On Design and Implementation of Real-time, High-fidelity Virtual Power System for Smart Grid Cyber Range

Muhammad M. Roomi
IEEE Senior Member
Illinois ARCS
Singapore
roomi.s@iarcs-create.edu.sg

Isaac Lim Kok Hwee
School of Computing
National University of Singapore
Singapore
izlim@u.nus.edu

Daisuke Mashima
Illinois ARCS
Singapore
daisuke.m@iarcs-create.edu.sg

***Abstract*—Due to the emerging cyber threats and the necessity to advance the security of smart grids, an environment to conduct intensive cyber security research is essential. Cyber security experiments on a real infrastructure is infeasible due to the practical complications and thereby, virtual testbeds are a viable solution. However, many of the existing solutions either rely on lower-fidelity power system simulator or high-end real-time simulation hardware, which is costly, and accessibility is limited. In this paper, a near-real-time, power system dynamics simulation, using Simulink, that can be integrated into a smart grid cyber range for interactive cyber security experiments is developed and explored as a solution. The interoperational control and data exchange with external components in the cyber range is achieved through TCP and low-latency in-memory database, respectively. Furthermore, Simulink parameters are calibrated in order to balance real-timeness and fidelity of dynamics simulation. Finally, cyberattack experiments are demonstrated using the developed interoperational simulator model.**

***Index Terms*—Cyber range, MATLAB Simulink, Database, Smart Grid, Cyber security**

I. Introduction

The advancement in the technology has facilitated the integration of automation and digitalisation in the electrical grid. The monitoring and the managing of electricity from generation end to distribution end is crucial. The communication between the generating station, end users, grid operators and stakeholders have to be coordinated efficiently in order to minimise the cost and environmental impacts and maximise reliability, flexibility and security [1]. However, the increased dependence on automation and digitalisation leads to increased exposure to cyber-attacks [2], [3]. Some of the prominent attacks are: 1) Iran's nuclear power plant attack - Stuxnet [4] in 2010; 2) Ukraine's power station attack - Trojanhorse malware BlackEnergy [5] in 2015; and 3) Global cyber-attack on popular organisations - WannaCry ransomware [6] in 2017. Some of the recent cyber-attacks are: 1) Ukranian's energy company attack - Industroyer2 [7] in 2022; 2) India's electrical grid attack - ShadowPad [8] in 2021 & 2022. As one of the critical infrastructures, the electrical sector has to be protected against these cyber-attacks.

The security posture of Industrial Control System (ICS) devices and the infrastructure resilience have to be assessed to protect the system against imminent cyber-attacks. However, due to the set-up, operating and the maintenance costs involved in physical or hybrid testbeds, it is neither practical nor cost effective to conduct cyber experiments on the real infrastructure. The consequences of these experiments could damage any costly component or even affect the availability of the infrastructure. As a result, digital twin or cyber-physical range is on demand. Cyber-physical range emulates any real infrastructure in a virtual environment, with high precision. These environments provide the flexibility in scaling up/scaling down the system, conducting wide range of cyber-attacks, evaluating cyber security measures, imparting training to industry personnel and academic research.

One such effort to evaluate the impact of the False Data Injection (FDI) attack in a 66/11 kV sub-transmission level cyber range was proposed in [9]. In this work, the power system is developed in Pandapower [10]. Virtual sensors/actuators are deployed to read the measurements from the simulator. Subsequently, the data is transmitted to the Programmable Logic Controller (PLC) and the Supervisory Control and Data Acquistion (SCADA) through Modbus TCP Python library [11]. Additionally, the loads, attack configuration and scenario databases are integrated into the system through MySQL [12]. Similarly, an advanced version of this cyber range, where the physical system can be generated automatically through a processor tool is detailed in [13]. Besides the physical simulator, the virtual IEDs/PLCs in [13] is implemented through OpenPLC61850 [14], [15]. In both the works, the simulator that is employed is Pandapower. This simulator can be considered as a lower-fidelity simulator due to the reasons mentioned below.

