

## Non-directional Overcurrent Protection Relay Testing Using Virtual Hardware-in-the-Loop Device

Le Nam Hai Pham, Raju Wagle, Francisco Gonzalez-Longatt, Martha Nohemi Acosta Montalvo

**Abstract.** Protective relays are an integral part of the power grid, making it reliable and secure against abnormal conditions. *Hardware-in-the-Loop* (HIL) can be employed to test and validate digital protective relay devices in the real-time simulation comprising hardware and software before actual implementation. However, the HIL technique requires a complex and high-cost technical environment consisting of hardware devices, sensors, communication, and simulation platforms. The Virtual HIL (VHIL) Device is a new real-time simulation approach to offer the entire experience and challenges coming from the actual implementation of HIL. It allows the users to approach the possibility of learning the principles and techniques used in HIL without any concern related to using real hardware or physical devices. This chapter aims to illustrate the use of the model-based system engineering toolchains of Typhoon HIL for understanding the implementation methodology of the VHIL technique, from creating a testing model to the process of running. A modified version of the well-known three-phase radial feeder of the European MV distribution benchmark system created by CIGRE Task Force C6.04.02 is modelled and simulated for evaluating non-directional overcurrent protective relay performance through multiple short-circuit fault scenarios. The main contribution of this chapter is to systematically introduce the modelling and simulation for testing purposes of non-directional overcurrent protection relay in VHIL that helps power engineers evaluate the protective relay settings under more realistic conditions.

**Keywords:** Hardware-in-the-loop, non-directional overcurrent, real-time simulation, Typhoon HIL, virtual protection relay.

---

Le Nam Hai Pham  
University of South-Eastern Norway, Porsgrunn, Norway  
e-mail: Le.Pham@usn.no

Raju Wagle  
UiT The Arctic University of Norway, Narvik, Norway  
e-mail: raju.wagle@uit.no

Francisco Gonzalez-Longatt  
University of South-Eastern Norway, Porsgrunn, Norway  
Centre for Smart Grid, University of Exeter, Exeter, United Kingdom  
e-mail: fglongatt@fglongatt.org

Martha Nohemi Acosta Montalvo  
University of South-Eastern Norway, Porsgrunn, Norway  
e-mail: Martha.Acosta@usn.no

## 1 Introduction

Power system protection plays a vital role in achieving a satisfactory level of reliability and security in power systems. Relaying power system protection is a branch of power system concerned with the principles of design and operation of equipment called protection relays or protective relays (or simply called relays). A protection relay is a device that detects abnormal power system conditions and initiates corrective action as quickly as possible to return the power system to its normal state [1].

One of the most functions of a protective relay that is used in distribution networks is overcurrent protection. The protection relays used in such a function could be directional (operating for in-front events) and non-directional (will work for all) depending upon the topology of the distribution system [2]. The non-directional relays are mostly used for radial distribution feeders, the most common type of worldwide distribution system. Currently, it is used in combination with the directional overcurrent protection relay according to the development of modern networks, including more than one source or multiple lines conveying electrical power in different directions.

Testing protection relay in software and hardware environment is vital in ensuring the relay setting configuration in real-world systems. *Hardware-in-the-Loop* (HIL) is a developing technology used in developing and testing complex real-time systems by providing an effective platform to help validate software systems on specially equipped test benches receiving real input data from physical devices [3]. The HIL technology has been used by several authors using the testing platform with different optimisation models [4]–[8].

Testing processes have been developed to be applied for a variety of subsystems and functionalities in the protection systems: SCADA systems [9], communication platforms [10], overcurrent and directional overcurrent relays [11], and distance protection systems as well [12]. For example, the authors in [13] investigate the behaviour of instantaneous overcurrent protection using an OPAL-RT digital simulator through HIL testing. In [14], virtual distance protection relay testing was conducted through HIL simulation in PSCAD software. However, the HIL technique requires a complex technical environment consisting of hardware devices, sensors, communication, and simulation platforms. For instance, the authors in [12] rely on micro-controllers costing around 5000 USD for a single testing platform [14]. To allow the users to approach the possibility of learning the principles and techniques used in HIL without any concern related to using real hardware or physical devices, Virtual HIL (VHIL) Device is a new technique to offer all the experience (including challenges) coming from the real implementation of HIL.

This chapter aims to illustrate the use of model-based system engineering toolchains of Typhoon HIL to understand the implementation methodology for evaluating non-directional overcurrent protection relay performance, from creating a testing model to the process of running.

A practical and illustrative example of “How to” create a model and simulate it inside the Typhoon HIL Control Centre environment using the “Virtual HIL Device” is provided. A simple radial distribution is used for illustrative purposes through multiple short-circuit scenarios: three-phase, three-phase-to-ground, single-phase-to-ground,

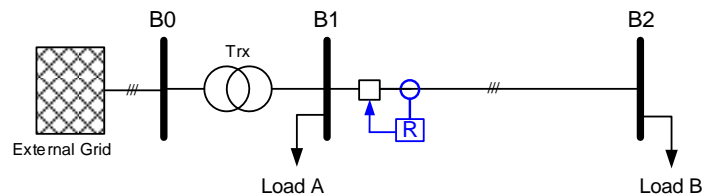
phase-to-phase, and two-phase-to-ground.

This chapter starts by introducing the problem and giving an overview of the overcurrent protection function of the protection relay. The workflow inside Typhoon HIL Control Centre is explained. The model creation starts with the use of HIL Schematic Editor; then, the methods of creating HIL SCADA are introduced to illustrate how to display numerical results and visualise signals. Finally, simulation results through short-circuit fault scenarios are displayed, and a discussion of the results is performed.

## 2 Problem Definition

This section is dedicated to modelling and simulating of non-directional overcurrent protection relays using a digital real-time simulation framework. Since overcurrent protection is well-known for being used in distribution systems, the authors have decided to use a straightforward radial system as a representative test system. However, all the methods explained in this chapter can be extended to more complicated topologies of distribution systems.

The authors started by considering the well-known three-phase radial feeder of the European MV distribution benchmark system created by CIGRE Task Force C6.04.02 [15]. However, for the sake of simplicity, the authors decided to use a modified version of the system. In this chapter, an equivalent of the CIGRE European MV distribution benchmark system is used; it is represented by a three-bus as shown in Fig. 1, and it is called the “*Test System*” from here onwards.



**Fig. 1.** *Test System*. Simplified equivalent model of the three-phase radial feeder of the European MV distribution benchmark system.

The 50 Hz *Test System* considers two voltage levels, 110 kV representing the connection to the external grid and 20 kV as the nominal voltage of a primary distribution feeder. The *Test System* consists of three buses, a lumped equivalent of an extensive network called “External Grid” at a nominal voltage of 110 kV (RMS, line-to-line) at bus B0; from there, a step-down two winding power transformer “Trx” is used to reduce the voltage to 20 kV. Buses B1 and B2 are connected throughout an overhead transmission distribution line called “Line 12”. On buses, B1 and B2, two lumped loads named “Load A” and “Load B” are connected to each bus.

**Table 1.** External Grid equivalent network parameters

Nominal system voltage [kV]	Short-circuit power [MVA]	R/X ratio
110	5000	0.1

**Table 2.** Transformer “Trx” parameters

Connection	Primary voltage [kV]	Secondary voltage [kV]	Rated power [MVA]
3-ph Dyn1	110	20	25

**Table 3.** Transmission line parameters with  $R'_{ph}$ ,  $X'_{ph}$ ,  $B'_{ph}$ ,  $R'_0$ ,  $X'_0$ ,  $B'_0$  that are positive and zero sequence resistance, reactance, and susceptance values, respectively.

$R'_{ph}$ [ $\Omega$ /km]	$X'_{ph}$ [ $\Omega$ /km]	$B'_{ph}$ [ $\mu$ S/km]	$R'_0$ [ $\Omega$ /km]	$X'_0$ [ $\Omega$ /km]	$B'_0$ [ $\mu$ S/km]	Length [km]
0.501	0.716	47.493	0.817	1.598	47.493	2.82

**Table 4.** Electrical parameters considered at the loads.

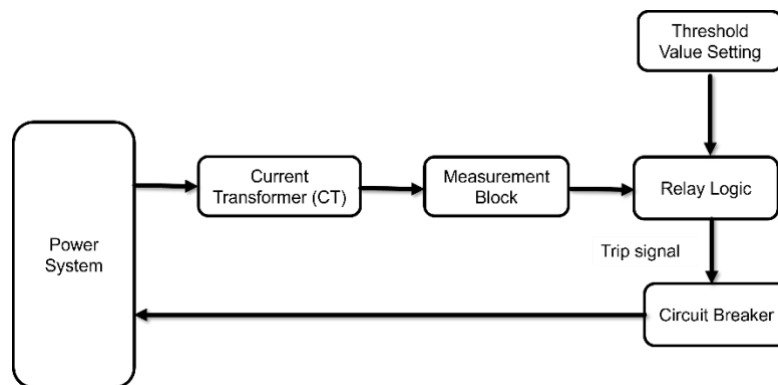
Load names	Apparent power [MVA]	Power factor	Load current [kA]
Load A (Commercial)	15.30	0.98 (ind)	0.43
Load A (Residential)	5.10	0.95 (ind)	0.14
Load B (Commercial)	3.48	0.97 (ind)	0.09
Load B (Residential)	1.11	0.85 (ind)	0.03

A non-directional overcurrent protection relay is connected to the overhead distribution line “Line 12” sending terminal. The protection relay located at the beginning of the overhead line offers feeder protection. In case of short circuit fault situations, it must be prevented from spreading to healthy parts of the network to ensure that the power grid continues to supply energy. The main requirements of the feeder protection are as follows:

- During the short circuit, the circuit breaker nearest to the fault should open, and all other circuit breakers remain in a closed position.
- If the circuit breaker nearest to the fault fails to open, then backup protection should be provided by the adjacent protection relay ordering the opening of its circuit breaker.
- The operating time of the protection relay should be small enough to maintain the system stability without the need to trip a circuit.

The classical non-directional overcurrent protection relays use a straightforward protection principle; the protection relay senses the feeder current through the secondary winding of the *current transformer* (CT) and compares it with a predetermined threshold value. The protective relay operates if the sensed current is above the threshold and the trip signal is generated and sent to the tripping circuit of the *circuit breaker* (BK) to isolate the faulty feeder from the rest of the healthy remaining part of the system.

Depending on the technologies (analogue or digital), the non-directional overcurrent protective relay may have a specific internal design, **Fig. 2** shows a simplified block diagram of a non-directional overcurrent relay and its signal flows with the other components of the protection system.



**Fig. 2.** Block diagram of overcurrent protection relay operation.

The operating characteristics of non-directional overcurrent protection relays are defined by the operating times typically governed by a time versus current magnitude curve ( $I-t$ ).

Considering the modern protection practice, there are three main types of time-current operating characteristics:

- *Instantaneous.*
- *Time-dependent.* Definite time and inverse-time.
- *Mixed.* Combination of the previously stated.

The *instantaneous overcurrent function* is characterised by no time delay intentionally provided. Nowadays, the operating time of the *instantaneous overcurrent function* is less than ten milliseconds and is sometimes expressed in cycles based on the power system frequency [16]. The *instantaneous overcurrent function* has only the pickup current setting and does not have any time delay setting. The pickup current is adjustable, and the application engineer can choose various settings from a wide range. The operating characteristics of this type can be shown in **Fig. 3**.

The *time-dependent overcurrent function* characteristics have been standardised in modern times to allow the protection engineers a coherent use of the protection characteristics. As the name implies, these overcurrent protection functions operate with an intentional time delay. The minimum current at which the protection relay operates (pickup current) and the delay time before the trip are both adjustable. Two types of time-dependant overcurrent functions are broken into two categories: *Define time* and *inverse-time*.

The *define-time* protection relays operate with some intentional time delay and are adjustable along with the current pickup level. *Inverse-time* protection relays have an operating time depending on the magnitude of the current, generally with an inverse characteristic (the operating time of the overcurrent protection relay is smaller as the current gets more significant).

The *inverse-time* operating characteristics have evolved over time, and then standardisation allowed homologating them. Two primary standards are used worldwide, *The American standard IEEE C37.112-1996* [17] and *the European IEC 60255* [18]. The inverse-time operating characteristic according to two standards are shown in **Fig. 4**.

**Table 5** and **Table 6** show the equations associated with US curves and IEC curves, respectively; where  $tp$  is operating time in seconds,  $TD$  is time-dial setting,  $TMS$  is time multiplier setting, and  $M$  is applied multiples of pickup current.

The *mixed overcurrent function* consists of the different types of overcurrent protection relay elements packed into a single programmable unit. The combination of instantaneous, definite-time, and inverse-time elements may be used in this type of protection relays.

**Table 5.** Equations associated with US curves following *The American standard IEEE C37.112-1996* [17]

Curve type	Relay operating time	Equation
U1 (Moderately Inverse)	$t_p = TD \times \left( 0.0226 + \frac{0.0104}{M^{0.02-1}} \right)$	(1)
U2 (Inverse)	$t_p = TD \times \left( 0.18 + \frac{5.95}{M^2-1} \right)$	(2)
U3 (Very Inverse)	$t_p = TD \times \left( 0.0963 + \frac{3.88}{M^2-1} \right)$	(3)
U4 (Extremely Inverse)	$t_p = TD \times \left( 0.0352 + \frac{5.67}{M^2-1} \right)$	(4)
U5 (Short-Time Inverse)	$t_p = TD \times \left( 0.0262 + \frac{0.00342}{M^{0.02-1}} \right)$	(5)

**Table 6.** Equations associated with IEC curves following *The European IEC 60255* [18]

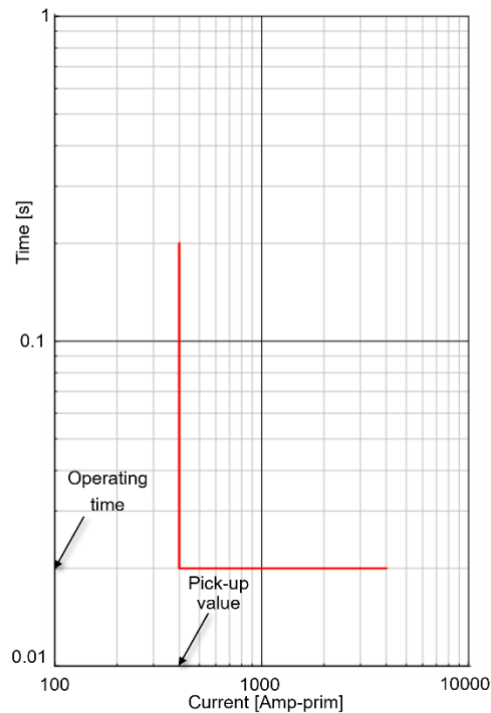
Curve type	Relay operating time	Equation
C1 (Standard Inverse)	$t_p = TMS \times \left( \frac{0.14}{M^{0.02-1}} \right)$	(6)
C2 (Very Inverse)	$t_p = TMS \times \left( \frac{13.5}{M-1} \right)$	(7)
C3 (Extremely Inverse)	$t_p = TMS \times \left( \frac{80}{M^2-1} \right)$	(8)
C4 (Long-Time Inverse)	$t_p = TMS \times \left( \frac{120}{M-1} \right)$	(9)
C5 (Short-Time Inverse)	$t_p = TMS \times \left( \frac{0.05}{M^{0.04-1}} \right)$	(10)

ANSI/IEEE standards are used to define the protection operating characteristics in America, while IEC standards are a common practice in Europe. Therefore, it is typical to use some codes and nodes when practically referring to protection functions according to these standards.

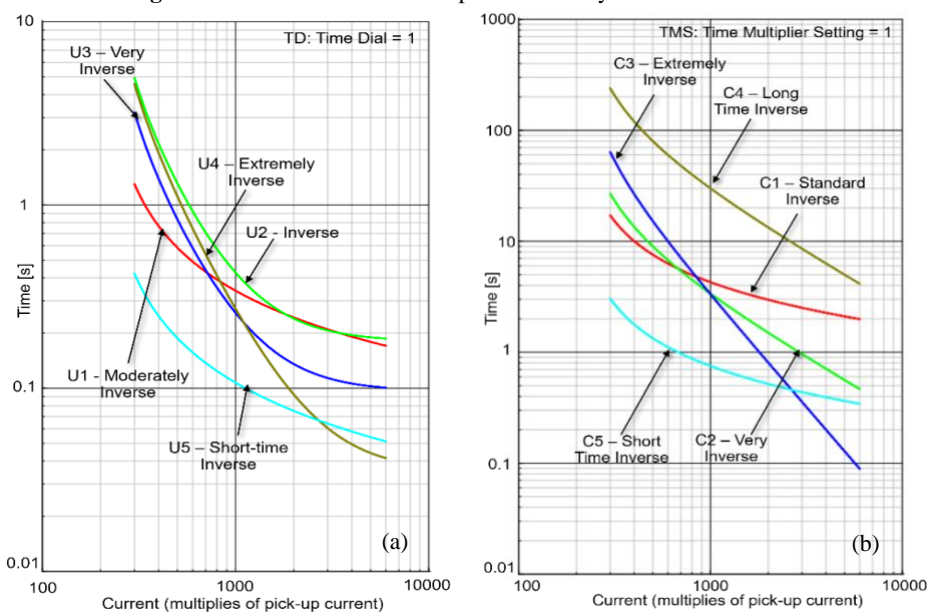
The American standards ANSI/IEEE use numerical coding to refer to protection functionalities. The ANSI device number helps identify the features of a protective device according to “*ANSI /IEEE Standard C37. 2 Standard for Electrical Power System Device Function Numbers, Acronyms, and Contact Designations*” [19].

