Over Current Relay Modeling using Modelica with Cross-Verification against a Validated Model

Manuel Navarro Catalan*, Luigi Vanfretti†,
Department of Electrical Computer and Systems Engineering
Rensselaer Polytechnic Institute Troy, NY, USA
Email: *navarm2@rpi.edu, †luigi.vanfretti@gmail.com

***Abstract*—Relays are one of the most important elements for power systems protection. In this paper, an over current relay model is implemented in the Modelica language. Once all of the different modules were tested individually, the modules would be integrated to the relay model which then would be subjected to different fault simulations. Thereafter, the different types of relays were implemented and the Modelica model results are compared to a Simulink model previously implemented. Finally, a sample power system was implemented in order to evaluate the relays performance in a system. The results show how effective the relay model is and lays a foundation for further fault simulation, real-time simulation and model exchange.**

***Index Terms*—Over Current Relay, Modelica, Simulation, FMI**

I. Introduction

A. Motivations

Today's relays are microprocessor based devices, making them smaller, more accurate, and cheaper than their previous electro-mechanical incarnations. Because of their crucial role in power system protection, microprocessor relays require careful modeling for different types of simulation studies across multiple platforms, including real-time Hardware-in-the-Loop (HIL) simulation.

The main function performed by Over Current Relays (OCRs) is to send a trigger signal to open a breaker when the current exceeds a designated pick up current value. Extensive simulation studies and testing is needed during the design phase of protection coordination that include these devices in order to ensure proper over current protection. However, the majority of power system simulation platforms make it difficult, if not impossible, to exchange models of protection relays without ambiguity or loss of information. Recently, the Modelica language [1] and the Functional Mock-up Interface (FMI) [2] standards have become a viable alternative for model exchange, such as relay models. These standards have begun to be adopted by some power system tools that would allow the exchange of models from the design stage all through real-time (RT) simulation. To leverage these standards for power system protection, this paper describes an OCR model is implemented using the Modelica Language and the Modelica Standard Library (MSL).

The work of L. Vanfretti was supported in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under Award EEC-1041877, and the CURENT Industry Partnership Program.

The increased sensing and need of networking of Information and Communication Technologies (ICT) to support 'smart grid' functionalities is transforming power grids into Cyber Physical Systems (CPS) [3], [4]. When designing new 'smart grid' functionalities, a common approach is to use real-time Hardware-In-the-Loop (HIL) simulation. In this approach, real-time simulators are used to model the behavior of an power system so to test 'real world' relay devices with an ICT system in HIL configuration. However, it is very expensive and time consuming to place all of the relays in the loop. Therefore, a model like the one presented herein that can be exported using FMI into different platforms, provided that the real-time simulation platforms support such standards. This would allow to use a limited number of relays performing the most important functionalities in HIL configuration, while having other relays needed in a model coupled to the power grid performing conventional functions. Such approach would help mitigating development and testing time because the same relay models could be used during off-line studies used in the design phase, without the need of re-implementing them entirely in a specific real-time simulation platform.

Compared to other modeling languages, the Modelica Language offers a more versatile and useful alternative due to its openly standardized equation-based multi domain object oriented approach, and has shown to have great potential for power system protection applications [5]. The goal of the work is to recreate, test, and validate the OCR presented in [6], which has been validated against actual hardware. By using the Modelica language it is possible to take advantage of model exchange portability in RT and offline simulation, for example, using power system software such as EMTP-RV [7] and ePHASORSIM [8]. By using the FMI [2], the relay model could be used in a larger, real-time, and large-scale power system models in order to perform protective relay studies of complex networks such as [9].

B. Related Work

The potential of using the Modelica language to perform power system studies, and in particular Electro-Magnetic Transient (EMT) studies needed for power system protection, was first presented in [10]. Since then, several libraries for the study of different power system phenomena have appeared, with [11] summarizing different Modelica libraries and approaches available to date.