Pandapower is a one-time solver that is widely used for steady-state power flow simulation. As a result, the simulator has to be configured to run periodically. In such cases, the effect/impact of any changes (such as circuit breaker (CB) status) in the system will be reflected only in the next round of simulation. Furthermore, measurements such as system frequency, generator/transformer dynamics and the relay status are not supported. To incorporate these functionalities, additional databases were used to run the cyber range model in [9].

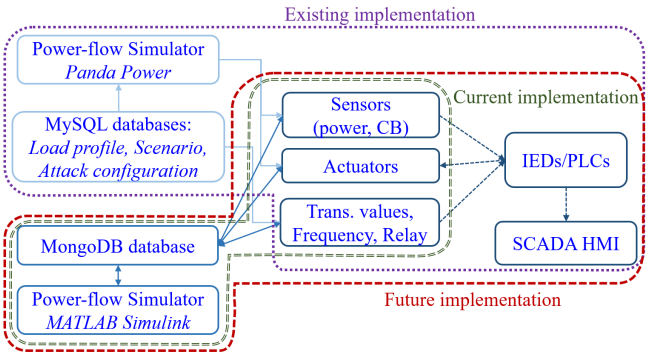

Fig. 1. Existing [9], [13], Current (this work) and Future Implementation of cyber range.

This limits the attack surface considerably. However, Simulink is a graphical block diagramming environment that allows to simulate and test multi-domain dynamic systems [16]. The compatibility with real-time hardware-in-the-loop and the functionality to operate and control in real-time makes Simulink to be a potential alternative for Pandapower in the cyber range. Therefore, in this paper the possibility of using Simulink as the power system simulator is analysed. The design requirements, the compatibility evaluation, the challenges and the approached solutions are the core contributions of this paper. Subsequently, the simulation analysis and the attack experiments are presented. Fig. 1 illustrates the schematics of the cyber range implementation. The section highlighted in purple is the existing implementation with Pandapower, the blocks inside green color highlights the current implementation with Simulink and the future implementation with the virtual devices are included within red colored section.

The rest of the paper is organised as follows. In Section II, related works are discussed. Section III presents the system design, the challenges and the approached solutions. The operation of the model is explained in Section IV, followed by cyber-attack experiment in Section V. Finally, the paper is concluded with future research directions in Section VI.

II. Related Works

Physical, hybrid and virtual testbeds are the effective approaches to conduct research, training and cyber experiments. Few significant physical testbeds are 1) Experimental testbed in the Institute for the Protection and Security of the Citizen, Italy to enhance cyber security of monitoring devices [17]; 2) SCADA testbed in Idaho National Laboratory [18] for the analysis of impacts of attacks; 3) Testbed in University of Arkansas to enhance cyber security against false data injection attacks on distributed resources [19]; 4) Electric Power and Intelligent Control (EPIC) testbed in Singapore University of Technology and Design [20] for evaluating novel cyber defense mechanisms for power grid. Similarly, there are hybrid testbeds that includes both virtual and hardware-in-the-loop components for the purpose of cyber security. Some of them are: 1) OPNET system-in-the-loop environment by Sandia National Laboratory to integrate physical component with the virtual network [21]; 2) Virtual testbed for power system by University of Illinois [22]; 3) cyber-physical environment to simulate using Real Time Digital Simulator (RTDS) is proposed in [23]; 4) EPICTwin [24], which is a virtual hardware-in-the-loop environment for the physical EPIC testbed. Due to the limitation in the experiments that can be conducted on the physical and hybrid testbeds and also the costs associated with the physical and hybrid test-beds, virtual testbeds or cyber-physical ranges are proposed. 1) In [25], a software based smart grid testbed named 'Softgrid' is proposed; 2) In [9], a cyber range created using only open-source software to evaluate cyber-attacks on Modbus protocol in SCADA system is proposed; 3) In [13], a framework has been designed to model a smart grid range based on a XML language.