When referring to overcurrent protection functions, the ANSI device number is 50 for an *instantaneous overcurrent* and *definite-time overcurrent*. In contrast, the *inverse-time overcurrent function* is denoted by the ANSI device number 51. The ANSI/IEEE C37.2 allows the use of suffixes. For instance, the suffix TD should be used (e.g., 50TD) to describe a definite time overcurrent function.

On the other hand, the European IEC standard defines the protection functions by using logical nodes according to “*International Standard IEC 61850-7-4 Communication networks and systems in substation*” [20]. As per the standard, PIOC is used for an instantaneous overcurrent and PTOC for an inverse-time overcurrent. The numerical suffixes that comes with PIOC and PTOC define protection elements. For instance, PIOC1 describes time overcurrent protection for ground overcurrent protection function.



**Fig. 3.** Instantaneous overcurrent protection relay characteristics.



**Fig. 4.** Inverse-time overcurrent protection relay characteristic. (a) *The American standard IEEE C37.112-1996*, (b) *The European IEC 60255*.

The operating characteristics of the protection overcurrent relay are used to offer protection to a particular situation by predetermined settings. The instantaneous overcurrent element has only one setting, the pickup current. On the other hand, the

inverse-time overcurrent element has two basic settings: the pick current and the time delay settings. The process of determining the time delay setting involves:

- (1) Calculation of a time-dial or time multiplier setting in definite-time overcurrent elements.
- (2) Selection in inverse-time overcurrent elements of a time-current curve from a family of curves is given in **Table 5** and **Table 6**.

These settings must be evaluated for reliability and correct operation before installing and deploying protective relays in a real-world system. The need for protection relay testing can be divided into four categories:

- **Type testing.** It is an extensive process where the quality of a newly fabricated relay or new software revision for a relay model is concerned.
- **Acceptance testing.** The protective relay is tested to prove that it is the correct model and that all the features work as they should. It consists of functional tests of inputs, outputs, displays, communication, and in some cases, predefined pickup and timing tests,
- **Commissioning testing.** It is a site-specific test to confirm all protective elements and logic settings are correct for their intended uses.
- **Maintenance testing.** It is used to ensure a protective relay continues to operate as it should.

At the same time, the protective relay testing methods and equipment become more powerful. However, they require time, effort and expensive physical equipment to conduct tests on the protective relay.

The authors in this chapter contributed an adequate protection relay testing method that overcomes the limitations of the previous testing methods by using real-time simulation. It allows the testing model to be executed at the same rate as actual physical devices making these tests more realistic under multiple testing scenarios.

For this purpose, the testing model is created that can be used for real-time simulation by using model-based engineering toolchains of Typhoon HIL. Two overcurrent protective functions are focused on, *instantaneous overcurrent* and *inverse-time overcurrent*. The implementation methodology of creating the model and conducting non-directional overcurrent protection relay testing are described in the following sections.

### 3 Workflow in the VHIL

The authors used the modelling and simulation framework for digital real-time simulations created by Typhoon HIL.

Typhoon HIL is the technology and market leader in Ultra-high-fidelity hardware-in-the-loop solutions for the design, testing, and validation of power electronics, microgrids, e-Mobility, EV powertrain software testing, and distribution control and protection systems. Typhoon HIL platform ushered a new era of model-based testing and validation of control software and hardware with its embedded, ultra-high fidelity real-time *Hardware-in-the-Loop* platform.

Virtual HIL or VHIL (as referred to in this chapter) is not an offline simulator. The VHIL device is a virtual machine created to emulate the computer system dedicated to real-time simulation; it brings the functionality of a physical HIL device with all “restrictions” of the real-time environment to a traditional PC.

In this chapter, the authors took advantage of the VHIL to implement and test the



proposed *Test System*. The proposed *Test System* is transformed into a model in the VHIL framework that can then be used to emulate the performance of the physical device.

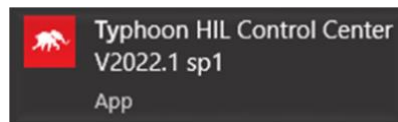
VHIL offers that opportunity by basically using two applications of *Typhoon HIL Control Centre*:

- (i) Schematic Editor, and
- (ii) HIL SCADA.

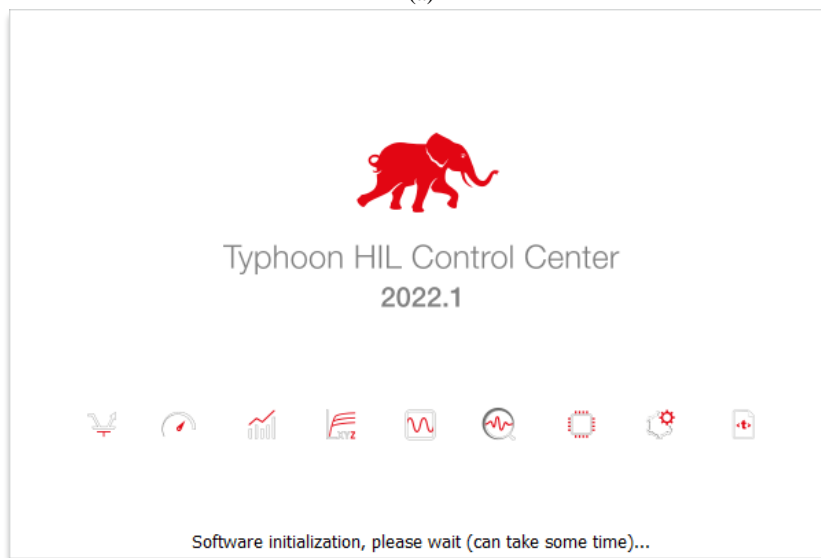
To access these toolchains, Typhoon HIL Control Centre is started up.

In this book chapter, the authors used the Typhoon HIL Control Centre version 2022 SP1, as shown in **Fig. 5a**. The software initialisation window will appear immediately after double-clicking on the application icon (see **Fig. 5b**).

The initialisation process of the Typhoon HIL Control Centre can take a few seconds depending on the user's PC specification; then, the Typhoon HIL Control Centre envisagement will appear in **Fig. 6** and be ready for the users to proceed with the subsequent progress.

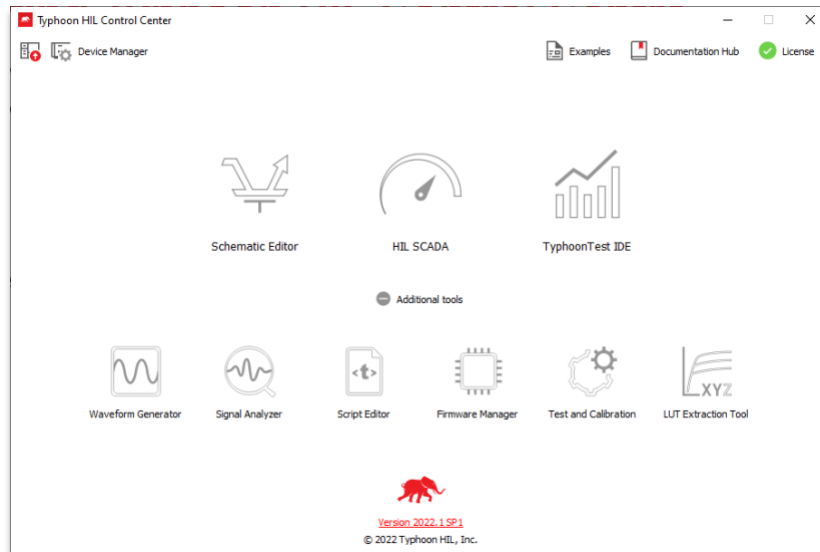


(a)



(b)

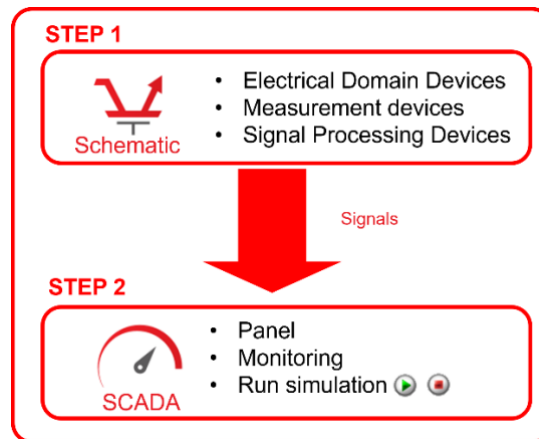
**Fig. 5.** (a) Desktop icon of Typhoon HIL Control Centre v2022.1 SP1, (b) Typhoon HIL Control Centre in software initialisation.



**Fig. 6.** Typhoon HIL Control Centre interface.

The workflow for using the Typhoon HIL modelling and simulation framework is straightforward (of course, more steps can be added depending on the complexity of the modelling and simulation framework).

The basic implementation of workflow in the VHIL can be shown in **Fig. 7**.



**Fig. 7.** Typhoon HIL Control Centre interface.

Considering the framework presented in **Fig. 7**, the steps are briefly explained below and then expanded during the following subsections:

**STEP 1:** The first step is to model the simplified benchmark network and protective relay using electrical domain devices in the available library in the Typhoon HIL Schematic Editor environment. The measurement devices and signal processing devices are distributed to the model to measure electrical signals in real-time simulation and send the signal to the monitor and control platform.

**STEP 2:** The second step is to create a control and monitor panel that displays the measured signals according to the multiple scenarios and different protection relay settings using HIL SCADA of Typhoon HIL.

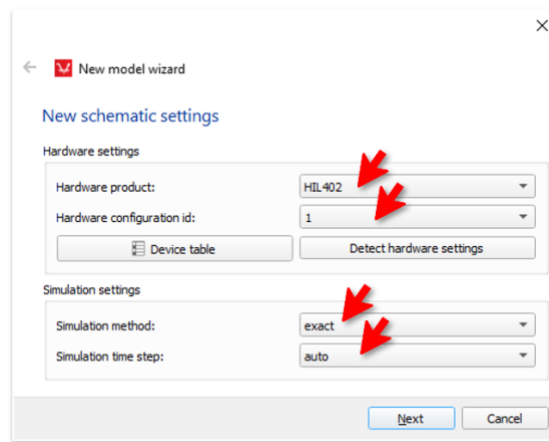
### 3.1 Schematic Editor Workflow

To start working in *Schematic Editor Environment*, there are two alternatives for creating a new schematic. The users can use the existing access toolbar to create a new

model or use the combination key “Ctrl +N” on the PC keyboard. After that, the setting window will appear and require users to define the schematic setting (see **Fig. 8**).

After finishing the setup of schematic settings, the Schematic Editor will load the *Library of Components*, and a new schematic design will be ready to start the building of the model.

To find components, the users can navigate the library tree or use the library explorer until having suitable components for the model. The model can be placed by classically dragging and dropping, and to rotate a component; the users can take advantage of the right bottom context menu when right-clicking on the component.

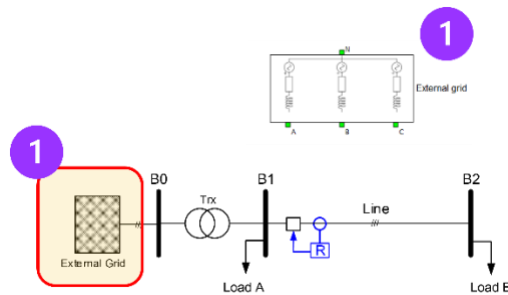


**Fig. 8.** Window showing the setting used for a New Schematic.

### 3.1.1 Defining the components in the test system

The simplified benchmark network used for the *Test System* circuit is created in the Schematic Editor. According to **Fig. 1**, the components of the *Test System* include an external grid, a two-winding power transformer (“Trx”), a non-directional overcurrent protection relay (R), three buses (B0, B1 and B2), a section of transmission line (Line 12), and three lumped three-phase loads (A and B). Additionally, short-circuit faults occurring at the end of the transmission line are simulated for the purpose of testing the non-directional overcurrent protection relay. These components are defined as follows:

- External grid (①): In branch “*Test Suite*” of the library tree, the component named “*Grid Simulator*” is used for modelling the external grid component of the test system.



**Fig. 9.** External Grid component.

- Transformer (②): For the transformer component, the “*Three-phase Two Winding Transformer*” in the “*Transformer*” branch of the library is selected.

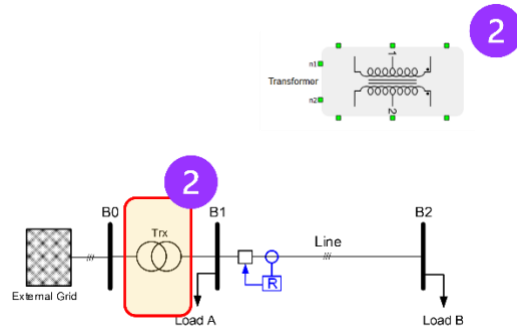


Fig. 10. Transformer “Trx” component.

- Line 12 (3): For the transmission line “Line12” component, “Transmission Line” in “PI section” part of the “Transmission Lines” of the library is selected.

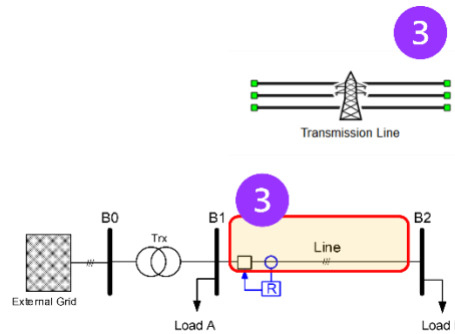


Fig. 11. “Line 12” component.

- Loads (4): Loads A and B are modelled as a constant impedance and it is implemented by using the “Constant Impedance Load” in section “Loads” of the “Microgrid” branch.

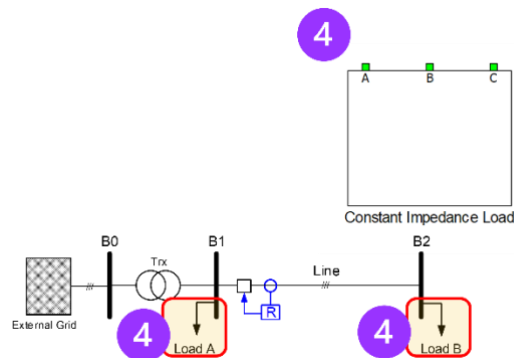
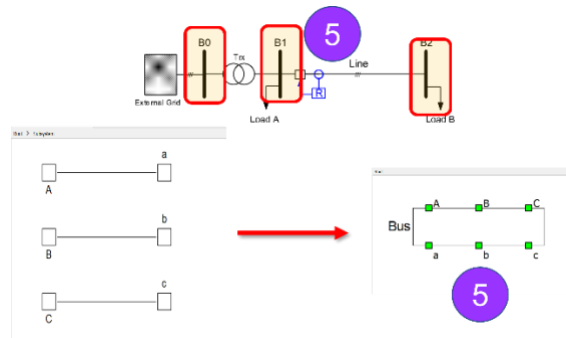


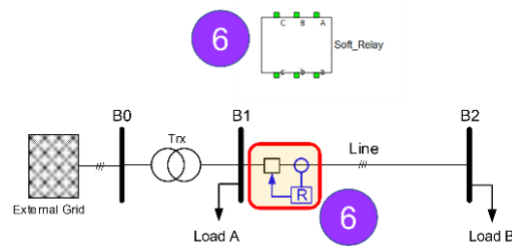
Fig. 12. Loads component.

- Buses (5): For bus B0, B1 and B2 components, users can use “Electrical port” in “Ports” branch and define a bus with three-port in and three-port out, then merge them under “Subsystem”.



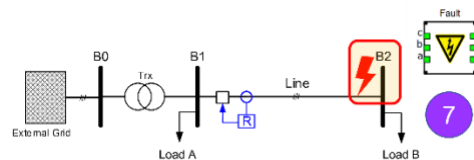
**Fig. 13.** Buses component.

- Overcurrent protection relay (6): The overcurrent protection relay device is different from the aforementioned components. It cannot be represented by individual components in the library but can be emulated by a combination of protective relay functions, current transformer and measurement devices. This component is defined in the later section (see **Section 3.1.3**).

