Modeling Cyber-Physical Systems for Automatic Verification

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 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*— In this paper we show how the open standard modeling language Modelica can be effectively used to support model-based design and verification of cyber-physical systems stemming from complex power electronics systems. To this end we present a Modelica model for a *Distributed Maximum Power Point Tracking* system along with model validation results.**

Keywords— Cyber-Physical Systems; Modeling; Simulation; Automatic Formal Verification; System Analysis and Design; DMPPT; Photovoltaic systems;

I. INTRODUCTION

A Cyber-Physical System (CPS) consists of physical (e.g., electrical, mechanical, etc) as well as computational (e.g. Microcontrollers with their software) components [1]. Design, verification and validation of CPSs is particularly challenging because of the complex interactions stemming from the simultaneous presence of analog (physical) and digital (computational) components. In fact, the former components yield continuous variables whereas the latter yield discrete ones. To save on design time and cost, CPS design and verification is first carried out on simulation models and only when the design has been thoroughly verified through simulation the actual physical systems are built. Many simulators (e.g., Simulink, Ngspice) are available to support such an activity. However, most of such simulators come with their own (possibly proprietary) modeling language. This hampers the possibility of exchanging or sharing models as well as of reusing them if the simulation environment changes.

This problem has been addressed by defining an open standard modelling language, namely Modelica [16] for which many commercial (e.g., Dymola) as well as open source (e.g., Jmodelica, OpenModelica) simulators are available. This enables reuse of simulation models across Modelica simulators as well as across operating systems. Furthermore Modelica models are a-causal, thus supporting an easy object-oriented modelling approach substantially reducing the modelling effort and enabling reuse of models, hardly possible with causal models (e.g., as those from the signal-flow modelling approach used, for example, within MATLAB/Simulink). Finally, much as, e.g., Simulink, Modelica semantics is synchronous (that is, software execution takes zero time) this enables an easy synchronization among the concurrently running system components.

While the use of Modelica for modelling and verification has been extensively investigated in the automotive and aerospace industries, it is much less so for Power Electronics System (PESs). However, the ever growing complexity of PESs, along with the ever decreasing time and budget available for their design, motivate investigating the possibility of using Modelica also for modelling of CPS stemming from PESs.

Model based design basically entails three steps: modelling the CPS being designed, validating the model (so that simulation results can be trusted) and finally verifying that the design satisfies given requirements. In this paper, we focus on the first two steps (modelling and validation) by showing with a relevant case study how Modelica can be effectively used to model PESs. To this end, *w*e present a Modelica model for the Distributed Maximum Power Point Tracking (DMPPT) system built out of the Perturb&Observe (P&O) based Maximum Power Point Tracking (MPPT) circuit described in [9]. We simulate our Modelica model for DMPPT using the JModelica [8] simulator. In future work we plan to address Modelica based verification of complex PESs. For example, by using approaches like those in [5, 6, 7] to formalise system requirements and like those in [10, 12, 13] to define admissible operating scenarios. Design exploration can also be addressed, for example, along the lines of [11].

978-1-5090-5052-9/17/$31.00 ©2017 IEEE

This paper is organised as follows. Section II presents our design methodology and outlines the three main steps to model our system. Section III describes the tools and environment used to carry out the simulation. The last section gives a summary of the related work and future research directions.

II. Design of the DMPPT-CPS

In this section, we explain our design methodology, which adopts the Model-Based Design [14]. Our flow consists on parameterizable models for the Physical System (PS), the Control Algorithm (CA), and the Control Software (CS), they will be instantiated during the simulation process, which is described later on.

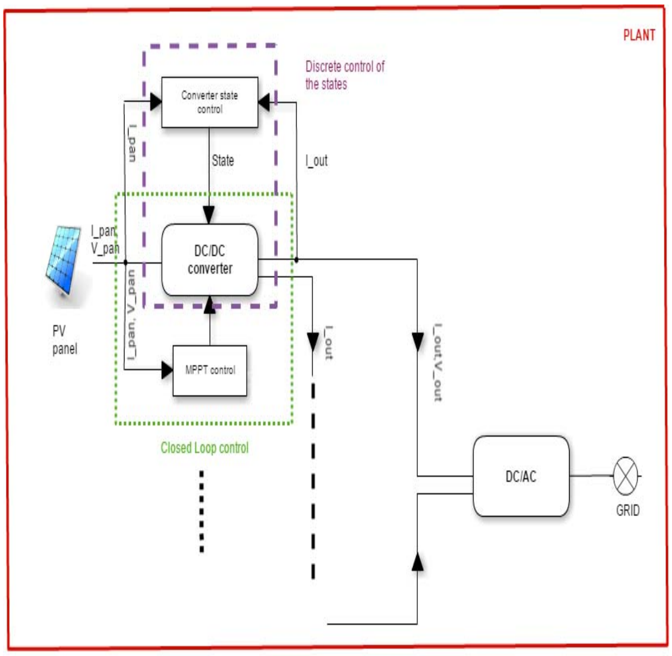

Figure 1: Architecture of the system

We use the work described in [4, 9] as the study case to model.

