January 31st, 2019

Dr. Meliopoulos,

Enclosed in this email you will find my submission for the 2019 Clayton Griffin student paper award paper competition. The paper is titled “Modeling Over Current Relays using Modelica”. This paper is the result from my research collaboration with my advisor Dr. Luigi Vanfretti during my doctoral studies at Rensselaer Polytechnic Institute. With this letter I include my paper and the author biographies. Thank you for your consideration.

Sincerely,

Manuel Navarro Catalan
Enclosure

Manuel Navarro Catalan
PhD Student
Electrical Computer and Systems Engineering

Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
USA

E-mail: navarm2@rpi.edu
Web: ALSETLab.com
iPhone: +1 (210) 990-9709i

# Modeling Over Current Relays using Modelica

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 ancestors. 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. In the case of an over current relays (OCRs), a relay activates a breaker when the current exceeds a designated pick up current value. Extensive simulation studies and testing is needed for such 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).

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. The goal of the work is to recreate, test, and validate the OCR presented in [3], 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 [4] and ePHASORSIM [5]. 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 [6].

L. Vanfretti is 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.

### B. Related Work

The authors in [3], present the design and implementation of a model of an over current relay implemented using Simulink. The model underwent HIL testing actual hardware relays. In the current 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 [3].

### 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 [3]. 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 describes the implementation of the model. Section III performs cross verification of the relay model against of that in [3]. 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. Implementation

Figure 1 shows the functional diagram of an OCR. The implementation of the functions is carried out using the MSL. Each element was designed and tested individually before assembling the OCR model. As shown in Fig. 2, the OCR consists of the following elements: (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

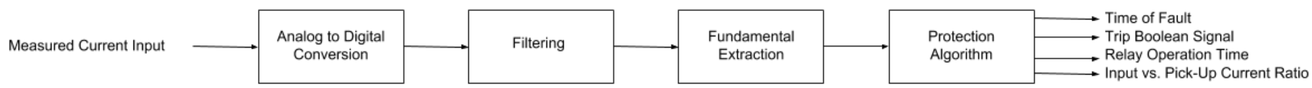


Fig. 1. Discrete Mean Value module.

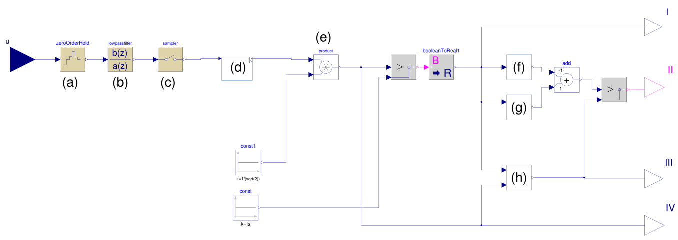


Fig. 2. Relay model.

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. Three other signals serve as a measure for the performance assessment of the model. Once the current signal is received, the first part of the ZOH samples the continuous input signal per the sampling time given. Next, 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. Fig. 2 depicts the final OCR model assembled using the sub-systems implemented. Four models have to be implemented form scratch, while others only required for their implementation to assemble together existing modeling blocks from the MSL. The models built from the ground up are: the timer, the fundamental filter, operation time calculation and extraction time of fault.

## A. Fundamental Filter Module

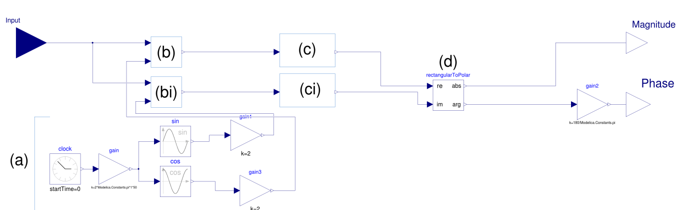


Fig. 3. Fundamental Filter module.