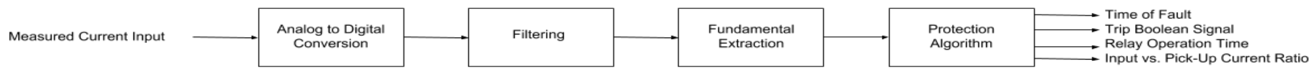

Fig. 1. Functional Diagram of an OCR.

For the scope of this paper, previous work related to EMT simulation and power system protection are of major interest. The work in [10] set the basis for accurate modeling of power system transients. In those days efficient simulation performance with Modelica tools was still challenging, which was one of the major challenges to adopt Modelica and Modelica-compliant tools for most power system studies to meet the performance of domain specific tools [12]. This has now begun to change [13]; in the last two years major breakthroughs in both proprietary tools (i.e. Dymola [14]) and Open Source Software (i.e. OpenModelica [15]) have been achieved.

With these advances the use of Modelica and Modelica-based tools is becoming more feasible for its use beyond research and academia in the power engineering domain, with major power grid players beginning to adopt this solution for their own studies [5], [16]. Of particular interest, a publication from the Electric Power Research Institute [17] describes the effective use of Modelica for the analysis of distribution network transients. Meanwhile, Électricité de France [5] describes how Modelica provides unmatched flexibility for protection relay modeling as compared to a domain-specific tool in the design and specification of a protection scheme where a zero sequence watt-metric relay. The authors highlight that in order to match specifications, Modelica models have a tremendous potential to be used towards a formal executable specification of power system equipment that can be instrumental in verifying relay performance.

The problem of relay performance verification is not trivial, and requires the use of real-time HIL simulation. HIL requires the re-implementation of the specified models in a specific simulation platform used by the real-time simulator. The authors in [6], present the design and implementation of a model of an over current relay implemented using Simulink, and deployed within Opal-RT's real-time targets. The model underwent HIL testing actual hardware relays. Using the results from this previous work, this paper proposes to develop the same over current relay model using the Modelica language, to test in a sample power system and against the model from [6], and to make it available as Open Source Software so that users of Modelica and FMI-based tools can re-utilize and improve upon these results.

C. Contributions

The contributions of this paper are the following:

- To provide an OCR model implemented using the Modelica language.
- To test the Modelica model against the reference Simulink implementation from [6]. The OCR is also testing using a power system example application.
- To enable further re-use and portability by making the model open source.

D. Paper Organization

The remainder of this paper is organized as follows. Section II will review the modeling and implementation of the OCR model. Section III performs cross verification of the relay model against of that in [6]. Section IV illustrates the use of a model in a simple power system study. Finally, Section V summarizes the results obtained and outlines future work.

II. Modeling and Implementation

A. Modeling

Fig. 1 shows the functional diagram of an OCR. The figure outlines the operations performed by the OCR, from the input current input signal to the trip time of the fault. Fig. 2 depicts the OCR and all of the elements that make it up. The OCR consists of the elements shown in Fig. 2: (a) Zero Order Hold (ZOH), (b) Low Pass Finite Impulse Response (FIR) filter, (c) Sampler, (d) Fundamental Filter, (e) RMS conversion, (f) Extraction of the Time of Fault module, (g) Timer module, and (h) Operation Time Calculation module. In addition, four different output signals have been designed for the correct analysis of the operation of the relay: (I) Time of Fault, (II) Trip Boolean Signal, (III) Relay Operation Time, (IV) Input vs Pick Up Current Ratio. Ultimately, the only output of the relay that will be linked to a power system via the breaker is the Trip Signal. The three other signals serve as a measure for the performance assessment of the model. As shown in Fig. 1, the firs step is to intake the OCR's current measurement through a current sensor from the power system. Next on Fig. 1 is the Analog to Digital Conversion, once the signal is received, the first part of the ZOH (b) on Fig.2 samples the continuous input signal per the sampling time given. Next, on Fig. 1 the input signal is filtered, the FIR filter removes the frequencies not wanted per the band pass frequency. Lastly, a sampler receives the signal and down samples it by a factor of two in order to prevent anti-aliasing effects. Both the FIR filter (b) and the sampler (c) are shown in Fig. 2. Then, the fundamental of the signal must be extracted, this happens at (d) on Fig. 2. Finally the Protection Algorithm initiates from the comparison of the pick up current and the input current.As shown in Fig. 2. The Protection Algorithm is made up of: (f) Extraction of the Time of Fault module, (g) Timer module, and (h) Operation Time Calculation module.