In the aforementioned virtual testbeds (2) and (3), the main component of the power system simulator is Pandapower. In order to include system dynamics with higher fidelity and to evaluate extensive cyber-attacks, MATLAB Simulink based power system simulator is proposed as a potential alternative. Furthermore, though [24] uses Simulink platform for EPIC twin, the cost associated with the hardware-in-the-loop setup is high. Therefore in this paper, a high-fidelity virtual power system is developed with comparatively low cost and operates in real-time. The details of the system model and the attack evaluation are presented in the following sections.

III. Design Requirements: Challenges & Solutions

In order to evaluate the potentiality of utilising Simulink for cyber range, integration of different components is necessary. The three core components are: a) Simulink: achieving near real-time simulation for any physical system; b) Python: providing integration with the database for real-time control of the simulation; c) Mongo Database: storing and retrieving of data from the simulation for logic implementation or a historian. In addition, an user interface is developed using Python to control the CBs in the simulation remotely.

In this section, the implementation and the challenges in integrating each component are detailed.

A. Challenges and Solutions

In order to evaluate the cyber range environment, a 66/11kV substation similar to the model in [9] is developed in Simulink environment. The single line diagram of the model is illustrated in Fig. 2. The substation model consists of a 66kV incoming feeder that delivers power to two high tension 66kV consumer loads (Load1 and Load2). In addition, the feeder voltage is also stepped down to 11kV distribution level voltage through a stop-down transformer. Three 11kV loads (Load3, Load4 and Load5) are connected to the secondary side of the transformer. The location of the CBs in the system are included in the single line diagram. These CBs are controlled externally but in real-time through an user interface and the control signal is represented as 'C'. The physical measurements from the system such as the bus voltages and line currents are sensed at different locations and the points of measurement are indicated as 'M' in the single line diagram.

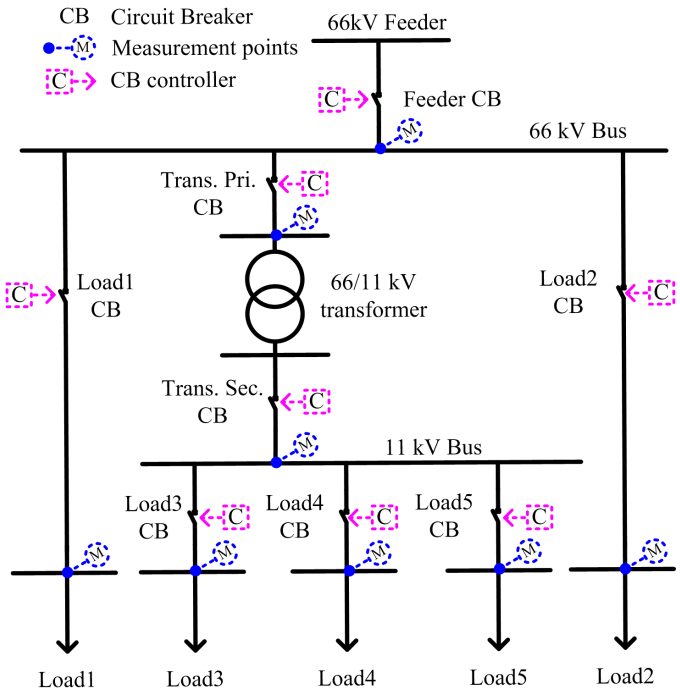

Fig. 2. Single line diagram of the sub-transmission level model.

In addition to the physical components required to build the system, communication components are also included in order to establish communication with external environments such as MongoDB [26] and user interface. The data at the measurements points 'M' (in Fig. 2) is written into the database. Similarly, the control points 'C' receives status command externally from the user interface through Python Server.