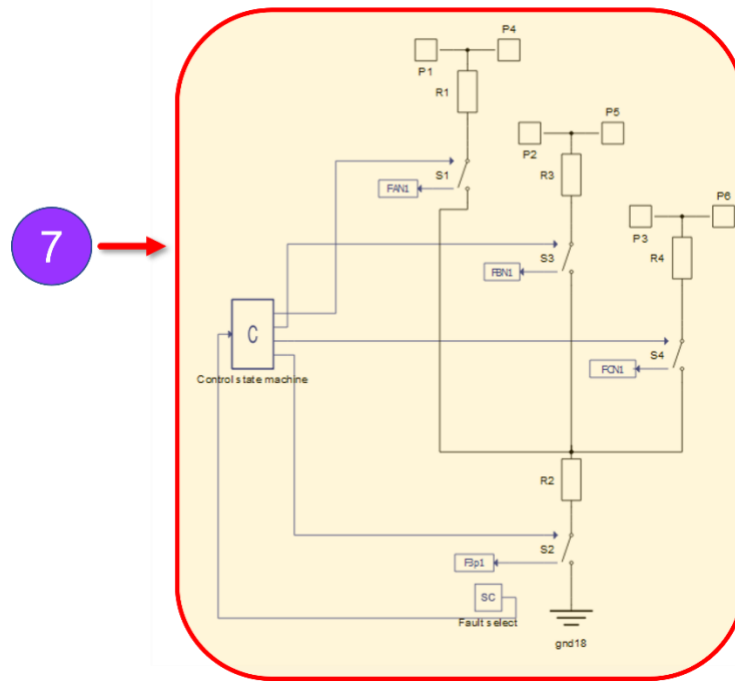


**Fig. 14.** Complete overcurrent protection relay component (R) named “Soft Relay”.

- Fault (7): This element is created to simulate a fault caused by short circuit short-circuit fault scenarios under overcurrent protection relay tests. It can be modelled under “Subsystem” of “Electrical port”, “Switch”, and “C function” of the library. The authors used switching cases of four “Switch”, S1, S2, S3 and S4, corresponding to three phases A, B, C and grounded neutral N, to simulate nominal operation and short-circuit fault scenarios (three-phase, three-phase-to-ground, single-phase-to-ground, phase-to-phase, and two-phase-to-ground) through programming scripts inside “C function” block named “Control State Machine”. These switches are labelled as "FANI", "FBNI", "FCNI", "F3PI" for S1, S2, S3 and S4, respectively. In case one of these switches is closed, a short circuit will occur. For example, four switches, S1, S2, S3 and S4, are closed for the three-phase-to-ground short-circuit case.



(a)



(b)

Fig. 15. (a) Fault element, (b) Nominal operating and short-circuit fault scenarios modelling.

### 3.1.2 Connecting the component and editing component data

After placing all components in the schematic environment, the components need to be connected. The users will notice that the line connecting the initial terminal is attached to the cursor and then move to the next terminal; then, perform a right-click at the ending terminal to finish the connection. The complete *Test System* model with the full connection of components can be shown in Fig. 16.

To edit the parameters of components, the users use double-click to open a properties window to include the appropriate data. For the external grid (1) and the “Trx” component (2), the user can edit the parameters according to Table 1 and Table 2, respectively. To edit the transmission line (3) and loads (4) component, the “*Model initialisation function*” is used to define these parameters of components by adding them into the namespace in the process of compilation. The users can start the “*Model initialisation function*” by clicking on the icon shown in Fig. 17 on the Schematic Editor of Typhoon HIL toolbar. Then, the command panel will appear in a few seconds, depending on the users’ PC and be ready for the users to perform parameters definition.

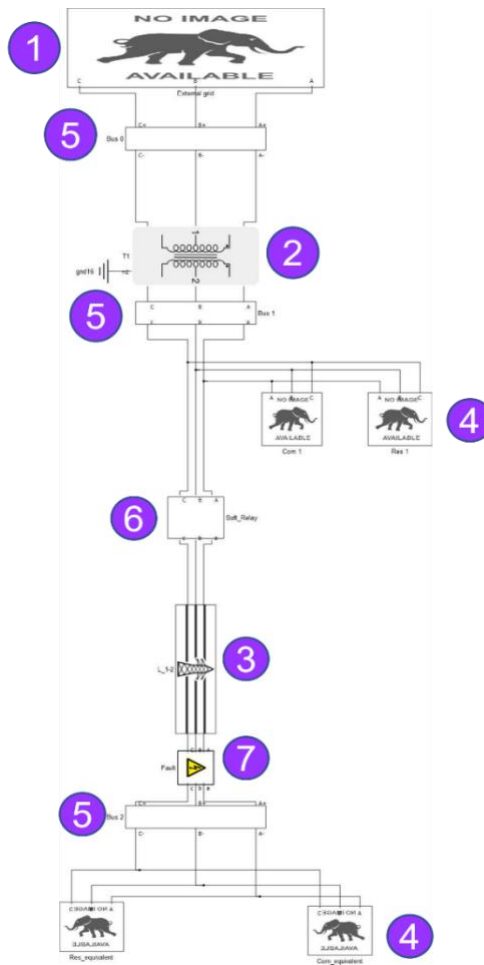
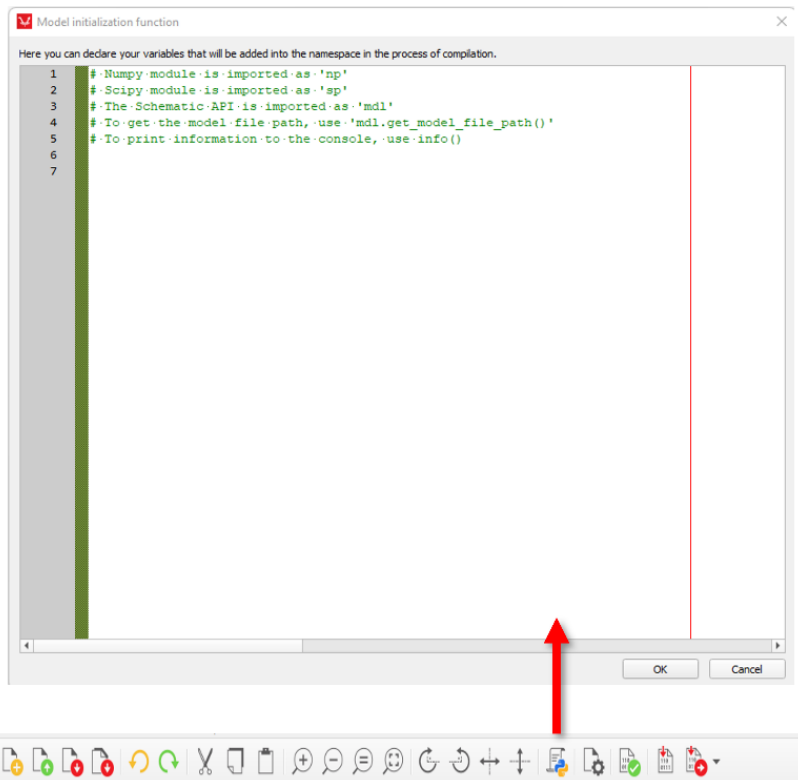


Fig. 16. Complete Test System model with the full connection of components.



**Fig. 17.** Setting parameter using “*Model initialisation function*”.

In the “*Model initialisation function*” environment, the programming language used is Python [21]. The users need to apply Python scripts to define the desired parameters for the *Test System*.

For transmission line, the authors used “*class*”, a code is used to construct zero and positive sequence impedance per unit length according to **Table 3**, under  $3 \times 3$  matrix form. As a consequence, the parameters set through the properties window will be filled in according to the nameplates of the users-set parameters. For instance, as shown in **Fig. 18**, “*class cable*” consists of common parameters of a typical cable: zero sequence and positive sequence of resistance, reactance and capacitance; variable “*LT*” is declared as “*class cable*”; the set data for resistance parameter of the transmission line is “*LT.Rabc\_matrix*” meaning resistance matrix in “*class cable*”.

Similarly, for load components (Load A and B), the set data for nominal three-phase power and power factor is under matrix form, as shown in **Fig. 19**.

The buses (5), non-directional overcurrent protective relay (6) and fault (7) are only connected and not necessary to be edited.

Transmission Line from library 'core'

Transmission line model implemented as either a PI section or RL coupled section.  
There are three ways to define the model parameters:  
- Geometry: x and y coordinates of the phase cables  
- RLC: per length resistance, inductance and capacitance values of cables in the phase domain  
- Sequence: per length resistance, inductance and capacitance values of cables in the sequence domain. Model parameters can be defined this way only for three phase transmission lines

Model: RL coupled  
Number of phases: 3  
Model definition: RLC  
Unit system: metric  
Length: length[0] km  
R: LT.Rabc\_matrix Ω/km  
L: LT.Labc\_matrix H/km  
Import from Geometry  
Help OK Cancel

```
class cable():
    def __init__(self, R0=0.817, R1=0.501, X0=1.598, X1=0.716,
                C0=0.1511749, C1=0.1511749, fn=50, unit='km'):
        self.R0 = R0 # ohms per self.unit [zero sequence resistance]
        self.R1 = R1 # ohms per self.unit [positive sequence resistance]
        self.X0 = X0 # ohms per self.unit [zero sequence reactance]
        self.X1 = X1 # ohms per self.unit [positive sequence reactance]
        self.C0 = C0 # nF per self.unit [zero sequence capacitance]
        self.C1 = C1 # nF per self.unit [positive sequence capacitance]
        self.fn = fn # Hz [nominal grid frequency]
        self.unit = unit # [distance unit used]
        self.w = 2*np.pi*self.fn # rad/s [angular speed]

        self.Rseq_matrix = np.matrix([[self.R0, 0, 0],
                                     [0, self.R1, 0],
                                     [0, 0, self.R1]]) # ohms/unit [sequence resistance matrix]
        self.Xseq_matrix = np.matrix([[self.X0, 0, 0],
                                     [0, self.X1, 0],
                                     [0, 0, self.X1]]) # ohms/unit [sequence reactance matrix]
        self.Zseq_matrix = self.Rseq_matrix + 1j*self.Xseq_matrix # ohms/unit [sequence impedance matrix]
        self.Lseq_matrix = self.Xseq_matrix/self.w # H/unit [sequence inductance matrix]
        self.Cseq_matrix = np.matrix([[self.C0, 0, 0],
                                     [0, self.C1, 0],
                                     [0, 0, self.C1]]) # nF/unit [sequence capacitance matrix]

        alpha, alphas = np.cos(-120.0*np.pi/180.0), np.sin(-120.0*np.pi/180.0)
        alpha = complex(alpha, alphas)
        A = np.matrix([[1, 1, 1],
                      [1, alpha**2.0, alpha],
                      [1, alpha, alpha**2.0]]) # sequence to phase components transf. matrix

        self.Zabc_matrix = np.dot(np.dot(A, self.Zseq_matrix), A.I) # ohms/unit [phase impedance matrix]
        self.Rabc_matrix = self.Zabc_matrix.real # ohms/unit [phase resistance matrix]
        self.Xabc_matrix = self.Zabc_matrix.imag # ohms/unit [phase reactance matrix]
        self.Labc_matrix = self.Xabc_matrix/self.w # H/unit [phase inductance matrix]
        self.Cabc_matrix = np.dot(np.dot(A, 1j*self.Cseq_matrix), A.I).imag # nF/unit [phase capacitance matrix]

        length = [2.82]
        LT = cable(R0=R0, X0=X0, R1=R1, X1=X1, C0=C0, C1=C1, fn=f, unit='km')
```

**Fig. 18.** Edit “Transmission line” parameters using “*Model initialisation function*”.



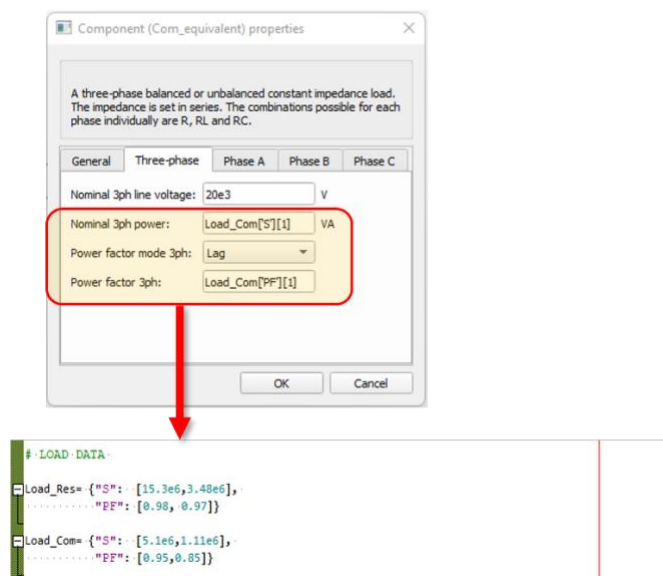
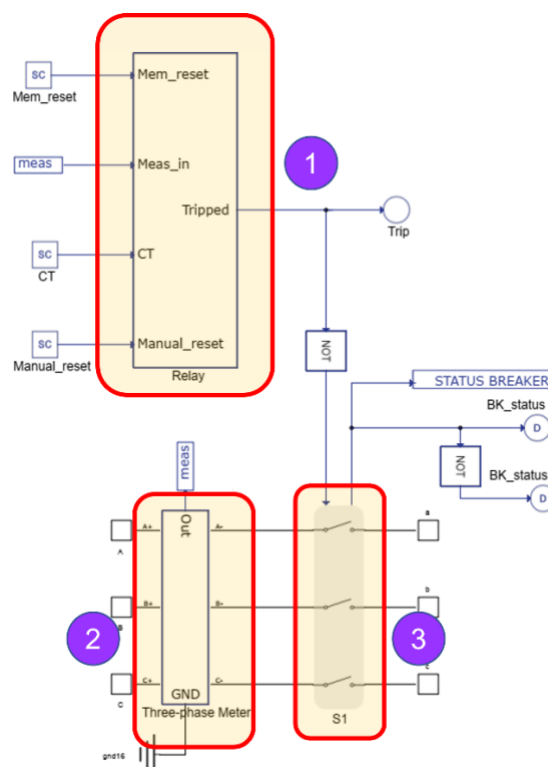
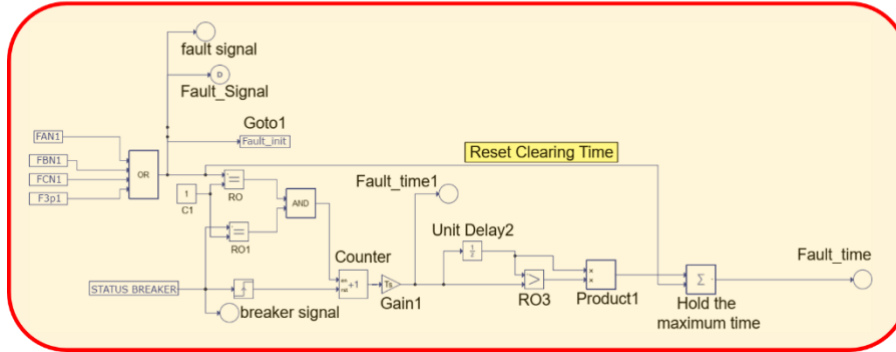


Fig. 19. Edit "Loads" parameters using "Model initialisation function".

### 3.1.3 Modelling non-directional overcurrent protection relay

The non-directional overcurrent protection relay component is named "Soft Relay" for testing purposes. It consists of four main elements as enumerated and shown in Fig. 20. ① represents the implementation of the virtual or soft-relay, in this case, it is the combination of available non-directional overcurrent protection models as included in the Typhoon HIL Schematic library. ② is a measurement device responsible for measuring electrical quantities such as voltage and current signals. ③ is the circuit breaker, and finally, ④ is the connected blocks for outputting the operating time of the protective relay.

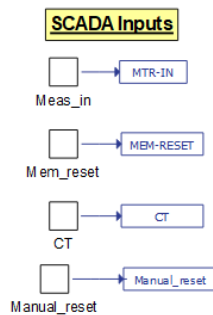




**Fig. 20.** Schematic diagram of the complete overcurrent protection scheme with four main elements, ❶ for overcurrent functions, ❷ for measurement device, ❸ for circuit breaker, ❹ for defining operating time of protection relay.

The first element (❶) named “Relay” is created based on available functionalities of relay SEL-751 included in the Typhoon HIL Schematic library, including the ANSI 50 and ANSI 51 elements. The inputs of this element are “Mem\_reset”, “Meas\_in”, “CT” and “Manual\_reset”.

These input signals are used by the subsystems inside the “Relay” as shown in **Fig. 21**. The input “Mem\_reset” aims to send of the protective functional blocks inside “Relay” which will reset the memory of the protective functional blocks. The input “Meas\_in” is used to receive the measured RMS current from ❷, the input “CT” is intended to receive the ratio of current transformer, and the input “Manual\_reset” is used to receive the trip signal reset of the “Relay”. These input values can be configured in the HIL SCADA interface.