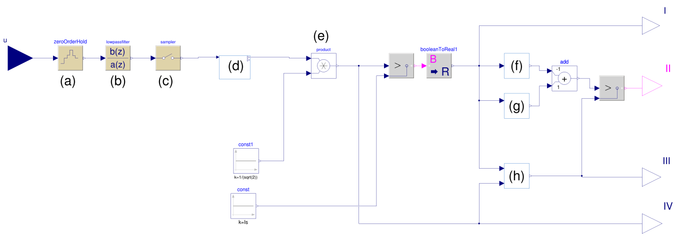

Fig. 2. Relay model.

B. Implementation

The implementation of the functions is carried out using the MSL. implementation of the model was approached form a bottom-up method, where each element was designed and tested individually before assembling the complete OCR model.

1) Timer Module: Shown in Fig. 3, this module has the task of counting the time for which the input current has been higher than the pick up current. The timer receives the input current and will output 0 until the input current is greater than the pick up current (this is when a fault happens). Then, the switch will activate and a timer will start counting the time since the fault happened.

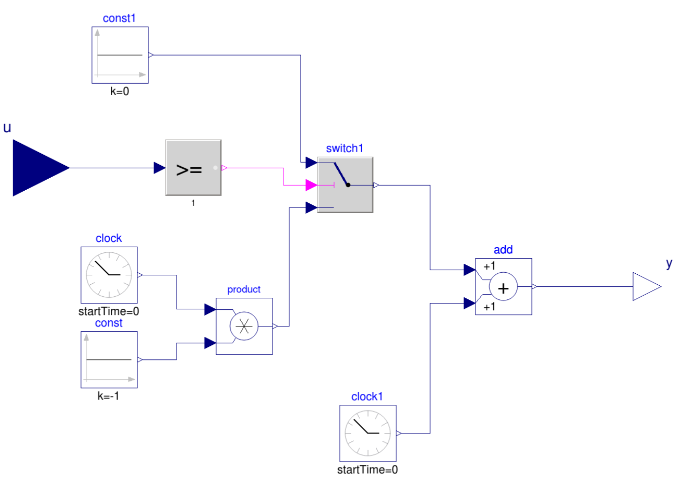

Fig. 3. Timer module.

2) Fundamental Filter Module: Fig. 4 shows the Fundamental Filter model developed. The purpose of this module is to receive the sampled current signal and output of the fundamental magnitude. The signal is split into sine and cosine components as shown in Fig. 4 in block (a), the signal is split into real (b) and imaginary (bi) components, then the mean is extracted using the subsystem 'Discrete Mean Value' (DMV) found in (c) for the real component and (ci) for the imaginary component. Finally, the magnitude is converted from rectangular to polar form in (d). The DMV subsystem was developed from scratch. The purpose of this module is to

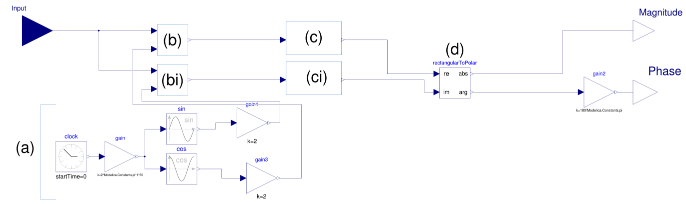

Fig. 4. Fundamental Filter module.