The developed model is run on a Windows 10 Pro system with the following specifications: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41 GHz, 32 GB of RAM. The final build of the simulation was run on MATLAB Simulink R2022b, Python 3.11 and MongoDB 6.0.3. Based on the configurations, the challenges faced during the modelling and the approached solutions are detailed in the following subsections.

a) Datatype Compatibility: The communication between the Simulink and the database is essential for external control of the model. To start with, the three-phase data of voltages and currents and also the status of the CB are to be retrieved from the Simulink model. Further, the retrieved data should be in a readable format. The 'TCP/UDP' blocks from the Simulink can be used for this purpose. As the motive is to utilise the model for cyber experiments, a reliable communication is necessary. As a result, TCP is preferred over UDP. Therefore, 'TCP Send/TCP Receive' block are preferred. Subsequently, a Python TCP Server was created to analyse and test the data send from the simulation. Python follows double datatype and according to [27], "MATLAB constructs the double data type according to IEEE® Standard 754 for double precision". This standard specifies a binary64 as a 64 bit binary. Therefore, 8 bytes are required for any value sent. In order to convert the data retrieved into a readable format, the 'struc' library in Python is used. Furthermore, the testing revealed some discrepancies in the data that is obtained from the simulation with the data that is written to the database. This is due to the mismatch in the byte order between the TCP Send block and the Python server. Thus, the byte order is changed to 'LittleEndian' and the values that are written into the database matched the simulation output.

b) Database selection: As the modeling involves real-time control, database plays a crucial role. The main goal was to choose a database that could read the data from the simulation in real-time and provide feedback to the simulation. The initial choice was MySQL as it is an open-source and also its flexibility to connect to Python server using MySQL connector. Though MySQL is a relational database management system, the main purpose of the database in this model is to store the data from the simulation and to use some data as a feedback to control the simulation. As such, the operation is: *Python server receive packets from TCP Send block in Simulink, unpack it using the 'struc' library and write/store the data into the MySQL database.*

Preliminarily, the data was uploaded during the runtime of the simulation. However, the data continued to upload even after the runtime, suggesting that there is a difference between the simulation time and the database writing time. While this latency is acceptable if the database is used as a historian, the feedback mechanism will be greatly affected. Therefore, an in-memory database called MongoDB is considered as an alternative. In-memory databases rely on the main memory rather than disk storage and thus it is faster in comparison with the disk memory based databases. This removes the dependency on the Python server and the data can be directly uploaded to the database. Thus, any latency introduced due to the inclusion of the Python server will be eliminated. However, populating the database through TCP Send block was not effective as the data were appended rather than sorted properly to their respective tables. Additionally, Simulink is limited in terms of direct communication with database. Conversely, MATLAB being a high-level coding language provides various interfaces for connection to databases. 'MATLAB Function' block and 'Level-2 S-Function' block supports custom MATLAB code. The MongoDB C++ interface in MATLAB code is used to create a direct interface between the database and

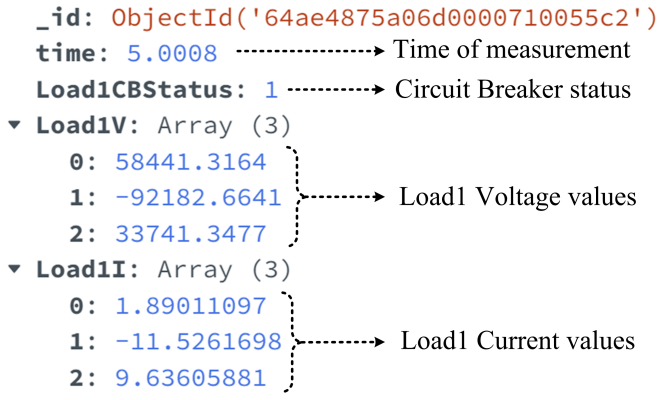

Fig. 3. Snippet of data format stored in MongoDB.