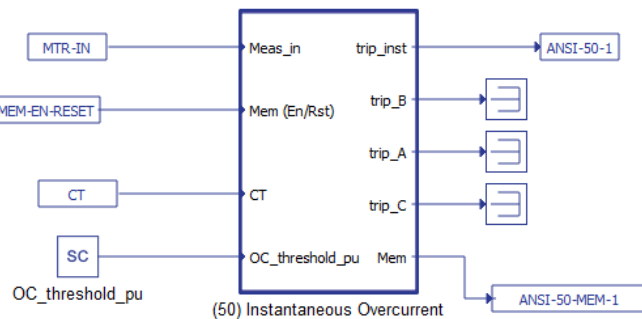


**Fig. 21.** HIL SCADA inputs to “Relay” with their nameplates.

**Fig. 31** shows the subsystems inside the “Relay” that are “Protective functions”, “Trip Source combination”, “Trip Memory reset zone”, and “Trip Signal zone”. Each of these subsystems are explained as follows:

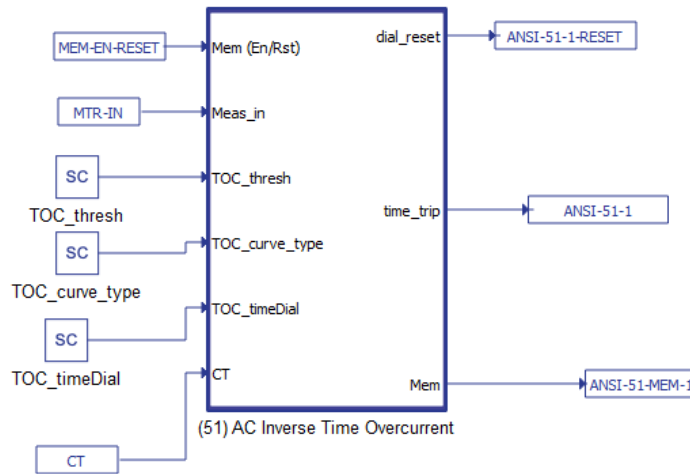
- *Protective functions*: In this subsystem, there are two main overcurrent protective functional blocks with the specific description and parameters settings that are available in [22], including:
  - “(50) Instantaneous Overcurrent” (see **Fig. 22**). This block is dedicated to providing instantaneous protection against high currents. This block issues a trip signal without any time delay. The inputs of this block are “Meas\_in”, “Mem (En/Rst)”, “CT”, and “OC\_threshold\_pu”. Two inputs, “Meas\_in” and “CT”, are connected with the “Relay” inputs through previously declared nameplates, the input “Mem (En/Rst)” is intended to reset the trip

memory, and the input “*OC\_threshold\_pu*” is used to receive the threshold value from HIL SCADA interface. The output signals are “*trip\_inst*”, “*trip\_A*”, “*trip\_B*”, “*trip\_C*” and “*Mem*”. The outputs “*trip\_A*” is used to move the protective relay dials toward trip position if the measured current crosses the pre-set threshold value on phase A. Same purpose goes for “*trip\_B*” and “*trip\_C*” on phase B and phase C, respectively. However, these outputs are not used; instead, using the output “*trip\_inst*” sends the trip signals when any currents at any of the phases cross the defined threshold. The output “*Mem*” is used to send the feedback of trip memory; when the tripping occurs, this output is active; otherwise, this output is inactive.



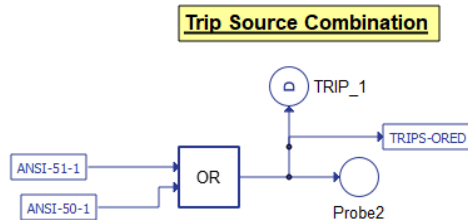
**Fig. 22.** “(50) Instantaneous Overcurrent”.

- “(51) AC Inverse Time Overcurrent” (see **Fig. 23**). This block is dedicated to protecting against high currents based on time dial and inverse-time overcurrent curves. There are six inputs of this block including “*Mem (En/Rst)*”, “*Meas\_in*”, “*TOC\_thresh*”, “*TOC\_curve\_type*”, “*TOC\_timeDial*”, and “*CT*”. Two inputs, “*Meas\_in*” and “*CT*”, are connected with the “*Relay*” inputs through declared previously nameplates, the input “*Mem (En/Rst)*” is used to reset the trip memory; the input “*TOC\_thresh*” is used to receive the threshold value from HIL SCADA interface; the input “*TOC\_curve\_type*” is used to define the inverse-time overcurrent curves corresponding to a family of curves in **Table 5** and **Table 6**, and the input “*TOC\_timeDial*” is used to receive the TD or TMS values from HIL SCADA. The output signals are “*dial\_reset*”, “*time\_trip*”, and “*Mem*”. The output “*dial\_reset*” is intended for feedback on the dial states, the output “*time\_trip*” is used for an inverse-time overcurrent trip command, and the output “*Mem*” to send the feedback of trip memory; when the tripping occurs, this output is active; otherwise, this output is inactive.



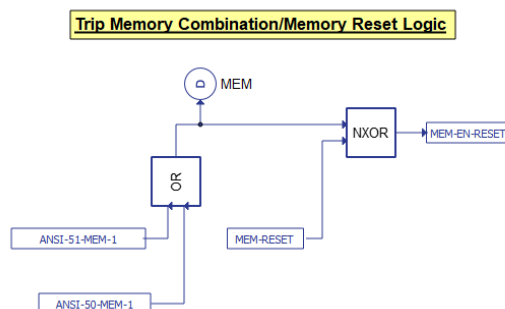
**Fig. 23.** “(51) Instantaneous Overcurrent”.

- “Trip Source Combination” (see **Fig. 24**): the “OR” block (out is True if any inputs are True) in this subsystem is dedicated to receiving the output signal “trip\_inst” from “(50) Instantaneous Overcurrent” and the output signal “time\_trip” from “(51) AC Inverse Time Overcurrent”. When any protective functions detect a fault, the trip is issued, which triggers a command to the output of the “OR” block that sends the trip signal to the HIL SCADA interface.



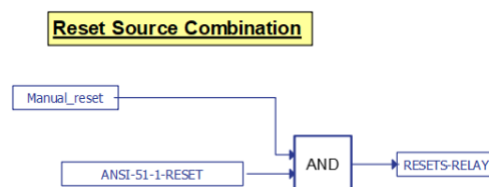
**Fig. 24.** “Trip Source Combination” zone.

- “Trip-Memory Reset Zone”: In this area, there is a combination of multiple functional blocks for memory, and trip reset of protection functional blocks as follows:
  - “Trip Memory Combination/Memory Reset Logic” (see **Fig. 25**). There are two logical blocks in this area, the “NXOR” block (out is True if all inputs are True or False) and the “OR” block. The “OR” block is dedicated to receiving the output “Mem” of two protective functional blocks “(50) Instantaneous Overcurrent” and “(51) AC Inverse Time Overcurrent”. The “NXOR” block has two inputs; one is the output of the “OR” block; and one is “Mem\_reset”, an input of the “Relay” that can be configured from the HIL SCADA interface. When one of two protection functional blocks detects a fault, the output signal of “OR” is active; and if the “Mem\_reset” signal is active contemporaneous via the HIL SCADA interface, the output of “NXOR” is activated and sends the memory reset signal to the input “Mem (En/Rst)” of two protection functional blocks.



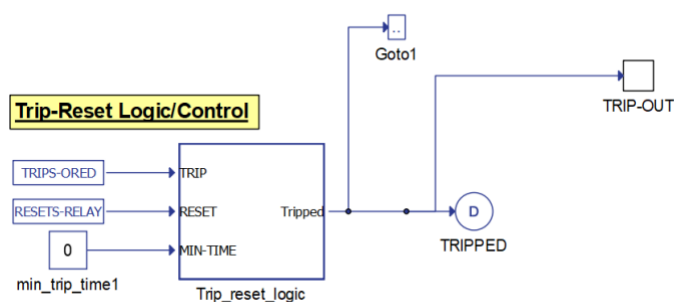
**Fig. 25.** “Trip Memory Combination/Memory Reset Logic” zone.

- “Reset Source Combination” (see **Fig. 26**). The “AND” block (out is True if all inputs are True) in this subsystem has two inputs, one is the output “dial\_reset” of “(51) AC Inverse Time Overcurrent” and one is “Manual\_reset”, a input of the “Relay” that can be configured from HIL SCADA interface. The output of “AND” block is “Resets-Relay” that is dedicated to reset the trip state of protective functional blocks.



**Fig. 26.** “Reset Source Combination” zone.

- “Trip-Reset Logic/Control” (see **Fig. 27**). This block is dedicated to issuing the trip signal to the HIL SCADA interface. The inputs of this block are “Trip”, “Reset”, and “Min-Time”. The input “Trip” is used to receive the trip signal from two protective functional blocks and is the output of “Trip Source Combination” the input “Reset” is used to receive the trip reset signal and is the output of “Reset Source Combination”, the input “Min-Time” is to define the time delay signal for needed backup protection (this input is set at 0 because no backup protection in this section). The output of this block is a trip signal; if the protective relay is active, this output is enabled, otherwise, in case the protective relay is not in the trip position, or there is a trip reset signal, this output is inactive.



**Fig. 27.** “Trip-Reset Logic/Control” zone.

- “Trip Signal zone” (see **Fig. 28**): this area consists of two “SR Flip Flop” blocks which implement the functionality of the SR Flip Flop sequential logic [23]. S input can be viewed as a “Set input” and R as a “Reset input”. If S is active, the “Flip Flop” will store the value 1. If R is active, the “Flip Flop”

will store the value 0. A particular case is when both S and R are active. This is an invalid case, and the state of the “Flip Flop” will be -1. This area is intended to identify the status of the “Relay” with two protective functional blocks, whether it is at the trip position or at reset state. For the inputs of these blocks, S input is “trip\_inst” of “(50) Instantaneous Overcurrent”, “time\_trip” of “(51) AC Inverse Time Overcurrent” and “Resets-Relay” of “Reset Source Combination”. The outputs of these “SR Flip Flop” blocks are named “TRIP 51\_1” and “TRIP 50\_1” to determine which functional block operates.

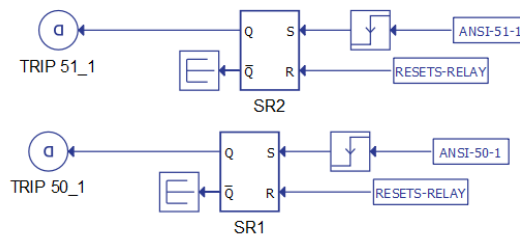


Fig. 28. “Trip Signal” zone.

The second element (2) is represented by the “Three-phase meter” in the Schematic Editor library to measure RMS values of current flowing through the protective relay. The output of this element is named “meas”, including RMS values of current that is the input of 1.

“Triple Pole Single Throw Contactor” is used to define the circuit breaker as the third element (3) that is connected with the “Three-phase meter”. This element has one input and one output. The input is used to receive the trip signal of the “Relay”, and the output is used to define the status of the circuit breaker in two states, open or closed. The output of 3 is nameplated as “Status Breaker”. The connection of 2 and 3 can be depicted in Fig. 29.

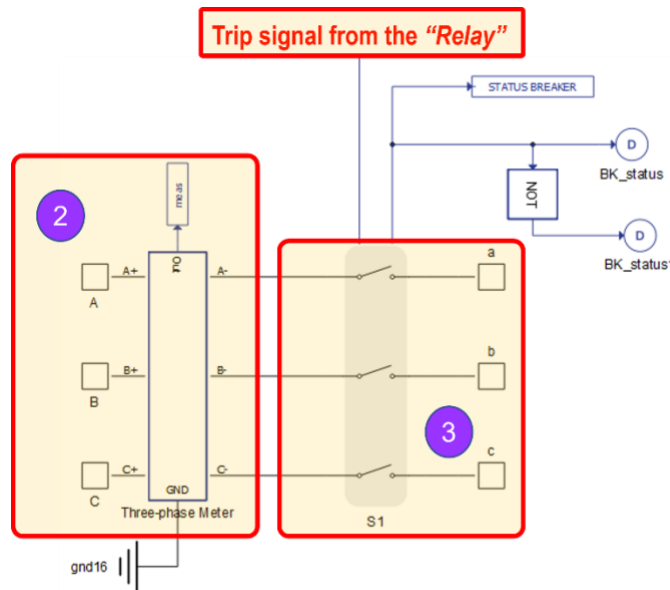
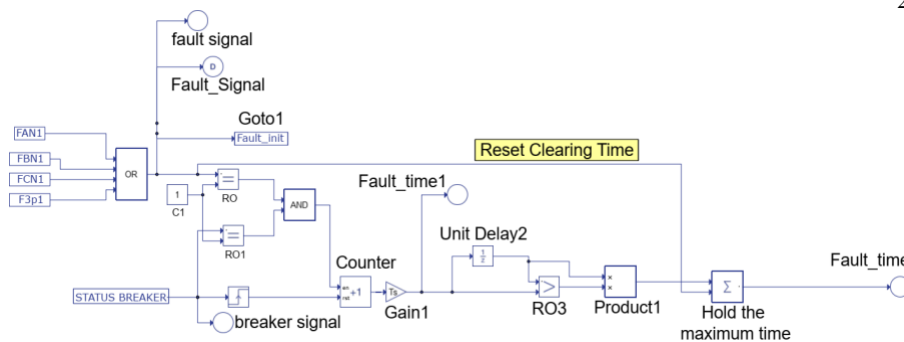


Fig. 29. “Three-phase meter” connected with “Triple Pole Single Throw Contactor”.

Finally, 4 is the combination of a series of logic and mathematical blocks that are available in the Schematic Editor library for calculating the protection relay operating time (see Fig. 30).



**Fig. 30.** Protection relay's operating time calculation zone.

The inputs are the short-circuit cases tagged with nameplates from the “*Fault*” element of the *Test System* (see **Section 3.1.1**) and the name “*Status Breaker*” from the third element “, *Triple Pole Single Throw Contactor*” of “*Soft\_Relay*”.

When a short-circuit fault occurs (tagged by nameplate “*Fault\_Signal*”) and the status of the circuit breaker (labelled by the name “*Status Breaker*”) is in the open state, the time counter is enabled. The counter is deactivated when the short-circuit fault is removed; the removal time depends on the output of two overcurrent protection functional blocks, “*trip\_inst*” of “(50) *Instantaneous Overcurrent*” and “*time\_trip*” of “(51) *AC Inverse Time Overcurrent*”. The protection relay's operating time is maintained when the counter stops working when short-circuit fault is cleared.

The algorithm of protection relay's operating time is provided to illustrate precisely the construction method of the associative blocks of 4.

Algorithm for the relay operating time calculations

1. Inputs:
  - Ts = Execution rate
  - SB = Status of Circuit Breaker
  - SBC = Circuit Breaker change position state
  - en = Encounter
  - t = Operating time of relay
  - SB close = 1; SB open = 0
  - Fault on = 1; Fault off = 0
  - Ts = 20e-06
2. if (Fault = 1) and (SB = 1) [Condition to enable the counter]
3.     en = 1;
4. else
5.     en = 0;
6. end if
7. if SB (t-1) = SB (t) [Identify state change of SB]
8.     SBC = 0;
9. else
10.     SBC = 1;
11. end if
12. count = 0;
13. if en = 1 and SBC = 1 [Condition to enable the counter]
14.     count = count + 1;
15.     t\_faultON = count \* Ts;
16. else
17.     count = 0;
18. end
19. Maintain the relay operating time
20. Max = 0;
21. if t\_faultON(t-1) > t\_faultON(t)

24

```
18.     c = 1;
19. else
20.     c = 0;
    end
21. Max =Max + t_faultON(t-1)*c
22. Outputs:
    The operating time of protection relay
```

The algorithm includes two sections: (a) Calculations of the protection relay's operating time and (b) maintaining the operating time of the protection relay.

In the first section of the algorithm, the inputs are execution rate, the status of the circuit breaker, state change of circuit breaker, state of fault and encounter. The discrete execution rate is set as default according to the sample time assigned system components of 20  $\mu$ s. This section includes three logical conditions with "If" statements. In the first logical condition, it is the condition to enable the encounter when there is a short-circuit fault, and the circuit breaker is still in the closed state. The short-circuit fault is happening, and the circuit breaker needs to switch to the open state; this is the second condition to identify the status of the circuit breaker. The third condition is to count the timer of operating time when the counter is active and the state of the circuit breaker is in an open state.

The second section of the algorithm includes one logical condition to stop the time counter from maintaining the operating time of the protection relay. If the short-circuit fault is cleared, the counter is deactivated, resulting in the timer stopping. The logical condition is the product of counter time and previous time 1.0 second if greater than the product of counter time (is forced to stop when the fault is cleared) and real-time, the counter time or the operating time of protective relay is held that will be displayed in HIL SCADA interface.