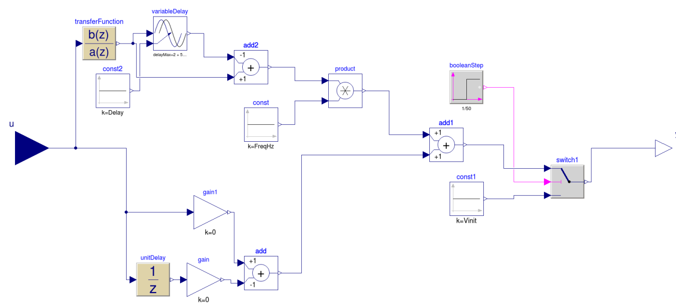

Fig. 5. Discrete Mean Value module.

output the mean of the signal during an average window of a cycle of a fundamental frequency. The final implementation of the DMV is shown in Fig. 5.

3) Calculation of Operating Time Module: As seen in Fig. 6, the Calculation of Operating Tine module calculates, using the equation:

$$T = \frac{C}{\frac{I^{\alpha}}{I_s} - 1} * TMS \tag{1}$$

where T - relay operating time, C - relay characteristic constant, Is - pick up current, I - input current, α - inverse time type constant (greater than zero), and TMS - time multiplier setting. The operating time of the relay is governed by the type of relay that is being modeled, which is varied depending on the C and α values as seen in Table 1.

One important implementation consideration is that when I = Is, the denominator will equal 0, causing a division by 0 error. This was addressed by implementing the `y3_1` block highlighted in a red box in Fig. 6, which takes the result of the denominator into the block and performs the following operation:

$$y = \frac{1}{max(x, eps)} \tag{2}$$

where eps is the tolerance level and x is the denominator. The time for which the relay will trip the on signal will be affected by the tolerance that is set. Each type of relay has

TABLE I
TYPES OF RELAYS DEPENDING ON C AND α VALUES.

Type of Relay	α	C
Standard Inverse (SI)	0.02	0.14
Very Inverse (VI)	1	13.5
Extremely Inverse (EI)	2	80
Long Inverse (LI)	1	120

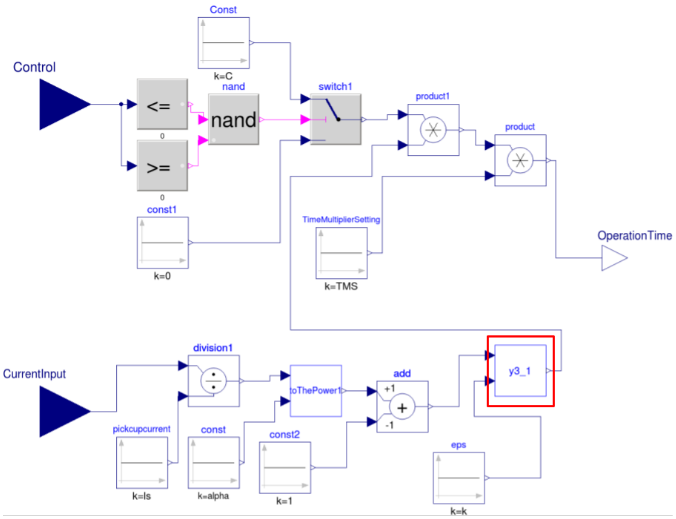

Fig. 6. Calculation of Operating Time module.

a different tolerance level in order to obtain the correct trip signal. For the Standard Inverse relay $eps = 0.041\%$, for the Very Inverse relay $eps = 0.07\%$ and for the Extremely Inverse relay $eps = 0.102\%$.

4) Extracting Time of Fault Module: This model is simple and its purpose is to detect the fault and output the time of the fault. Fig. 7 depicts the Extracting Time of Fault module. When the signal input u is received by the module comparing if its greater or equal to one. If true, then output will be the time for which the fault happened, if false, then the output is 1.