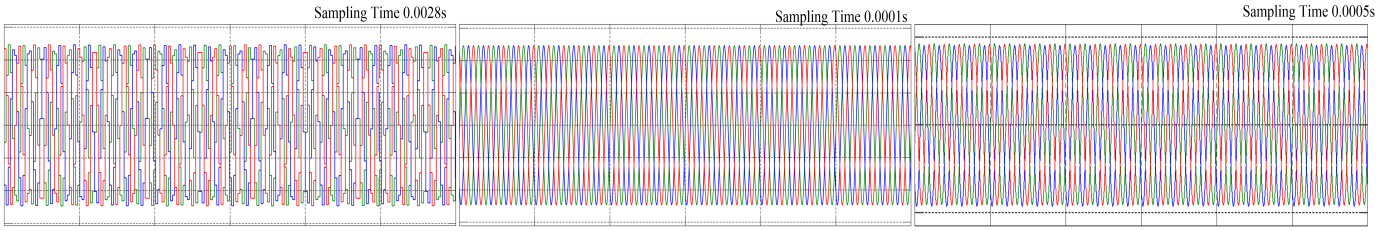

Fig. 4. Analysis of different sampling times.

the simulation. Though both the blocks support multi-inputs, the Function block processes each line of code at every time-step. This introduced more latency to the operation as the connection to the database starts and terminates at every time-step. Whereas, S-Function block has a start function that runs at the beginning of the simulation and an output function that runs at every time-step. This procedure allows to start a connection once and continue to insert data from the simulation. Through these methods, the latency was greatly addressed and the operation of the system is: *'Level-2 S-Function block receives the data from the simulation and write/store the data into the MongoDB'*. A snippet of the data that is written into the database is shown in Fig. 3.

c) Syncing simulation and database: In general, simulating a model in real-time demands a hardware-in-the-loop set-up. In this work, the model is run in real-time only through simulation. The requirements and the considerations to achieve real-time through simulation is detailed in this section. The simulation data are inserted into the database at a regular time interval. The number of data that is send from the simulation depends on the sampling interval defined in the simulation. Similarly, the interval at which the insertions into the database should happen is defined in the S-Function block. The correlation between the simulation time and the number of data inserted into the database is tested for different sampling intervals. In the case of inserting to the database at every time-step as the simulation, frequency of the model achieves an ideal sensitivity to run-time ratio at 0.0028s per time-step (first plot in Fig. 4). This is the highest rate at which the simulation can still run at real-time without affecting the speed and number of data stored per time-step. Though the insertions into the database are done at the same time-step as the simulation, as seen from the figure, the waveforms are not sinusoidal. Alternatively, if less data are inserted into the database, the sampling interval can be as high as 0.0001 sec (second plot in Fig. 4). However, less data will be inserted into the database. Therefore, a trade-off between the simulation sampling interval and the number of data required for efficient operation is necessary. With this trade-off, even a more complicated model can be simulated efficiently. As such, when insertions are decreased to once every 0.1s, the simulation can be sped up to 0.001s per time-step. To demonstrate even higher sampling intervals are acceptable, the model is run at 0.005s (last plot in Fig. 4). It is observed that the sampling time suggested is specific to the pertinent system configuration.

d) CB control & Real-time: CBs are switches with close and open mechanism that are used for protection by isolating the system from any disturbances. Nevertheless, the same mechanism can be manipulated by the attacker to destabilise the system. In this model, a user interface is created to control the operation of the CB externally. Similar to the export function of data from Simulink through TCP Send block, the command signals can be imported into the system through TCP Receive block. This includes the Python server receiving the CB status commands from the user interface and controls the CB in the model through the TCP Receive block.