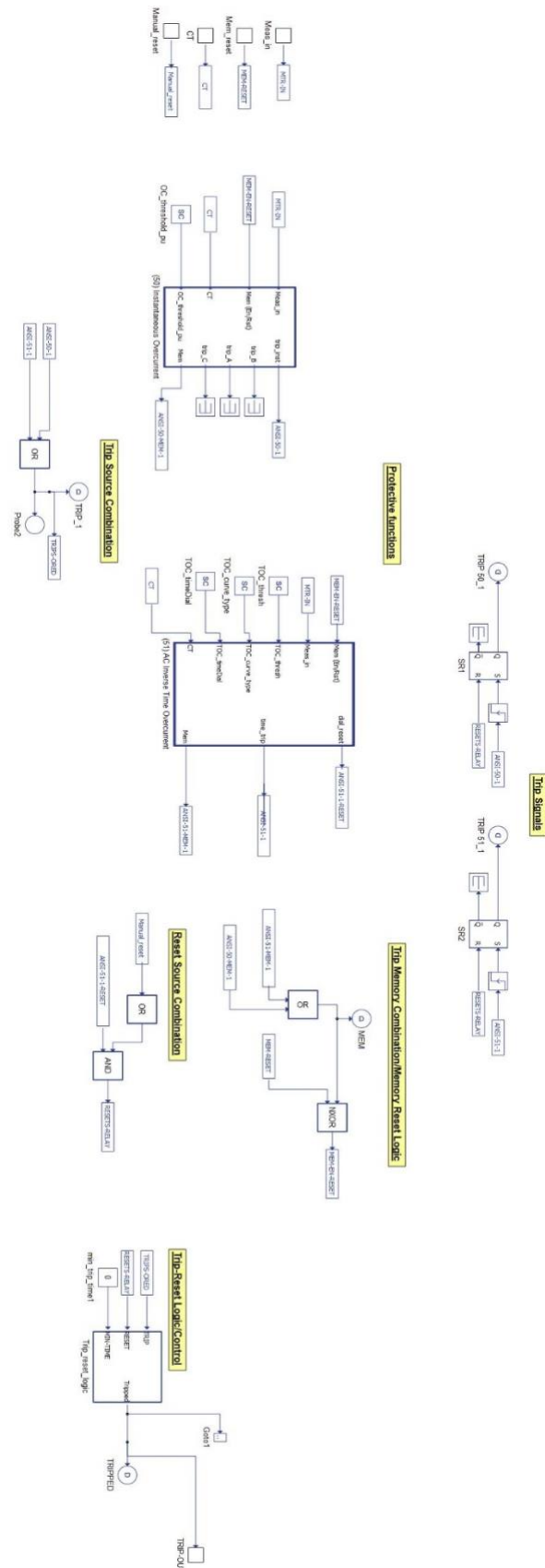


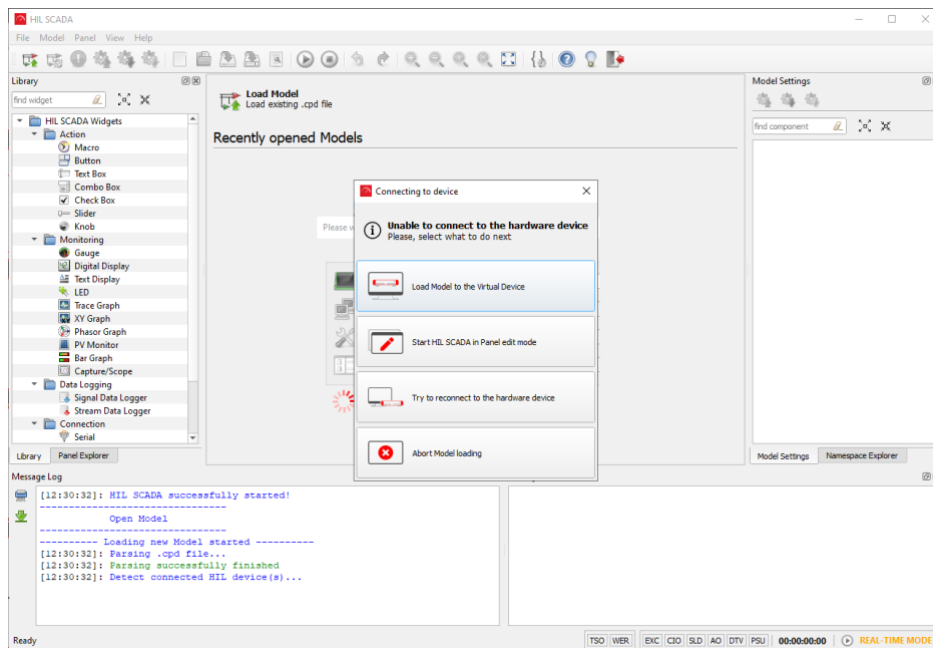
Fig. 31. The “Relay” elements.

### 3.2 SCADA HIL Workflow

The model of the *Test System* is built using the Typhoon HIL schematic, and then the

model is compiled to translate the computer model created in Schematic Editor into a programming language code that can be used in the over elements of the VHIL toolchain.

The users use two buttons in the toolbar to compile the model, “*Compile schematic*” (Compile the current schematic model) and “*Compile and load model in HIL SCADA*” (Compile the current schematic model and load it into HIL SCADA). The alternative way is using the Schematic Editor menu “*Model*” and selecting the option “*Compile schematic*” or “*Compile and load model in HIL SCADA*” as needed.



**Fig. 32.** Loading the created schematic model into the virtual HIL device.

The HIL SCADA interface can be depicted in **Fig. 32**. The users need to select the option “*Load Model in the Virtual Device*” to load created model in the schematic to HIL SCADA. This option emulated the real HIL 40x/60x device, but no external I/O to real devices is possible. Then, the HIL SCADA editor is ready to use, and the graphical environment is prepared to create a specific interface with the real-time model. To create a new panel, the users need to press the “*Create new Panel*” at the command toolbar. The blank panel appears and is ready for the users to place available widgets from the library, as shown in **Fig. 33**.

An efficient monitoring dashboard requires users to customise the widget for the panel depending on the arrangement and the quantities that users want to display and control. The panel includes two main functions: *results displaying* and *adjusting the desired values*.

In this chapter, the complete monitoring and controlling panel was created using HIL SCADA, and its implementation is depicted in **Fig. 34**.

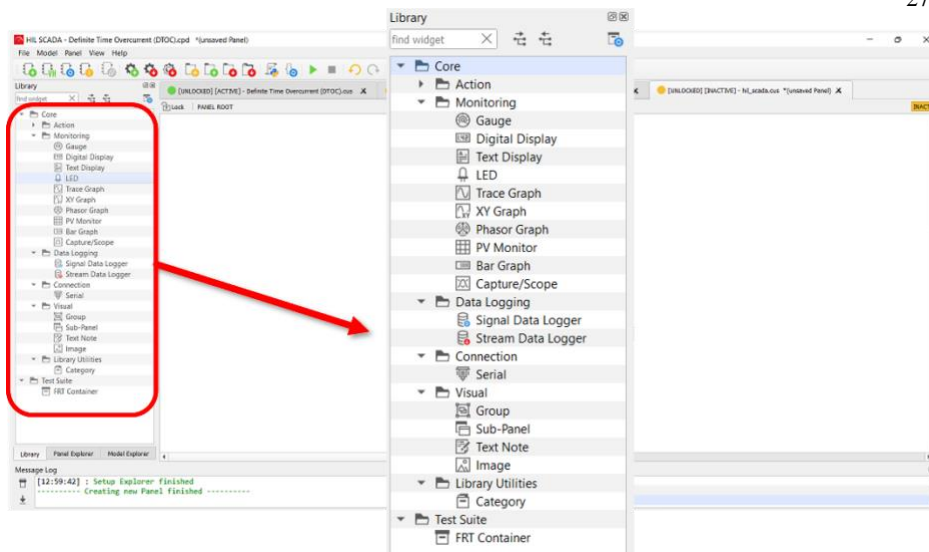


Fig. 33. Working panel in HIL SCADA with available widgets library.

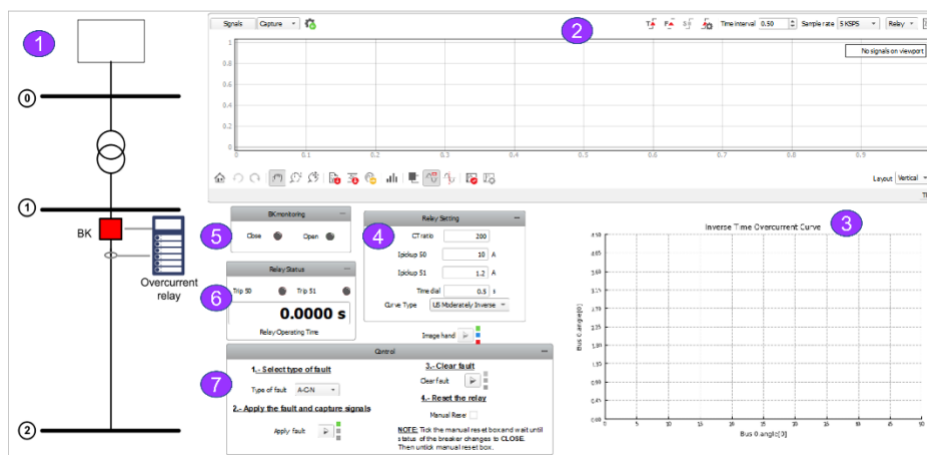


Fig. 34. Complete monitoring and controlling panel created by HIL SCADA, ① for the visual image, ② for capturing voltage and current waveforms, ③ for inverse-time overcurrent curves visual graph, ④ for protection relay settings dashboard, ⑤ for circuit breaker's status monitoring, ⑥ for protective relay's status monitoring, and ⑦ for control dashboard.

The widgets in the “Monitoring” and “Action” core of the widget library are intended to create the monitoring and controlling panel. These widgets are adjusted by performing a double-click on the widget to open the properties window. In the window settings, the users can do navigation commands that require widgets to perform requests of the users, including displaying desired values or signals for monitoring functions or adjusting parameters.

The first element (①) is used to visualise protective relay tests through images. To create this element, the users need to select the “Image” widget in the “Visual” of the HIL SCADA widget library; then, the users perform a right-click to define the direct path of the desired image (see Fig. 35).

The visual image used in the panel is *Test System* based on Fig. 1. Additionally, the authors used the image in SVG (Scalable Vector Graphics) format to reconstruct the image under two cases, nominal operating and short-circuit fault cases. For the nominal operating case, the circuit breaker (named BK in the image) is highlighted in red colour, which indicates the closed state. For short-circuit fault cases, the circuit breaker is in green colour, which shows the opened state, and the fault icon appears at the end of the transmission line (Line 1-2).

To perform the image format in the two aforementioned conditions, the authors used

the “Macro” widget in “Action” of the HIL SCADA widget library; then, the users need to use a right-click to open the properties window and start to parse SVG image as XML (the Extensible Markup Language). The “Fault Signal” and “Status Breaker” from the fourth element (④) of “Soft Relay “ (see Section 3.1.3) are the input data to execute “BK” and fault icon via “xml.etree.ElementTree” module [24] with associated commands (see Fig. 36).

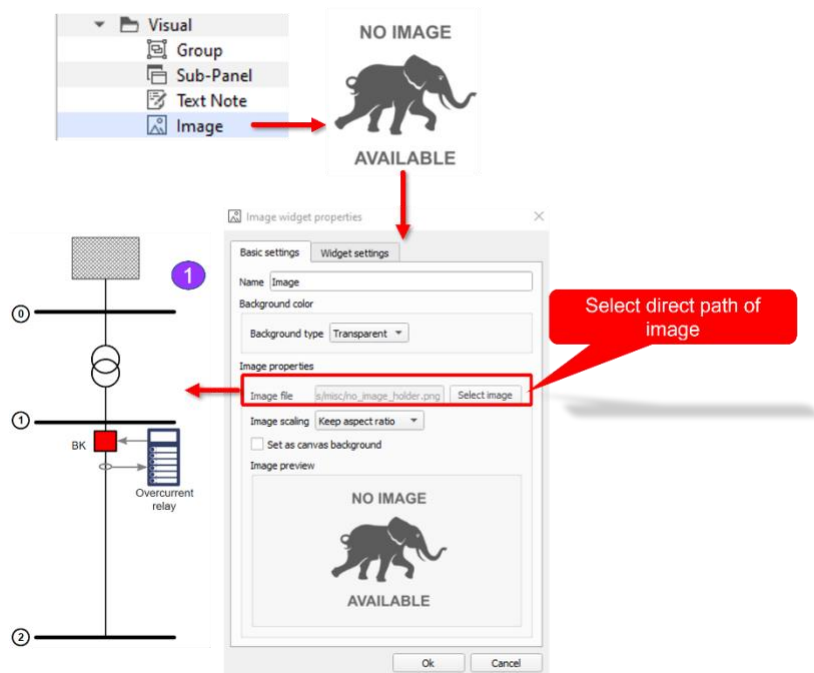


Fig. 35. The first element (①) “Visual Image”.

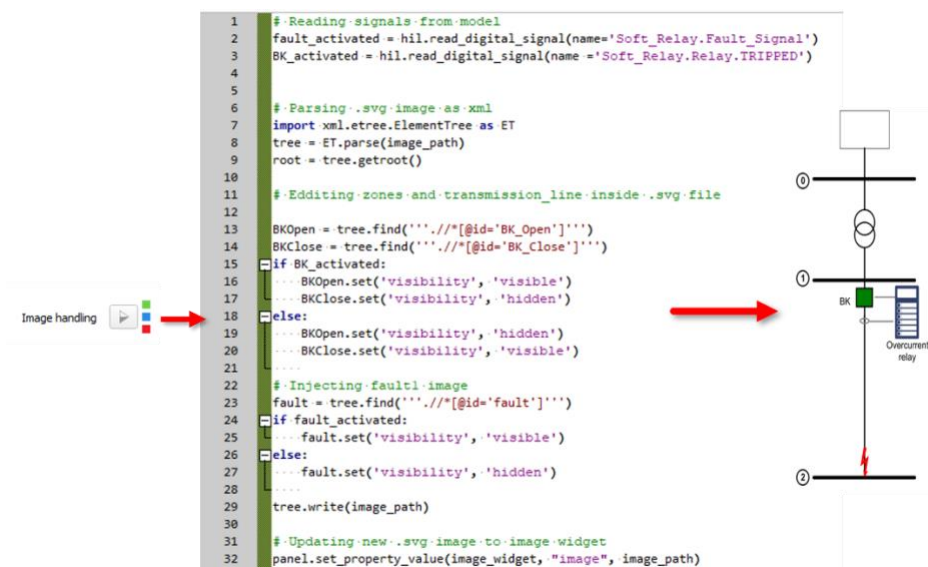
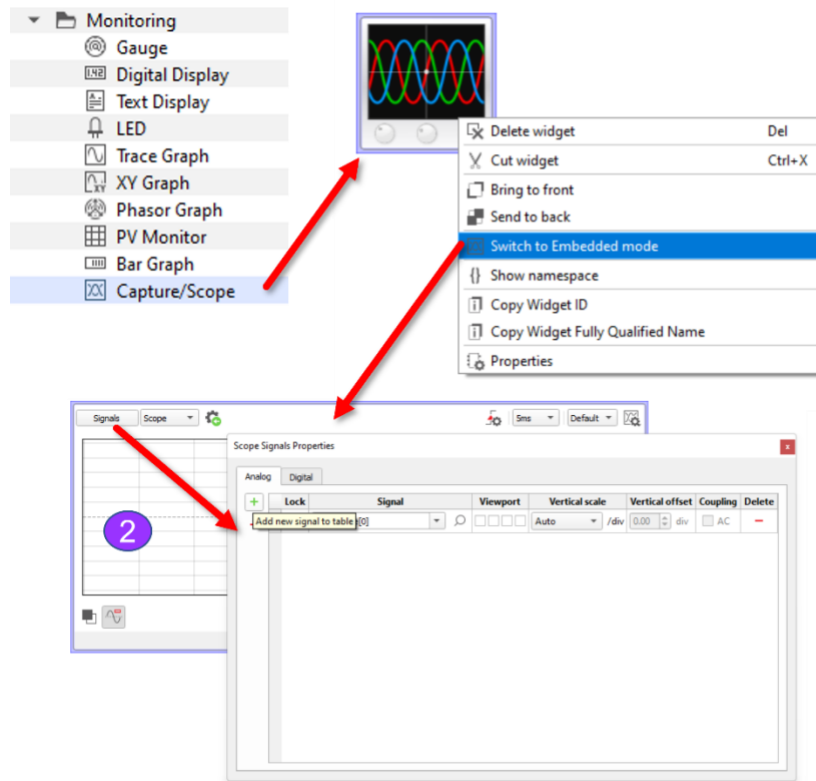


Fig. 36. Define “Visual Image” in specific circumstances, normal operation and fault cases.

The second element (②) (see Fig. 37) is “Capture/Scope” in “Monitoring” of HIL SCADA library. The users can right-click to select “Switch to embedded mode”, then select “Signal” to capture desired signals to observe during the real-time simulation.



