://www.modelica.org

[17] A.Garro, A. Tundis, D. Bouskela, A. Jardin, N. Thuy, M. Otter, L. Buffoni, P. Fritzson, M. Sjölund, W. Schamai, H. Olsson, "On formal cyber physical system properties modeling: A new temporal logic language and a Modelica-based solution "(ISSE), 2016, DOI: 10.1109/SysEng.2016.7753137

[18] E.M. Clarke, Jr. , O. Grumberg, D. A. Peled, "Model Checking", ISBN:978-0262032704