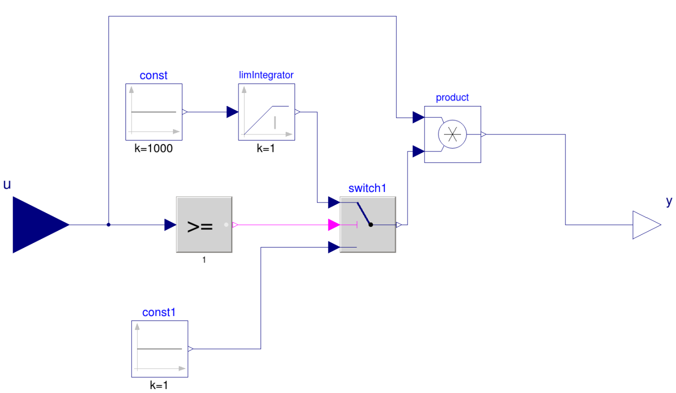

Fig. 7. Extracting Time of Fault module.

III. Cross Verification

In order to verify if the OCR model performs adequately it was necessary to capture the input signal that goes into the OCR (from Simulink) and reproduce it in Modelica. Fig. 8 displays the input relay signal.

The signal is provided to the Modelica model using a time table, that acts as an input source for the relay and ensures a common input when comparing both implementations (Simulink vs. Modelica). Fig. 9, shows how the time table

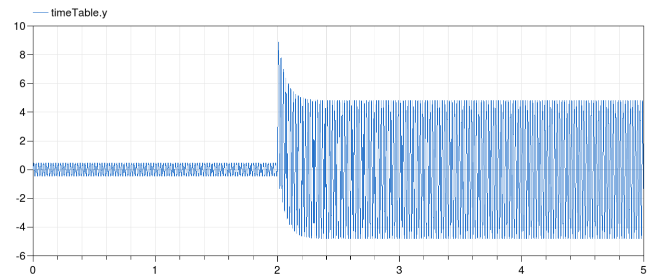

Fig. 8. Input Fault Current Signal (Modelica).

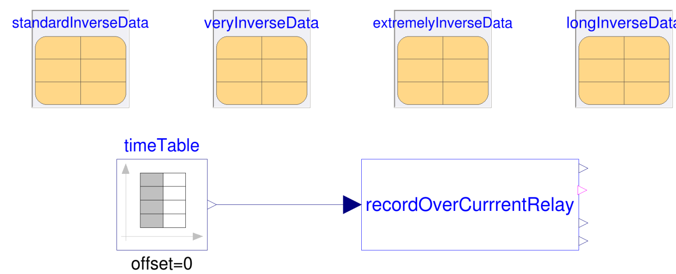

Fig. 9. Implementation of the relay model with the time table as a source.

is a source. Because there is more than one type of relay to characterize each within the same model, the information is encapsulated in a `package` that contains all data required to model each relay type. Fig. 9 also shows how `Modelica` `records` were created in order to implement the four relay types and easily switch between them.

As shown in Fig. 10, the results are satisfactory. The model attained similar as the Simulink model from [6]. On average, the trip time discrepancy was 0.44% between the trip signal in Simulink vs the trip signal in Modelica. These discrepancies are likely due to the different numerical solvers used in the simulation tools.

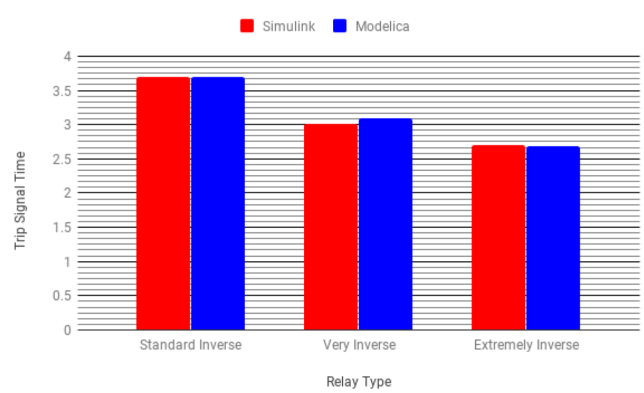

Fig. 10. Signal Trip Time depending on the relay type.