**Fig. 37.** The second element (2) “Scope”.

The third element (3) (see **Fig. 38**) is created to visualise inverse-time overcurrent curves during the test. The users can select “XY graph” in “Monitoring”, then right-click to open the properties window. At this window, the authors have set the X-axis as multiples of pickup current and the Y-axis as the protective relay’s operating time based on the equation of the curves’ family. In this programming part, the authors used the input of the RMS values of the 3-phase current measured from the three-phase meter (see **Section 3.1.1**); CT and time dial values, along with the parameters of curve equations, are set in the fourth element (4), “Relay setting”.

The fourth element (4) (see **Fig. 39**) is the “Relay setting” dashboard, which is a group of widgets in “Action”, including “Text Box”, for importing the desired values of protection relay settings. They are CT ratio, ANSI 50/51 pickup current, time dial and inverse-time overcurrent curves. The users use the “Text Box” widget to define the values of CT ratio, ANSI 50/51 pickup current and time dial. The “Combo Box” widget is used to select inverse-time overcurrent curves. These settings can be defined by double-clicking on the widget and using the Python command in the “Macro code” section. They are the inputs of the overcurrent protection functional block created in Schematic Editor previously (see **Section 3.1.2**).

The fifth element (5) (see **Fig. 40**) is the “BK monitoring” dashboard, including the “LED” widget in “Monitoring”. The LED expression is defined inside the widget properties according to the state of BK of the *Test System*; the LED is red when BK is closed and green when BK is opened.

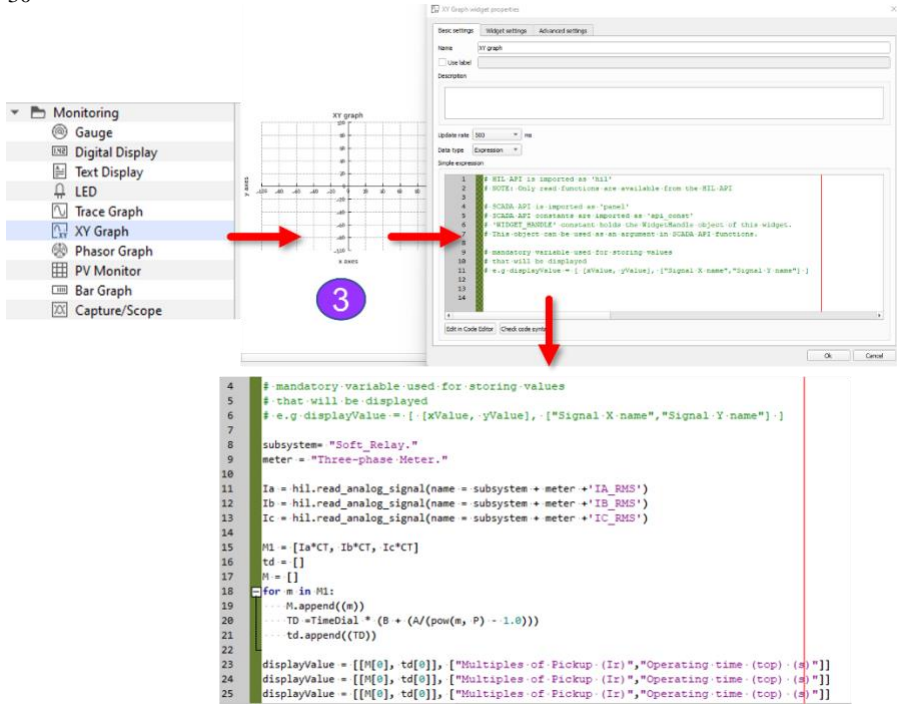


Fig. 38. The third element (3) "XY graph".

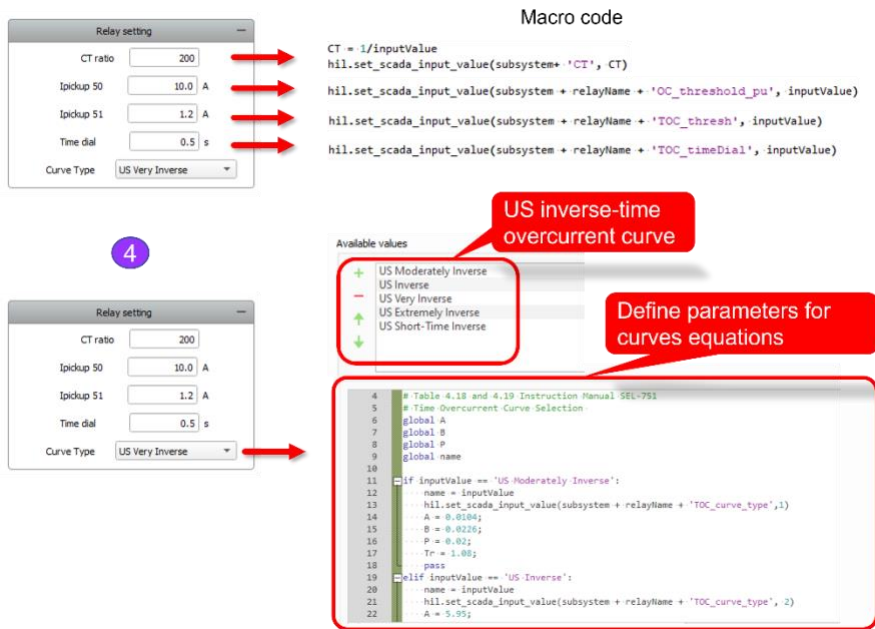


Fig. 39. The fourth element (4) "Relay setting" dashboard.

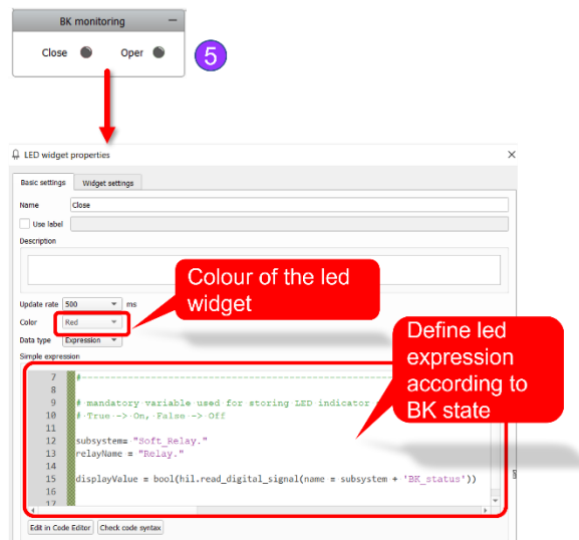


Fig. 40. The fifth element (5) "BK monitoring".

The sixth element (6) (see Fig. 41) is "Relay status", including "LED" and "Digital Display" widgets. The "LED" widget is used to represent the trip signal coming from which overcurrent functional blocks, "(50) Instantaneous Overcurrent" or "(51) AC Inverse Time Overcurrent". The "Digital Display" widget represents the operating time of the protective relay. The users can use right-click on the widget and select the desired signal.

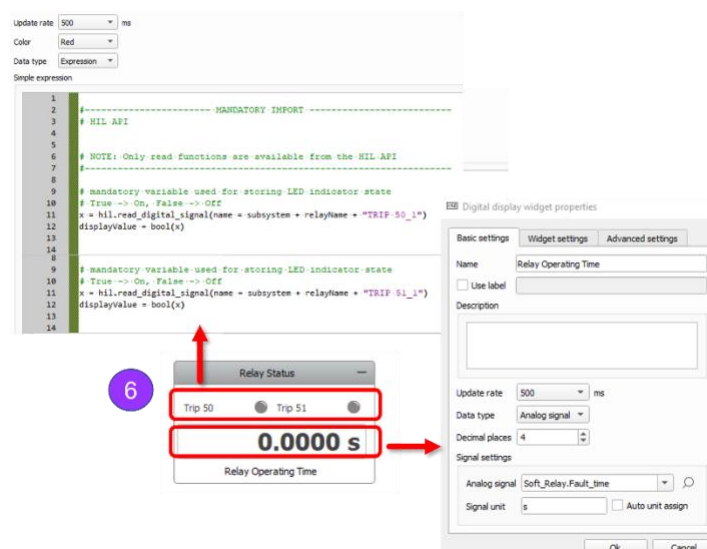


Fig. 41. The sixth element (6) "Relay status"

The seventh element (7) is the "Control" dashboard created to perform multiple actions, "Fault selection", "Apply Fault", "Clear Fault", and "Reset Relay" (see Fig. 42).

For the "Fault selection", the "Combo Box" widget in "Action" of the HIL SCADA library is used to perform this action. The users can define the name of short-circuit cases in "Available values" of widget properties, then, according to declared types of fault in the Fault element of the test system (see Section 3.1.1). The nominal operating



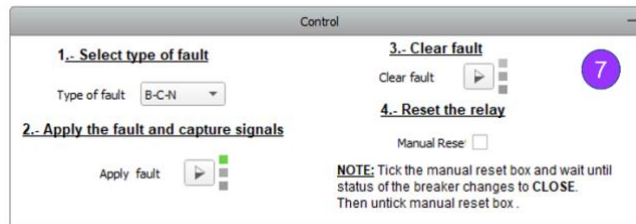
case is named “*NO FAULT*”. At the same time, short-circuit fault can be of different types: three-phase, three-phase-to-ground, single-phase-to-ground, phase-to-phase, and two-phase-to-ground. The short circuit is typically identified according to the *A*, *B*, *C* phases and the grounded neutral *N*, as shown in **Table 7**. The users can select one of these short circuits to conduct the appropriate tests.

**Table 7.** Nameplates corresponding with short-circuit fault cases.

Name	Short-circuit fault cases
A-N	Single-line-to-ground
B-N	
C-N	
A-B	Phase-to-phase
A-C	
B-C	
A-B-N	Two-phase-to-ground
A-C-N	
B-C-N	
A-B-C	Three-phase
A-B-C-N	Three-phase-to-ground

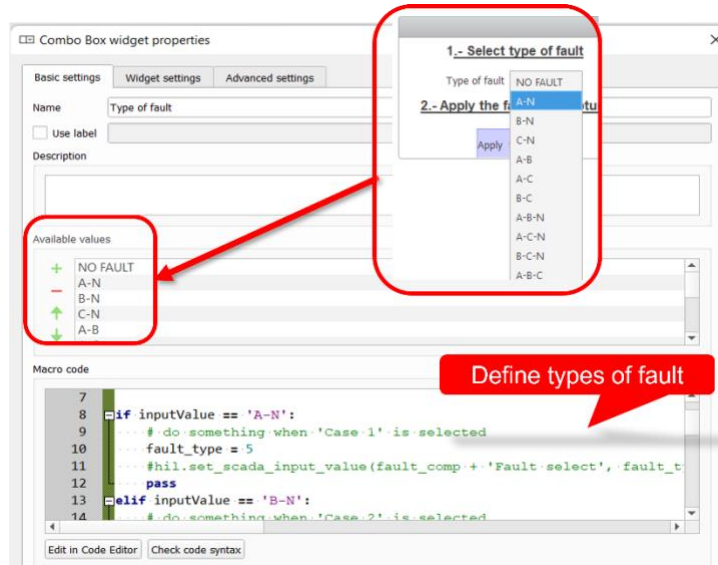
For the “*Apply Fault*”, the users select the “*Macro*” widget in “*Action*” of HIL SCADA library. This button defines the sample rate and time interval for captured signal in “*Capture/Scope*”. The users need to right-click the “*Capture/Scope*”, then select “*Copy Widget ID*” and use this ID number for Macro widget settings of the “*Apply Fault*” button. Additionally, the types and starting times of short-circuit cases can be defined in these widgets via Python scripts. For the convenience of observing short-circuit events, the authors selected the starting time of short-circuit at 0.1s.

For “*Clear Fault*”, the “*Macro*” widget is used. In widget settings, the short-circuit fault is cleared by defining the type specified as a nominal operating condition via Python scripts inside the “*Macro*” code. For “*Reset Relay*”, the “*Check Box*” is used to send reset signals to overcurrent functional blocks in “*Relay*”.

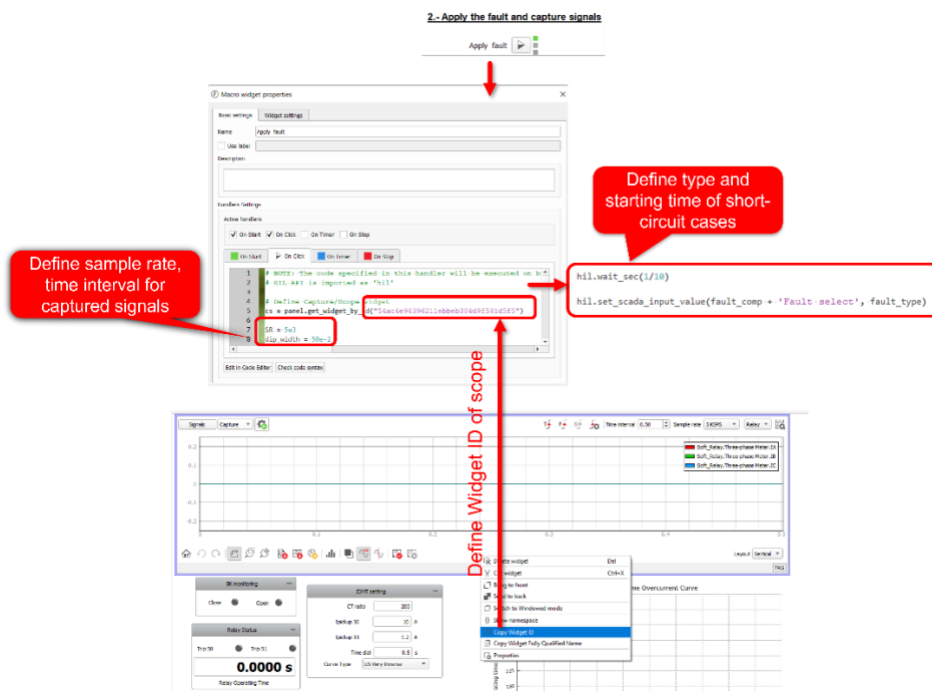


(a)





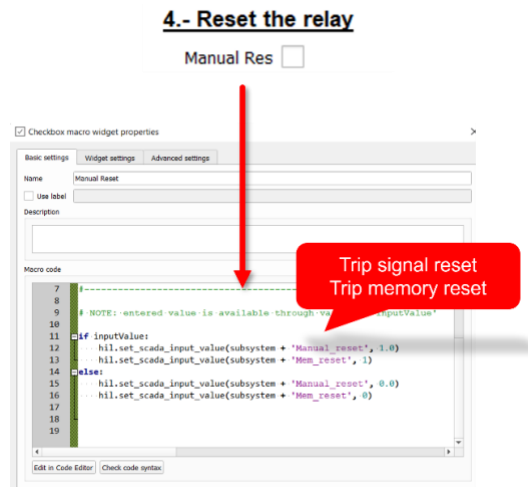
(b)



(c)



(d)



(e)

**Fig. 42.** (a) "Control" dashboard; (b) "Fault selection" action; (c) "Apply Fault" action; (d) "Clear Fault" action; (e) "Reset Relay" action.

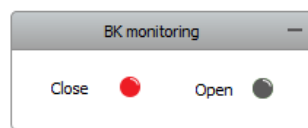
#### 4 How to use the created panel

The monitoring and controlling panels were created inside HIL SCADA that can be used for non-directional overcurrent protective relay testing.

The users need to run the real-time simulation by pressing the "Start simulation" button on the toolbar (after the HIL Schematic has been appropriately compiled).

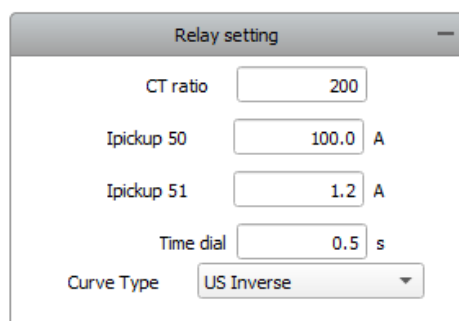
Simulation using the VHIL will start, and time will be presented at the Control bar.

The circuit breaker monitoring dashboard has two LED indicators displaying the status of the circuit breaker; when the circuit breaker is closed, the LED named "Close" is lighted in red colour, as shown in **Fig. 43**.



**Fig. 43.** The red colour of LED in "Close" indicates the ready state to perform tests.

To test the non-directional overcurrent functions, the protection relay settings configuration, including CT ratio, ANSI 50/51 pickup current, time dial and inverse operating time curves, can be predetermined at the "Relay setting" dashboard as shown in **Fig. 44**.