Figure 3 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. 3 in block (a), the signal is split into real (b) and imaginary (bi) components, then the mean is extracted using the subsystem

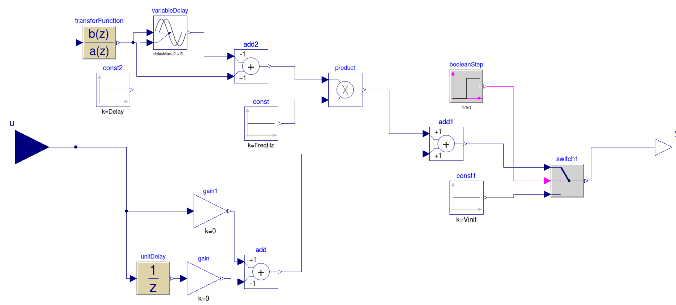


Fig. 4. Discrete Mean Value module.

’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 output the mean of the input signal. The final implementation of the DMV is shown in Fig. 4.

## B. Timer Module

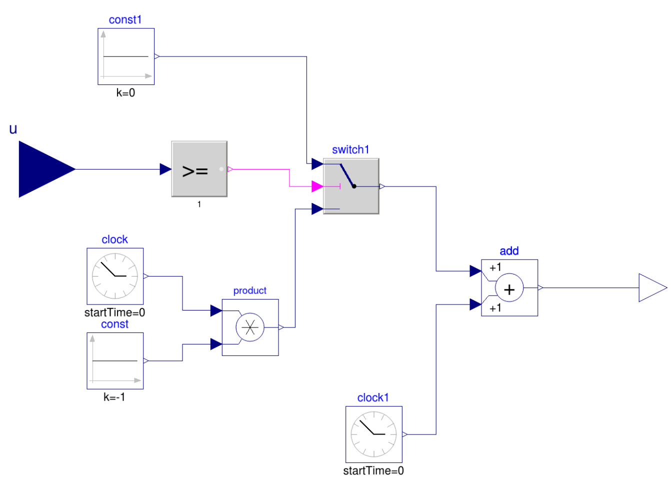


Fig. 5. Timer module.

Shown in Fig. 5, this module has the task of counting the time for which the input current has been higher than the pick up current.

## C. Calculation of Operating Time Module

As seen in Fig. 6, this module calculates, using

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

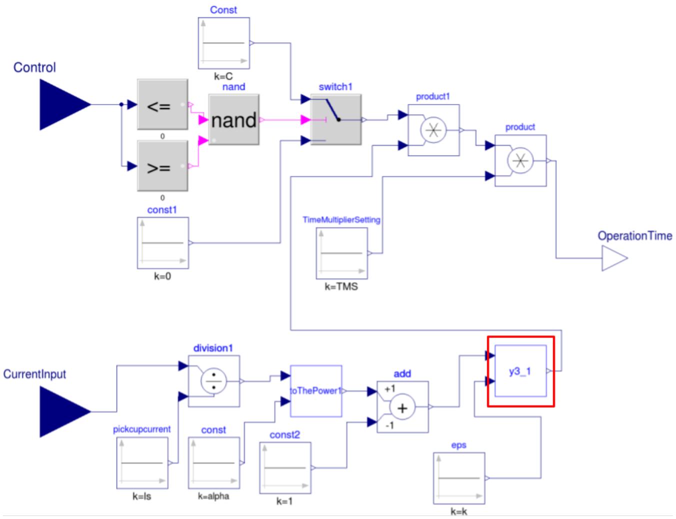


Fig. 6. Calculation of Operating Time module.

where $T$ - relay operating time, $C$ - relay characteristic constant, $Is$ - pick up current, $I$ - input current, $\alpha$ - 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 $\alpha$ values as seen in Table 1.

TABLE I
TYPES OF RELAYS DEPENDING ON C AND $\alpha$ VALUES.

| Type of Relay | $\alpha$ | C |
|---|---|---|
| Standard Inverse (SI) | 0.02 | 0.14 |
| Very Inverse (VI) | 1 | 13.5 |
| Extremely Inverse (EI) | 2 | 80 |
| Long Inverse (LI) | 1 | 120 |

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 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 $eps = 0.102\%$.