A. Physical modeling:

Physical modeling deals with the expression of the physical system consisting of plant and actuators in mathematical or logical terms [11]. Continuous dynamics modeling is required since we want to know how the PS regulates the production of energy in function of the irradiation variability. In the modeling of the DMPPT-CPS, the model of the overall circuit is composed by a set of physical components: the solar cells, the photovoltaic panel, a dc/dc converter and a dc/ac inverter (see Fig. 1). We use an equation-based modeling approach to express the physical dynamics of the circuit.

As usual, the output current (I) of each solar cell is computed by solving the Kirchhoff's law:

$$I = Iph - Ir \cdot q \cdot exp((V + I \cdot Rs)/\eta \cdot k \cdot T) - 1 - V + I \cdot Rs\ Rp \quad (1)$$

where V is the output voltage of the cell, Iph is the photocurrent, Ir is the saturation current, Rs is the resistance in series, Rp is the resistance in parallel, q is the electrical charge ($1.6 \times 10{-}19$ C), η is the p-n junction quality factor, k is the Boltzmann constant ($1.38 \times 10{-}23$ J/K), and T is the temperature (in Kelvin degrees).

Fig. 1 describes the overall architecture. The physical dynamics is implemented with the following Modelica modules [14]: the file *PV_cell.mo* implements (1), the equations of the solar cell's temperature, photocurrent and the diode saturation current. *PV_panel.mo* models the photovoltaic panel constituted by the cells distributed in series and finally the dynamics of the irradiation, which is the variation during the simulation of the irradiance, in the file *config_irradiance.mo*.

B. Control Algorithm:

In our approach to model the DMPPT-CPS, 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*). 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. This is crucial for our verification purposes. 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. A bird's eye view of the part of our algorithm where we describe this control is the following:

if (*state_controller (I_out, I_pan)* != *actual state*) **then**
 new_value of *duty cyle*;
else
 keep the current value of *duty_cycle*;
endif;

In our simulation software, the function *state_controller(I_out, I_pan)* is implemented in the Modelica model *state_controller.mo* (the rectangle *Converter state control* in fig. 1). The *MPPT control* (see fig. 1) is implemented in *duty_controller.mo (*and these models are called respectively within *state_control.mo* (dashed lines) and *closed_loop.mo* (dotted lines), [15].

C. Cyber System:

As said in the previous section, the objective of the controller is to provide two main parameters that control the state of the DMPPT-CPS system, the switching mode and the computation of the output voltage.

According to [4] there are four possible states of the DMPPT system. First the MPPT state, 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 second state is the CUT-OFF: this typically occurs when the converters are connected to un-shaded panels and generates much more power than shaded ones; the converter still switches and the output voltage is computed as follows:

$$V\_out = V\_max \quad (2)$$

The third state is the PASS-THROUGH: to boost converters which 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:

$$V\_out = Vpv - Rs * I\_out \quad (3)$$

The fourth state 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:

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

In particular, once the control algorithm implemented in *state_controller.mo* detects the mode, according to the values of the currents, the formula for computing *V_out* is communicated to *state_control.mo*, 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 RESULTS

To validate our design approach, the system has to satisfy the intended specifications (requirements). In this case study, the system has to respond promptly to all variations of irradiation conditions, by harvesting the maximum power in each of these conditions. The requirements of the system can be formally expressed using Boolean formulas as follows:

- Conv_1(MPPT) ∧ (Conv_2(MPPT) ∧ I rr1[60%,100%]) ∨ (Conv_2(BY-PASS) ∧ Irr1[60%,100%])
- Conv_1 (CUT-OFF) ∧ Conv_2 (PASS-THROUGH) ∧ Irr_1[50%, 100%]
- Conv_1 (PASS-THROUGH) ∧ Conv_2 (CUT-OFF) ∧ Irr_1[25%, 40%]
- Conv_1 (BY-PASS) ∧ Conv_2 (MPPT, CUT-OFF, BY-PASS) ∧ Irr1[5%, 20%]

Where Conv_n (*STATE*) is the actual state of the n-th converter and Irr_n [*RANGE*] is the irradiation level of the n-th panel.

A. Simulation:

We have used the JModelica software for the simulation of the DMPPT-CPS, which is an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems.

We created the executable model of our circuit in Modelica (see fig. 2) 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 is compiled, we obtain the simulation executable code. The input to this code are the number of panels, the number of cells for each panel and the time-map of the irradiation for all the photovoltaic cells. The output is the total power produced by the overall system (*P_out*).

B. Setup and execution:

The execution is done by instantiating the simulation executable code. In our experiment, we modeled two photovoltaic panels, twelve cells for each panel with different level 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. The pre-compiled simulation permits us to finish the simulation within 2450 seconds on a 32-core Intel Xeon@2.7GHz machine for 0.5 seconds in simulation time.

Parameters	Values
Nominal Irradiation of the panels	1000 W/m^2
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

Table 1: Simulation's parameters.

Table 1 describes the parameters used to execute the simulations.