**Fig. 44.** The red colour of the LED in "Close" indicates ready to perform tests.

The procedure of using created panel includes four steps shown in the “Control” dashboard in **Fig. 45** as follows:

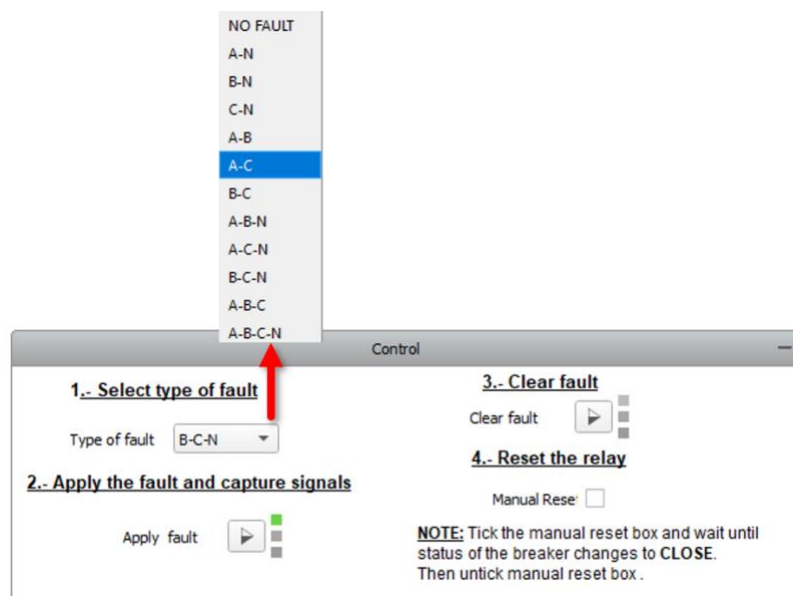
**STEP 1:** Select the type of short-circuit fault.

**STEP 2:** Apply the short-circuit fault and capture signals.

**STEP 3:** Clear the short-circuit fault.

**STEP 4:** Reset the non-directional overcurrent protective relay.

In **STEP 1**, there are many scenarios, including nominal operating cases and short-circuit fault cases. The nominal operating case is named “*NO FAULT*”. At the same time, short-circuit fault cases can be defined by many types of short-circuit: three-phase, three-phase-to-ground, single-phase-to-ground, phase-to-phase, and two-phase-to-ground; in each case, according to the *A*, *B*, *C* phases and the grounded neutral *N* as shown in **Table 7**. The users can select one of these scenarios to conduct the appropriate test.



**Fig. 45.** Procedure of performing the real-time simulation.

**STEP 2** is to press the button “Apply fault”. The captured signal is automatically activated by the condition predefined in “Capture/Scope” (see **Section 3.2**). In the visual image, the BK is lighted in green colour, and the fault icon appears simultaneously. At the “*BK monitoring*” dashboard, the LED expressing the open state of the circuit breaker is turned on. At the “*Relay Status*” dashboard, the LED indicates which overcurrent function is active, and the operating time of the protective relay is shown. The inverse-time overcurrent curve is plotted through the XY-graph widget according to pre-set values in the “*Relay setting*” dashboard, as shown in **Fig. 47**.

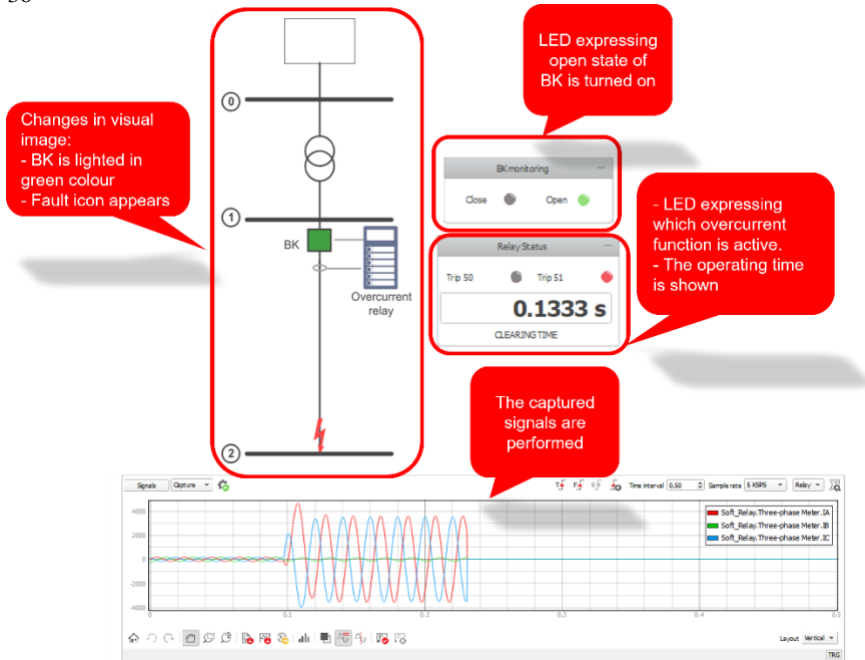


Fig. 46. Procedure of performing real-time simulation.

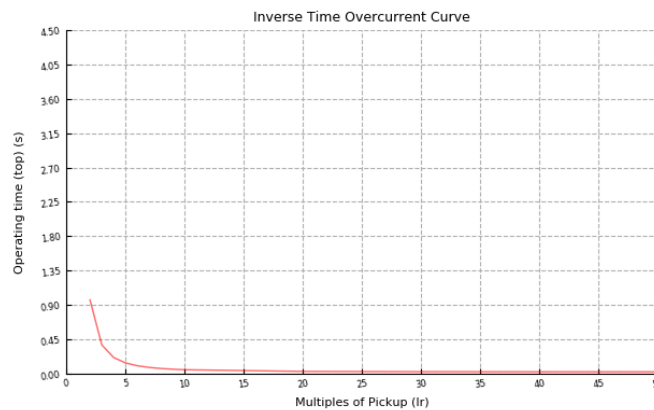


Fig. 47. Inverse-time overcurrent curve plotted after performing “Apply Fault”.

**STEP 3** and **STEP 4** are performed simultaneously for the purpose of returning the protective relay to its initial state to conduct further tests. The users need to click the button “Clear Fault” first and then tick the box “Manual Reset”. The clicking of the button “Clear Fault” clears the fault in the test system. And, ticking “Manual Reset” box returns the test system to the initial state. All necessary calculation and observation are computed and stored before running another test. The users need to untick the “Manual Reset” box before implementing further tests.

## 5 Results and Discussions

This section presents the results of simulating different short-circuit fault scenarios at the end of the test model’s overhead distribution line (Line 1-2). The main objective of this section is to observe the behaviour of overcurrent protection relay through model-based toolchains of Typhoon HIL as the following assumptions:

- The overcurrent protection functions ANSI 50 and ANSI 51 are considered. Time dial, ANSI 50/51 pickup current and curve characteristics are known as protection relay settings. The operating time of the protection relay in two cases, instantaneous ANSI 50 and different inverse overcurrent time curves of ANSI 51, is presented.

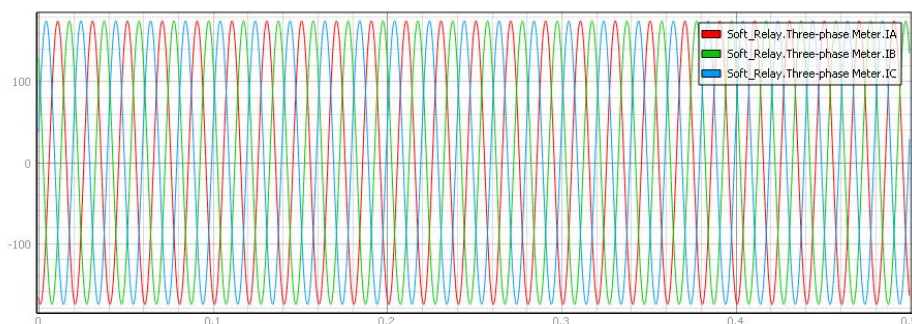
- The current waveforms under multiple short-circuit cases, three-phase, three-phase-to-ground, single-phase-to-ground, phase-to-phase, and two-phase-to-ground, are presented using Typhoon HIL.

For these objectives, the authors provided a proposed set of predetermined settings for non-directional overcurrent functions for testing the operating time of the protection relay.

### 5.1 Predetermined protective relay settings

The set of predetermined settings consists of CT ratio and values of pickup current ANSI 50/51. This section illustrates the proposed methodology of predetermined settings selection.

For CT ratio, the accurate CT ratio reflects secondary values of current through the protection relay and directly influences to protection relay's operation to send a trip signal or not. The methodology to select the CT ratio for the overcurrent protection relay is based on the nominal primary operating current according to [25]. Therefore, to calculate the CT ratio, the total load current up to the point of the protection relay should be considered. The total load current through the protection relay is the sum of load B current which is given in **Table 4** with a value of 120A. The users can retrieve this value from the waveform of nominal operating current by selecting the "NO FAULT" option in **STEP 1** (see **Section 4**), as shown in **Fig. 48**.



**Fig. 48.** Waveforms of three-phase current in nominal operating case.

The authors used CT ratio for protection relay due to 125% nominal operating current as an overload situation. As a consequence, the current transformer CT ratio of 200 (the nearest value above the overload current) is selected and used for all the simulation scenarios.

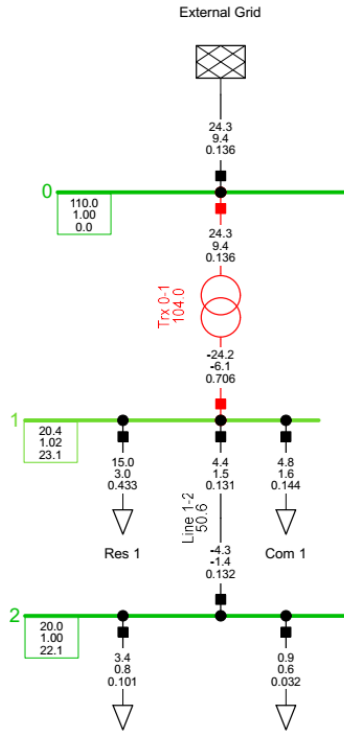
The pickup current is the operation condition of the overcurrent protective relay upon the current through it exceeds that value. The selection of threshold values depends on the specific system in the particular situations corresponding to the stated objective of protection engineers. The value of pickup current (ANSI 50 and ANSI 51) proposed by the authors is twice the value of the nominal operating current through the protection relay of 240 A on the primary side and 1.2 A on the secondary side, according to the ratio of CT.

There are two specific testing cases with particular settings according to the main purpose of this chapter is testing overcurrent functions, ANSI 50 and 51, as follows:

- **CASE I, INSTANTANEOUS:** For testing the operating time of the instantaneous overcurrent function (ANSI 50), the pickup current of the ANSI 50 function is 1.2A.
- **CASE II, INVERSE-TIME:** The pickup current settings in inverse time overcurrent testing (ANSI 51) is 1.2A and the time dial, TD, is 0.5.

### 5.2 Short-circuit study validation

The RMS values of short-circuit scenarios performed by using VHIL Device in the *Test System* are compared with the results from the DIgSILENT PowerFactory modelling. DIgSILENT PowerFactory is the widespread and leading power system analysis software, which covers the full range of functionality from standard features to the novel, sophisticated and advanced applications [26].



**Fig. 49.** Modelling *Test System* in DIgSILENT PowerFactory.

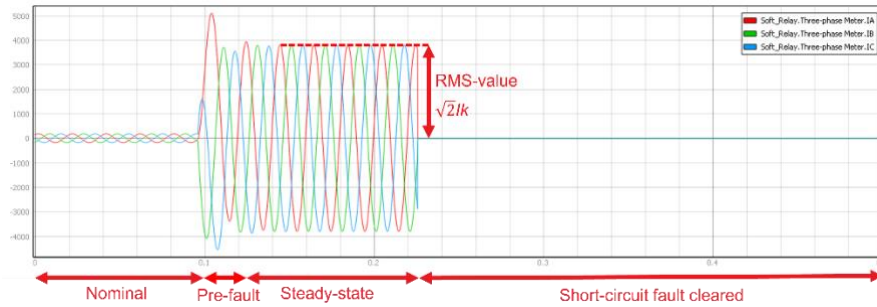
The authors used SP2 DIgSILENT PowerFactory version 2022 to model *Test System* and to perform the implementation of short-circuit calculation. The “*Complete*” method in calculating short-circuit is applied.

To compare the RMS short-circuit values obtained from VHIL and DIgSILENT PowerFactory, a summary of RMS values of short-circuit current ( $I_k$ ) for all scenarios is presented in **Table 8**. The RMS value from DIgSILENT PowerFactory is obtained by RMS simulation. The RMS value of short-circuit current from VHIL device is extracted from waveform as shown in **Fig. 50**. The waveform in **Fig. 50** is obtained from the steady-state waveform of short-circuit current when the settings in **CASE II** is applied to the protective relay.

**Table 8.** Comparison of RMS short-circuit values between VHIL Device and DIgSILENT PowerFactory.

Short-circuit case	VHIL Device $I_k$ [kA]	DIgSILENT PowerFactory $I_k$ [kA]	Per cent error [%]
A-N	2.38	2.42	1.65
B-N	2.38	2.42	1.65
C-N	2.38	2.42	1.65
A-B	2.28	2.28	0.00
A-C	2.28	2.28	0.00
B-C	2.28	2.28	0.00
A-B-N	2.53	2.55	0.78
A-C-N	2.53	2.55	0.78

B-C-N	2.53	2.55	0.78
A-B-C	2.68	2.64	1.51
A-B-C-N	2.68	2.64	1.51



**Fig. 50.** State period of short-circuit event.

The maximum relative error between RMS short-circuit current in all scenarios is 1.65%. This indicates a great similarity between the RMS values of the short-circuit cases retrieved from the VHIL Device and DIgSILENT PowerFactory.

### 5.3 Testing Non-directional Overcurrent Protective Relay using VHIL Device

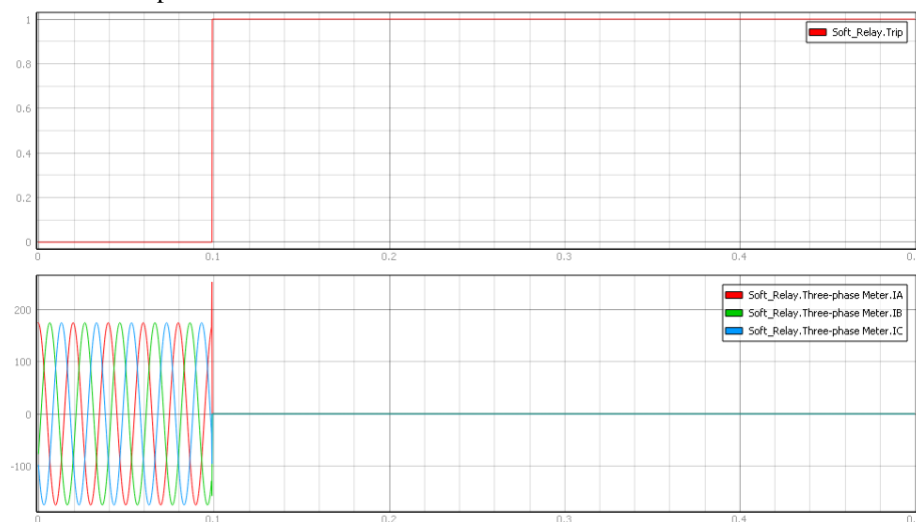
Considering the aforementioned settings of non-directional overcurrent protection relay, the results of two testing cases are presented in this section.

#### CASE I: Testing the operating time of the instantaneous overcurrent protective relay

The idea of instantaneous overcurrent function is no intentional time delay when the current gets higher than a pre-set value.

For illustration purposes, the waveforms and tripping sequences for three-phase fault are presented using SCADA HIL in **Fig. 51**.

A summary of the operating time of the entire simulation considering several short-circuit cases is presented in **Table 9**.



**Fig. 51.** Trip sequence and waveforms of three-phase current considering a three-phase short-circuit in instantaneous overcurrent testing. The short-circuit event starts at 0.1s, and the trip operates instantly at 0.1s (from 0-inactive to 1-active).

**Table 9.** The operating time of protection relay in instantaneous overcurrent testing

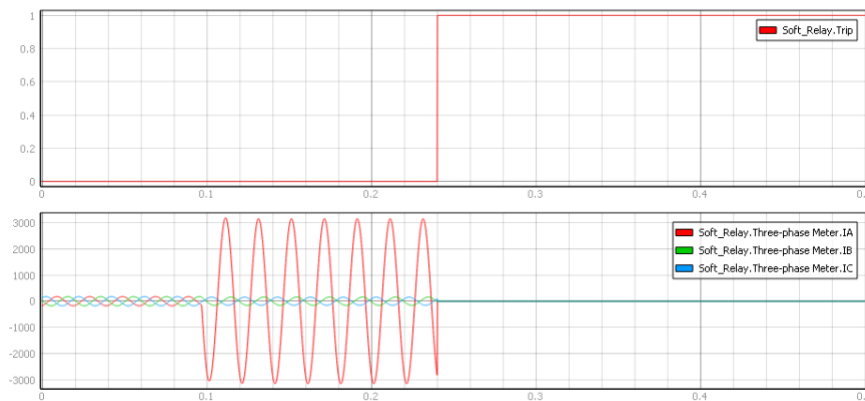
Short-circuit cases	Operating time (ms)	Cycles <sup>1</sup>
A-N	0.200	0.010
B-N	0.200	0.010
C-N	0.200	0.010
A-B	0.080	0.004
A-C	0.080	0.004
B-C	0.080	0.004
A-B-N	0.060	0.003
A-C-N	0.060	0.003
B-C-N	0.060	0.003
A-B-C	0.060	0.003
A-B-C-N	0.060	0.003

The operating time of the protection relay under short-circuit fault scenarios is extremely small (less than 10 ms), showing the concept of the overcurrent function testing that is consistent with the idea of instantaneous overcurrent function ANSI 50.