Initially, a direct control method is implemented. In this method, the CB in the simulation receives a constant signal directly from the TCP Receive block to keep it closed. During the opening of the CB, the constant signal is stopped. This approach created greater lag while opening the CB compared to closing it. The time taken by the Python server to propagate

TABLE I
COMMAND PROPAGATING DELAYS (DIRECT CONTROL)

Trials	20s (open)	40s (close)	60s (open)	80s (close)
1	22.1	40.3	61.0	80.3
2	22.1	40.0	61.3	80.2
3	21.1	40.2	62.2	80.2

TABLE II
COMMAND PROPAGATION DELAYS (SWITCH CONTROL)

Trials	20s (open)	40s (close)	60s (open)	80s (close)
1	20.2	40.3	60.2	80.3
2	20.3	40.1	60.1	80.2
3	20.1	40.2	60.2	80.2

TABLE III
REAL-TIME VS SIMULATION-TIME

Sampling Interval	Real-time	Simulation-time	Ratio
0.0001 sec	23.84 sec	10 sec	2.4
0.0005 sec	10.27 sec	10 sec	1.027
0.0010 sec	10.49 sec	10 sec	1.049
0.0028 sec	10.37 sec	10 sec	1.037

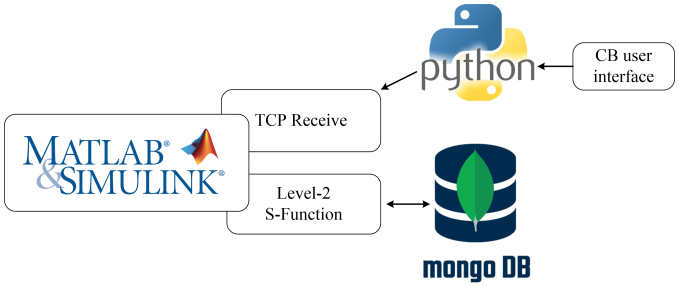

Fig. 5. High-level representation of the implementation.

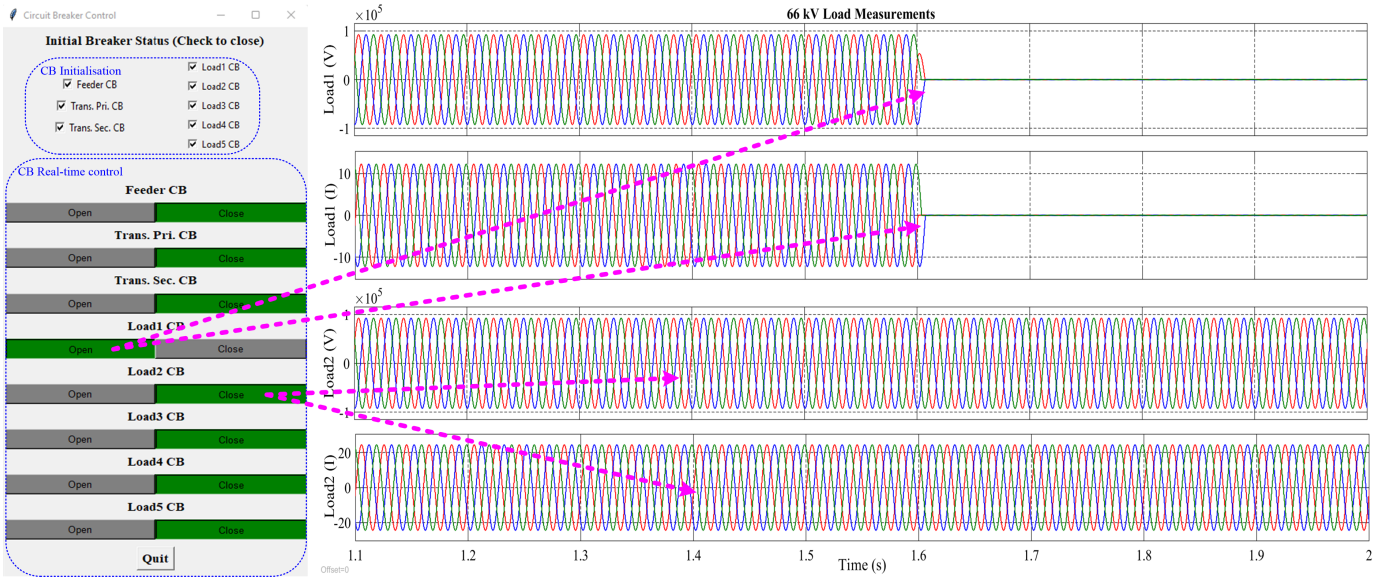

Fig. 6. Real-time control of CB through GUI in Simulink model.