IV. Simulation Performance and Impact on Realy Performance

In Modelica-compliant tools, in order to perform a simulation, the model needs to be flattened, teared, and finally transformed into the programming language for which the there are various different types of integration methods. The user must choose the numerical solver to be used for this purpose. In this process, the first step is to create the model in the Modelica language and choose the solver to be used. The second step, carried out automatically by the Modelica tool, is to convert the model to C language including the numerical simulation solver chosen by the user.

The user can repeat this process to and select different built-in solvers in order to asses the validity of the solution or to determine the most efficient solver for the model chosen [18]. The Dymola software provides 10 different variable step-size integration's methods that can be used, in addition to multiple fixed time step simulation solvers. Observe that power system tools do not provide any choice of solvers, and typically use a trapezoidal integration routine, which makes simulation very slow. To illustrate the advantages of Modelica tools, for all of the simulations presented on this paper, the Dassl solver was used.

To understand the choice of this solver, this section presents results on using all different variable time step solvers. In addition, this is also useful to determine which solvers are not suitable for use with the relay model. To asses the impact of the solver on the relay model's simulation, a trip time signal for a SI OCR was compared for the following solvers: Dassl, LSODAR, Dopri45, Esdirk45a and Cerk45. For these integration methods, the model successfully compiled and gave the same output of 3.7 seconds. However, the elapsed time for the simulation was drastically different. The parameters for the solver were: 500 intervals, .0001 tolerance, 0 start time and 5 seconds end time. The comparison between simulation time with different solvers is seen in Fig. 11.

As seen in Fig. 11, the best solvers to use are Dassl and LSODAR, which take around 15.7 seconds to perform a simulation. However, the time to integrate for the other three

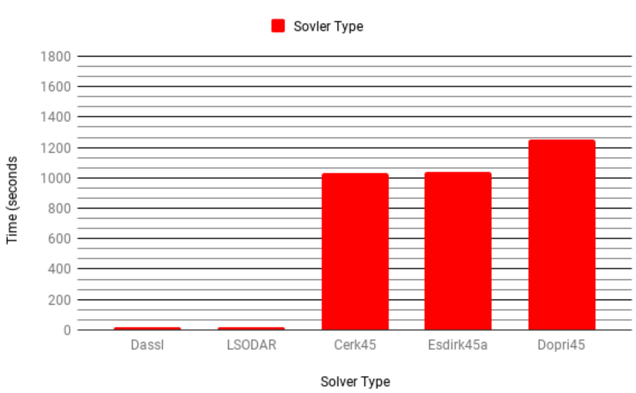

Fig. 11. Solver Simulation Time Comparison

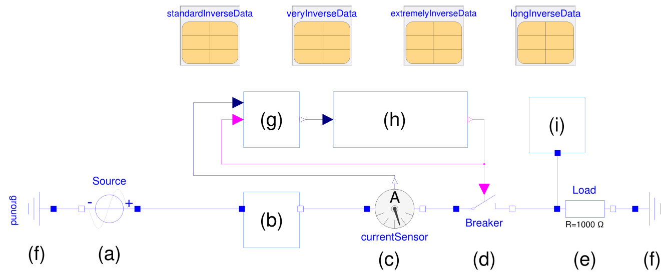

Fig. 12. Power system with OCR

solvers (Dopri45, Esdirk45a and Cerk45) increase dramatically to around 1000 seconds. Observe that all solvers had the same trip signal output time for the OCR, hence, solver accuracy was not an issue.