### D. 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.

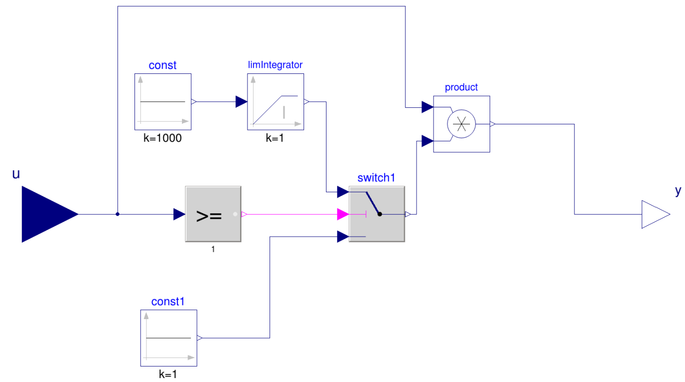


Fig. 7. Extracting Time of Fault module.

## III. CROSS VERIFICATION

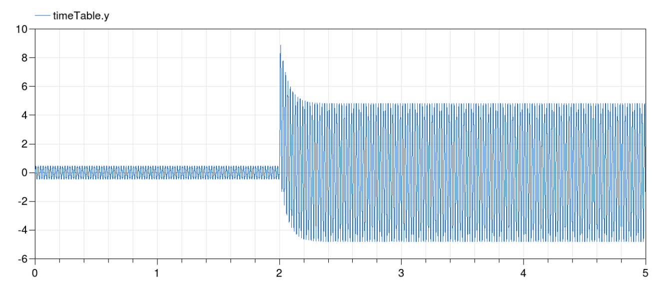


Fig. 8. Input Fault Current Signal (Modelica).

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. Figure 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). Figure 9, shows how the time table 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.

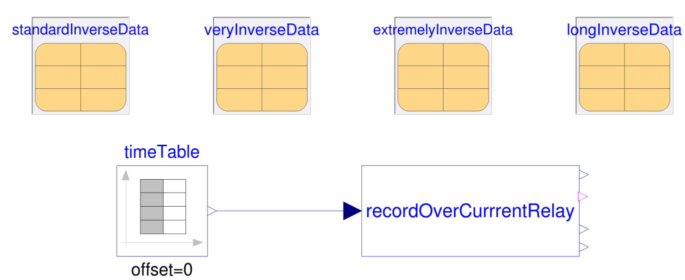


Fig. 9. Implementation of the relay model with the time table as a source.

As shown in Fig. 10, the results are satisfactory. The model attained similar as the Simulink model from [3]. 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.

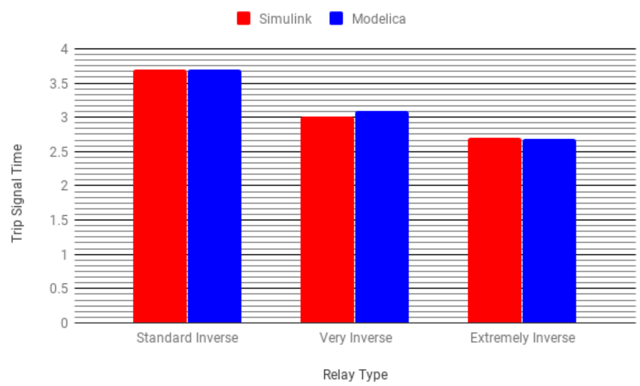


Fig. 10. Signal Trip Time depending on the relay type.