C. Results:

	Irradiation range 1st panel	Irradiation range 2nd panel	State 1st converter	State 2nd converter
a	[60%,100%]	[60%,100%]	MPPT	MPPT
b	[50%,100%]	[25%,40%]	CUT-OFF	PASS-THROUGH
c	[5%,20%]	[50%,100%]	BY-PASS	MPPT
d	[5%,20%]	[5%,20%]	BY-PASS	BY-PASS
e	[5%,20%]	[20%,50%]	BY-PASS	CUT-OFF

Table 2: Associative table of the irradiation level and the converter's states.

We have designed a script that automatically explores many possible variations of panel irradiation in function of the time. In each simulation, at fixed time steps we issue a variation step (increase or decrease) of 5% of the nominal irradiation of the panel. For example, starting with 500 W/m^2, the variation will be increase to 550 W/ m^2 or decrease to 450 W/ m^2. Table 2 describes the states the circuit enters into in function of the irradiation: the columns show the range of irradiation for the first and second panels and the state of the first and second converters, with the corresponding panel irradiation values as shown in Table. 2. We briefly present three cases of the table above (*a, b, c*). We assume that the ideal state of the system is when both panels are working in the MPPT mode (*a* on the Tab. 2), see Fig. 3 where the plot of the power of the first panel is shown. The maximum output power reached is around 17.65 W. For the first half of simulation time (when simTime <= 0.25s) the first panel is irradiated 100% and the second is irradiated

70% and for the second half (when simTime > 0.25s) the irradiation of the first panel is 60% and the second panel is 90%.

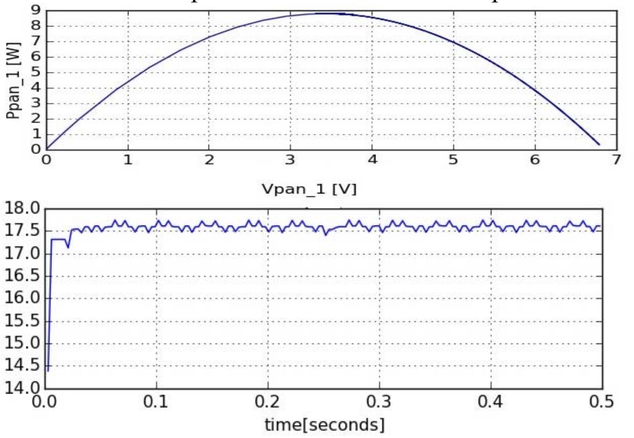

Figure 2: *Power of the panel in function of its voltage and the power of the system when both converters are in MPPT/MPPT.*

The fig. 3 describes the power outside the system when the converters work in MPPT/BYPASS states (*c* on Tab. 2), the second panel produces no energy, since it is by-passed. The first panel is irradiated 80 % and 70% and the second is irradiated 5% and 10%, respectively on the first and second half of the simulation time. The value of P_out on this case is around 8.8W because only one panel is harvesting energy.

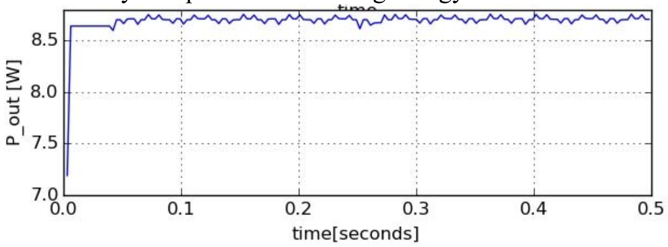

Figure 3: *Power outside the system in MPPT/BYPASS states.*

The third case (fig. 4) is when the panels work on CUT-OFF/PASS-THROUGH states (*b* on Table 2): the figure describes the power outside the system *P_out*. The maximum output power reached is around 17.35 W; the power lost is insignificant (0.3 W).

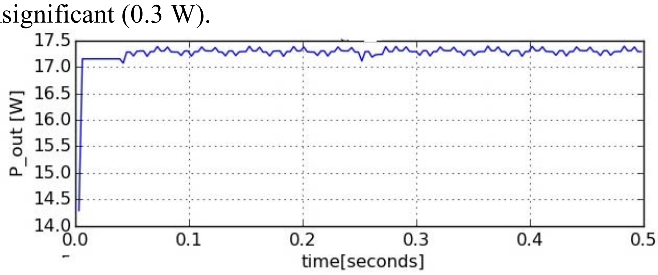

Figure 4*: Power outside the system in CUT-OFF/PASS-THROUGH states*.

The first panel is irradiated 80 % and 60% and the second is irradiated 40% and 30%, respectively on the first and second half of the simulation time.

Our simulation results validates the system. Namely, the system was able to produce the maximum estimated power, which proves the efficiency of the state controls in one hand and on the other hand, the fact that the system adapts itself to generate the maximum power in every conditions of irradiation. Relying on the obtained test data, the system does exactly what it is designed to do in a consistent manner.

IV. CONCLUSION

Using a case study, the DMPPT from [4], we have shown how Modelica can be effectively used to model CPS stemming from PESs. 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 modeled and validate using Modelica, which thus provides an open standard environment for a thorough simulation based analysis of PES designs.

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