The results shown in **Table 9** give the excellent performance of the non-directional overcurrent protective relay implemented in the real-time simulation.

#### **CASE II: Testing the operating time of the inverse-time overcurrent protective relay**

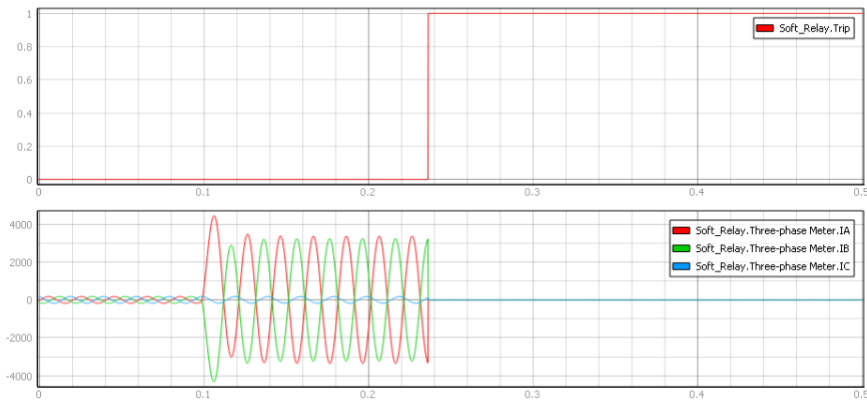
For inverse time overcurrent relay, the operating time is changed according to the case of short-circuiting and inverse time overcurrent curves. For illustration purposes, the waveforms of three-phase current and tripping sequences for multiple short-circuit fault scenarios are presented using SCADA HIL in **Fig. 52**.



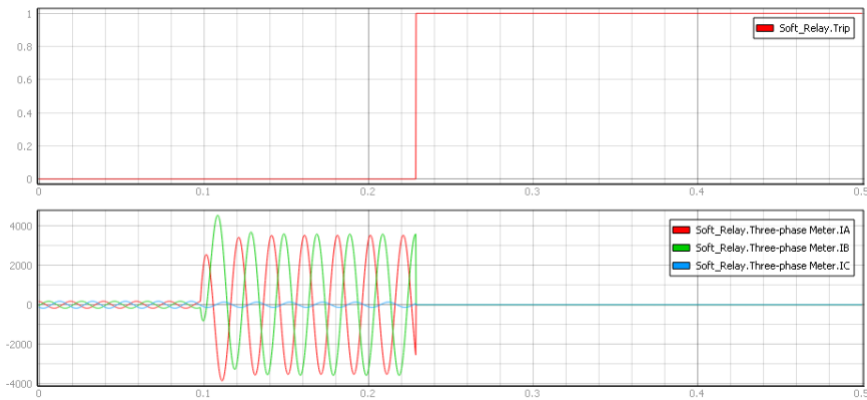
(a)

<sup>1</sup> One cycle is equal to 20 milliseconds (ms) at 50 Hz.

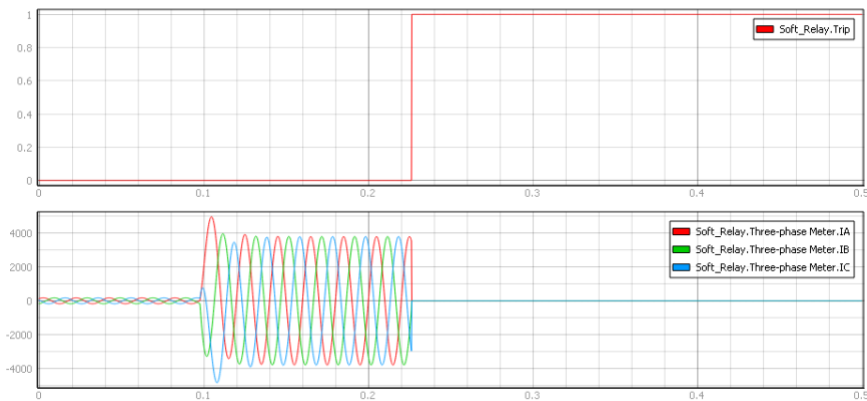




(b)



(c)



(d)

**Fig. 52.** Trip sequences and waveforms of inverse time overcurrent testing using U1 curve for short-circuit cases (a) single line to ground (A-N), (b) phase-to-phase (A-B), (c) two-phase-to-ground (A-B-N), (d) Three-phase (A-B-C).

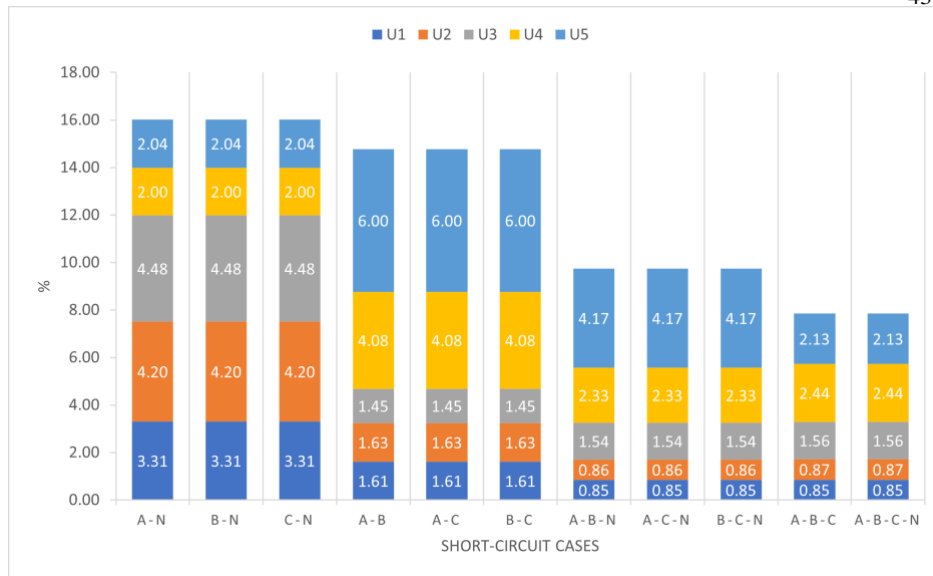
The operating time of the protective relay in this section can be retrieved from equation (1) to equation (5) based on the RMS values of short-circuit current derived from DIgSILENT PowerFactory in **Table 8**. A summary of the protection relay's operating time of the entire simulation, considering several short-circuit cases, is presented in **Table 10** and **Table 11**.

**Table 10.** The operating time of protection relay in milliseconds. (1) Equation-based calculation, (2) VHIL Device simulation.

Curve Cases	U1		U2		U3		U4		U5	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
A-N	125	121	124	119	70	67	49	50	48	49
B-N	125	121	124	119	70	67	49	50	48	49
C-N	125	121	124	119	70	67	49	50	48	49
A-B	122	124	121	123	68	69	47	49	47	50
A-C	122	124	121	123	68	69	47	49	47	50
B-C	122	124	121	123	68	69	47	49	47	50
A-B-N	119	118	115	116	64	65	42	43	46	48
A-C-N	119	118	115	116	64	65	42	43	46	48
B-C-N	119	118	115	116	64	65	42	43	46	48
A-B-C	116	117	114	115	63	64	40	41	46	47
A-B-C-N	116	117	114	115	63	64	40	41	46	47

**Table 11.** The operating time of protection relay in cycles. (1) Equation-based calculation, (2) VHIL Device simulation.

Curve Cases	U1		U2		U3		U4		U5	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
A-N	6.25	6.05	6.20	5.95	3.50	3.35	2.45	2.50	2.40	2.45
B-N	6.25	6.05	6.20	5.95	3.50	3.35	2.45	2.50	2.40	2.45
C-N	6.25	6.05	6.20	5.95	3.50	3.35	2.45	2.50	2.40	2.45
A-B	6.10	6.20	6.05	6.15	3.40	3.45	2.35	2.45	2.35	2.5
A-C	6.10	6.20	6.05	6.15	3.40	3.45	2.35	2.45	2.35	2.5
B-C	6.10	6.20	6.05	6.15	3.40	3.45	2.35	2.45	2.35	2.5
A-B-N	5.95	5.90	5.75	5.80	3.20	3.25	2.10	2.15	2.30	2.40
A-C-N	5.95	5.90	5.75	5.80	3.20	3.25	2.10	2.15	2.30	2.40
B-C-N	5.95	5.90	5.75	5.80	3.20	3.25	2.10	2.15	2.30	2.40
A-B-C	5.80	5.85	5.70	5.75	3.15	3.20	2.00	2.05	2.30	2.35
A-B-C-N	5.80	5.85	5.70	5.75	3.15	3.20	2.00	2.05	2.30	2.35



**Fig. 53.** Percent error of operating time of protection relay when comparing values from VHIL Device with Equation-based calculation in short-circuit cases.

Considering the aforementioned fault scenarios, it is noticed from the results that the short-circuit is cleared after delay time depending on the type of short-circuit cases and inverse time overcurrent curve. As can be seen from the above results, there is a great similarity between the cases study simulated in VHIL Device and equation-based calculation.

The most significant error or discrepancy between equation-based calculation and VHIL simulation is 6% for a single-line-to-ground short-circuit fault as shown in Fig. 53. The difference between equation-based calculation and VHIL simulation may be due to the differences (assumptions made during modeling, errors in measurement and signal processing) between the real object and its assumed model. However, considering the overall discrepancy, this error can be assumed in an acceptable level. This allows comparing the operation results with theoretically expected ones, since both models are comparable.

## 6 Conclusions and Recommendations

This chapter implemented a simplified version of the three-phase radial feeder, MW CIGRE European's distribution benchmark system, using a real-time simulation platform.

The implementation of a non-directional overcurrent protection relay using the Virtual HIL Device from model-based toolchains of Typhoon HIL is presented in this chapter. The so-called "*Soft Relay*" can be used to test and validate protection settings without the need to use the real hardware or any other physical devices.

The test system was used to test the overcurrent protection functions of a protection relay. Instantaneous overcurrent and inverse-time overcurrent functionalities of a non-directional protection relay were tested considering predetermined setting configuration.

Presenting the methodology for creating and testing a model, including a protection relay to be used in real-time simulation, allows the scientist community to use this chapter as a starting point for many other possible protection schemes.

## Reference

1. S. H. Horowitz and A. G. Phadke, Power system relaying. John Wiley & Sons, 2014.
2. M. M. Aman, M. Q. A. Khan, and S. A. Qazi, 'Digital directional and non-directional over current relays: Modelling and performance analysis', NED Univ. J. Res., vol. 8, no. 2, pp. 70–85, 2011.

3. Review of hardware-in-the-loop - a hundred years progress in the pseudo-real testing', *E+E Scientific Journal*, Aug. 09, 2019. <https://epluse.ceec.bg/review-of-hardware-in-the-loop-a-hundred-years-progress-in-the-pseudo-real-testing/> (accessed Aug. 07, 2022).
4. S. Hubschneider et al., 'Requirements for Power Hardware-in-the-Loop Emulation of Distribution Grid Challenges', in 2018 53rd International Universities Power Engineering Conference (UPEC), Glasgow, Sep. 2018, pp. 1–6. doi: 10.1109/UPEC.2018.8541851.
5. C. S. Edrington, M. Steurer, J. Langston, T. El-Mezyani, and K. Schoder, 'Role of Power Hardware in the Loop in Modeling and Simulation for Experimentation in Power and Energy Systems', *Proc. IEEE*, vol. 103, no. 12, pp. 2401–2409, Dec. 2015, doi: 10.1109/JPROC.2015.2460676.
6. P. Kelm et al., 'Hardware-in-the-Loop Validation of an Energy Management System for LV Distribution Networks with Renewable Energy Sources', *Energies*, vol. 15, no. 7, p. 2561, Apr. 2022, doi: 10.3390/en15072561.
7. M. Kezunovic, A. Esmailian, G. Manimaran, and A. Mehrizi-Sani, 'The Use of System in the Loop, Hardware in the Loop, and Co-modeling of Cyber-Physical Systems in Developing and Evaluating New Smart Grid Solutions', 2017. doi: 10.24251/HICSS.2017.385.
8. S. D'Arco, T. D. Duong, and J. Are Suul, 'P-HiL Evaluation of Virtual Inertia Support to the Nordic Power System by an HVDC Terminal', in 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Oct. 2020, pp. 176–180. doi: 10.1109/ISGT-Europe47291.2020.9248905.
9. D. A. M. Montaña, D. F. C. Rodriguez, D. Ivan Clavijo Rey, and G. Ramos, 'Hardware and Software Integration as a Realist SCADA Environment to Test Protective Relaying Control', *IEEE Trans. Ind. Appl.*, vol. 54, no. 2, pp. 1208–1217, Mar. 2018, doi: 10.1109/TIA.2017.2780051.
10. A. V. Pazdcrin, V. O. Samovlenko, V. A. Tashchilin, P. V. Chusovitin, A. V. Dymshakov, and Y. V. Ivanov, 'Platform for Testing Iec 61850 Control Systems Using Real-Time Simulator', in 2018 International Youth Scientific and Technical Conference Relay Protection and Automation (RPA), Sep. 2018, pp. 1–14. doi: 10.1109/RPA.2018.8537188.
11. D. F. C. Rodriguez, J. D. P. Osorio, and G. Ramos, 'Virtual Relay Design for Feeder Protection Testing With Online Simulation', *IEEE Trans. Ind. Appl.*, vol. 54, no. 1, pp. 143–149, Jan. 2018, doi: 10.1109/TIA.2017.2741918.
12. D. Celeita, A. Flores, G. Ramos, and M. Pohl, 'Design of Virtual Distance Protection for Offline Transmission Line Relay Testing', in 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII), Nov. 2018, pp. 1–6. doi: 10.1109/CONCAPAN.2018.8596426.
13. A. S. Makhzani, M. Zarghami, B. Falahati, and M. Vaziri, 'Hardware-in-the-loop testing of protection relays in distribution feeders with high penetration of DGs', in 2017 North American Power Symposium (NAPS), Sep. 2017, pp. 1–6. doi: 10.1109/NAPS.2017.8107191.
14. J. R. Camarillo-Peñaranda, M. Aredes, and G. Ramos, 'Hardware-in-the-loop Testing of Virtual Distance Protection Relay', in 2020 IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (I&CPS), Jun. 2020, pp. 1–6. doi: 10.1109/ICPS48389.2020.9176775.
15. C. T. F. C6. 04.02, Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources. International Council on Large Electric Systems Paris, France, 2014.
16. V. M. & R. Mehta, *Principles of Power System: Including Generation, Transmission, Distribution, Switchgear and Protection : for B.E/B.Tech., AMIE and Other Engineering Examinations*. S. Chand Publishing, 2005.
17. 'IEEE Standard for Inverse-Time Characteristics Equations for Overcurrent Relays', *IEEE Std C37112-2018 Revis. IEEE Std C37112-1996*, pp. 1–25, Feb. 2019, doi: 10.1109/IEEESTD.2019.8635630.
18. S. IEC, 'Measuring relays and protection equipment-Part 151: Functional requirements of over/under current protection', *IEC 60255-151*, 2009.
19. 'IEEE Standard Electrical Power System Device Function Numbers, Acronyms, and Contact Designations', *IEEE Std C372-2008 Revis. IEEE Std C372-1996*, pp. 1–48, Oct. 2008, doi: 10.1109/IEEESTD.2008.4639522.
20. T. Schossig, 'Testing in IEC 61850 - advanced topics and extended possibilities', pp. 61–61, Jan. 2010, doi: 10.1049/cp.2010.0233.
21. 'What is Python? Executive Summary', [Python.org. https://www.python.org/doc/essays/blurbl/](https://www.python.org/doc/essays/blurbl/) (accessed Jan. 16, 2022).
22. 'ANSI protective functions'. <https://www.typhoon-hil.com/documentation/typhoon-hil->

- software-manual/References/ansi\_protective\_functions.html (accessed Jul. 11, 2022).
23. 'SR Flip Flop'. [https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/sr\\_flip\\_flop.html](https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/sr_flip_flop.html) (accessed Jul. 29, 2022).
  24. 'xml.etree.ElementTree — The ElementTree XML API — Python 3.10.5 documentation'. <https://docs.python.org/3/library/xml.etree.elementtree.html> (accessed Jul. 30, 2022).
  25. P. Fonti, 'Current transformers: how to specify them', Schneider Electr. Cah. Tech. Merlin Gerin, no. 194, 2000.
  26. 'PowerFactory - DIgSILENT'. <https://www.digsilent.de/en/powerfactory.html> (accessed Jun. 06, 2022).