## IV. Application Example

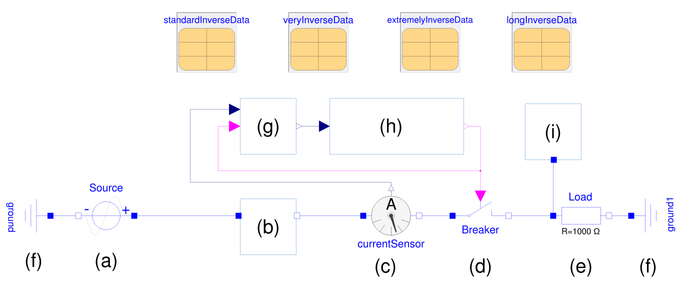


Fig. 11. Power system in which the relay is introduced.

To illustrate the use of the model for power system protection studies, Fig. 11 shows a simple power system containing the relay and the following components: (a) 1 $\phi$ 230 kV voltage source, (b) 1 $\phi$ 1 km $\pi$ model transmission line, (c) current sensor, (d) 1 $\phi$ circuit breaker, (e) 1 $\phi$ 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 [7].

The fault implemented is a line to ground fault which, switched with a resistance of 1000 $\Omega$ which will switch on at 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 [3] as seen on Fig. 12. The values of the relay trip time are similar. The discrepancies are likely due to the different numerical solvers used by each tool.

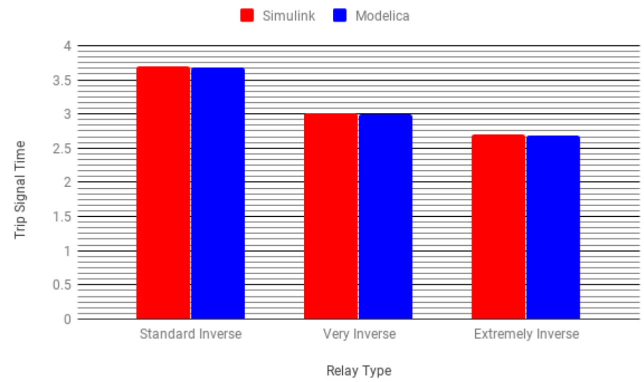


Fig. 12. Signal Trip Time depending on the relay type.

## V. Conclusion

This paper described the implementation of an OCR model in Modelica. The tests performed on the model were satisfactory. The cross verification matched the results obtained in [3]. The power system application example gave further validation in cross verification and assuring that them model provides the expected behavior of a relay in a power system. In future work, the model will be assessed in other tools using the FMI [2] in order to run real time simulation in ePHASOR SIM [5] and EMTP-RV [4]. To enable re usability and portability the model above has been made available as open source software in a GitHub repository [8], also facilitating research reproducibility.

## 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] 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.

[4] "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."

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

[6] 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.

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

[8] "Clayton Griffin Over Current Relay Paper Repository, howpublished = https://github.com/alsetlab/2019_clayton_griffin_overcurrentrelay, note = Accessed: 2019-01-30."

Manuel Navarro Catalan
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, USA

Manuel Navarro Catalan received the B.S. degree in electrical engineering from The University of Texas at Austin in 2018. In 2017, he was a research assistant at the Royal Institute of Technology in Stockholm, Sweden. He is currently pursuing the Ph.D. degree in power engineering at Rensselaer Polytechnic Institute. His research interests include power system protection, modeling and simulation of cyber-physical systems.

Luigi Vanfretti
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, USA

Luigi Vanfretti (SM'14) received the M.Sc. and Ph.D. degrees in electric power engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2007 and 2009, respectively. He was with the KTH Royal Institute of Technology, Stockholm, Sweden, as an Assistant, from 2010 to 2013, and as an Associate Professor (Tenured) and a Docent from 2013 to 2017, where he led the SmarTS Lab and the Research Group. He was with the Statnett SF, the Norwegian Electric Power Transmission System Operator, as a Consultant, from 2011 to 2012, and as a Special Advisor in research and development from 2013 to 2016. He joined the Rensselaer Polytechnic Institute in 2017, where he is currently a Tenured Associate Professor. His research interests are in the area of sycnhrophasor technology applications, cyber-physical power system modeling, simulation, stability, and control.