V. Application Example

To illustrate the use of the model for power system protection studies, Fig. 12 shows a simple power system containing the relay and the following components: (a)1 phase 230 kV voltage source, (b)1 phase 1 km π model transmission line, (c) current sensor, (d) 1 phase circuit breaker, (e)1 phase resistive load, (f) ground, (g) current transformer, (h) the OCR, and (i) a line to ground fault. This power system was taken from an example found in [19].

In Section III, there were 4 outputs that were measured in order to asses how the relay was working (Time of Fault, Trip Boolean Signal, Relay Operation Time, Input vs Pick Up Current Ratio). However, In Fig. 12, it shows how only one of the outputs is utilized: the Trip Signal. This is a Boolean signal where the relay Fig. 12 (h) will alert the breaker Fig. 12 (d) whether a fault has been detected and signal when the breaker should open/close. therefore the only links between the power system and the relay model are: the real system current input and the Boolean output that communicates with the breaker.

Fig. 13 is a snapshot of the source current, load current and the relay trip signal for an extremely inverse relay model. From the top subplot of Fig. 13, the source current spikes at t = 2 secs, and then goes to 0 A, at t =2.68 secs. The load current in the middle subplot in Fig. 13, shows how it goes to 0 A at t = 2.68 secs. Finally, the trip Boolean signal on the bottom subplot of Fig. 13 becomes true at t = 2.68 secs. This shows the correct operation of the OCR.

The fault contains a switch with a resistance of 1000 Ω that is activated after 2 seconds. The relay was evaluated by looking at the trip time value depending on the type of relay as seen in Table 1. The results were satisfactory. On average, the discrepancies of all of the different relay types were just 0.40% off of the trip values from [6] as seen on Fig. 14. Discrepancies are likely due to the different numerical solvers used by each tool.

VI. Conclusion

This paper described the implementation of an OCR model in Modelica. The tests performed on the model were satisfac-

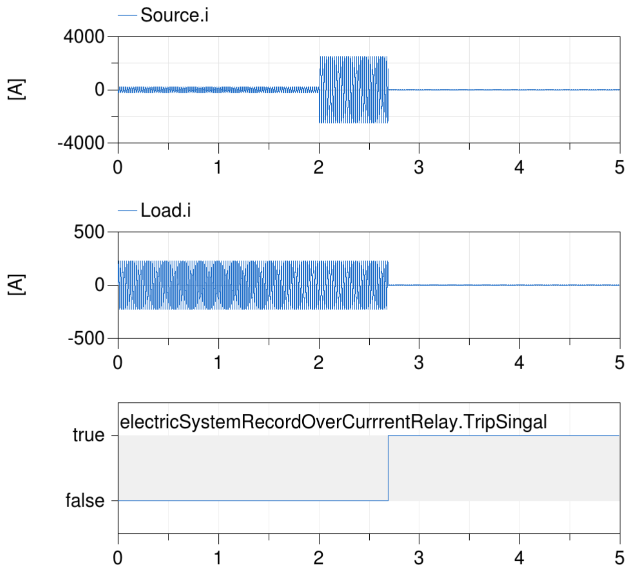

Fig. 13. OCR power system result.

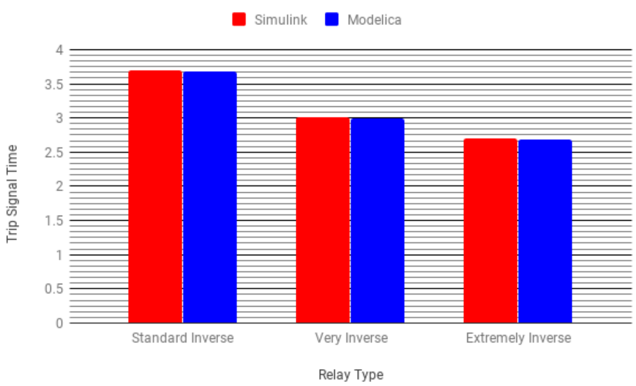

Fig. 14. Signal Trip Time depending on the relay type.

tory. The cross verification matched the results obtained in [6]. The power system application example gave further validation in cross verification, assuring that them model provides the expected behavior of an OCR relay in a power system.

In future work, the model will be assessed in other tools using the FMI [2] standard in order to run real-time simulation in ePHASOR SIM [8] and EMTP-RV [7]. To enable re usability and portability the model above has been made available as open source software in the following GitHub repository, facilitating research reproducibility and reuse:

`https://github.com/alsetlab/2019_clayton_griffin_overcurrentrelay`

REFERENCES

[1] P. Fritzson, *Principles of object-oriented modeling and simulation with Modelica 2.1*. John Wiley & Sons, 2010.

[2] "Functional Mock-Up Interface, howpublished = https://fmi-standard.org/, note = Accessed: 2019-01-30."

[3] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.

[4] A. Stefanov and C.-C. Liu, "Ict modeling for integrated simulation of cyber-physical power systems," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*. IEEE, 2012, pp. 1–8.

[5] O. Chilard, J.-P. Tavella, and O. Devaux, "Use of modelica language to model an mv compensated electrical network and its protection equipment : comparison with emtp," in *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany*, no. 63. Linkping University Electronic Press; Linkpings universitet, 2011, pp. 406–413.

[6] M. S. Almas, R. Leelaruji, and L. Vanfretti, "Over-current relay model implementation for real time simulation & hardware-in-the-loop (hil) validation," in *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2012, pp. 4789–4796.

[7] "Multi-time-step and multi-core simulations in EMTP using the new FMI options, howpublished = http://emtp-software.com/page/fmi-options, note = Accessed: 2019-01-30."

[8] V. Jalili-Marandi, "ephasorsim:real-time transient stability simulation tool," Event Lecture, 2015.

[9] J. Mahseredjian, V. Dinavahi, and J. A. Martinez, "Simulation tools for electromagnetic transients in power systems: Overview and challenges," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1657–1669, 2009.

[10] B. Bachmann and H. Wiesmann, "Advanced modeling of electromagnetic transients in power systems," in *Proceedings of the Modelica Workshop 2000*, 09 2000, pp. 93–97.

[11] D. Winkler, "Electrical power system modelling in modelica comparing open-source library options," in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th 27th, 2017*, no. 138. Linkping University Electronic Press, Linkpings universitet, 2017, pp. 263–270.

[12] J. Mahseredjian, V. Dinavahi, and J. A. Martinez, "Simulation tools for electromagnetic transients in power systems: Overview and challenges," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1657–1669, July 2009.

[13] F. Casella, A. Leva, and A. Bartolini, "Simulation of large grids in openmodelica: reflections and perspectives," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, no. 132. Linkping University Electronic Press, Linkpings universitet, 2017, pp. 227–233.

[14] E. Henningsson, H. Olsson, and L. Vanfretti, "Dae solvers for large-scale hybrid models," in *Proceedings of the 13th International Modelica Conference*, 03 2019.

[15] W. Braun, F. Casella, and B. Bachmann, "Solving large-scale modelica models: new approaches and experimental results using openmodelica," in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, no. 132. Linkping University Electronic Press, Linkpings universitet, 2017, pp. 557–563.

[16] L. Vanfretti, M. A. A. Murad, F. G. Lpez, G. Len, S. Machado, J. Heyberger, and S. Petitrenaud, "Towards automated power system model transformation for multi-tso phasor time domain simulations using modelica," in *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Oct 2016, pp. 1–7.

[17] T. A. Short, "Open-source distribution transient modeling with modelica," in *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, April 2008, pp. 1–5.

[18] Y. Sun, "Friedrich-alexander-universität erlangen-nurnberg."

[19] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.