the open/close signal, when triggered at 20s, 40s, 60s, 80s, is tabulated in Table. I. As can be seen, the command signal received by the CB to switch status faces a greater lag during opening command when compared with the closing command. Further analysis revealed that the simulation continues to buffer the constant signals that is previously received even after the signal is stopped. As such, the change in the CB status is reflected only when the buffered signals are exhausted. In order for the control to be realistic, there should be no delay in the controlling of CB in the model.

To address the aforementioned drawback, switch control method is implemented. In this method, a 'switch block' is introduced between the TCP Receive block and the CB. This block contains 3 inputs (2 data inputs and 1 control input) and the output can be controlled. The data inputs to this block are '0' and '1' to define the open and close status of the CB, respectively. The control input is from the TCP Receive block. A feedback mechanism is introduced to ensure the CB is continuously receiving either '0' or '1'. The trials with this set-up resulted in better propagation time when compared with the direct control method. The time taken for the propagation of open and close commands using the switch control method is tabulated in Table. II.

In order to achieve real-time simulation, a real-time pacer block is used in the Simulink model to identify and minimise the difference between the simulation-time and the real-time. The simulation is run for 10 sec and hence, the value for Simulation time in Table. III remains constant. Subsequently, the simulation is sampled at different sampling intervals. The average ratio between the simulation-time and real-time is calculated based on 10 iterations. From Table. III, it can be observed that the ratio is closer to 1 for the sampling intervals of 0.0005 sec, 0.0028 sec and 0.001 sec, whereas, it is high for 0.0001 sec. In order to achieve real-time simulation, the ratio has to be closer to 1. Furthermore, from Fig. 4, sinusoidal waves are obtained for the sampling intervals greater than 0.0005 sec. Based on the evaluation reported in Table. III and Fig. 4, the sampling interval of 0.0005 sec is utilised in this study.

IV. Simulation Analysis

The high-level representation on the implementation of the integrated model is depicted in Fig. 5. As shown in the figure, the model in Fig. 2 is developed in the Simulink environment and the network interface blocks are used to transmit measurements and receive command signals to the database and the user interface, respectively. Fig. 6 exemplifies the user interface module and the measurements (voltage and current values) of Load1 and Load2 in the 66/11kV substation model. This figure depicts the normal operation of the system through successful communication between Simulink, MongoDB and user interface. The initial status of the CBs can be defined using the user interface and the CBs can be controlled during the run-time. The active and the inactive status of the buttons are indicated using green and grey color, respectively. The functionality of the CB control is reflected in the 66kV load values. From the figure, it is evident that the status of Load1 CB is open and the respective branch voltages and currents changed to zero at 1.6s. Through this approach, any CBs in the system can be controlled externally to simulate any desired scenarios.

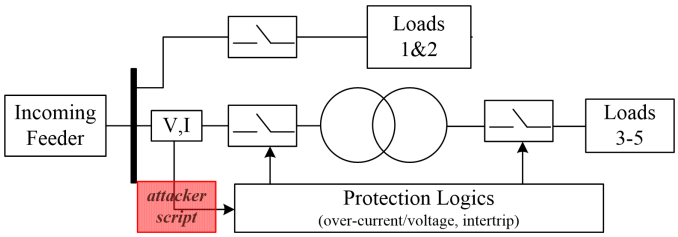

Fig. 7. Block Diagram of the system with the attacker script.

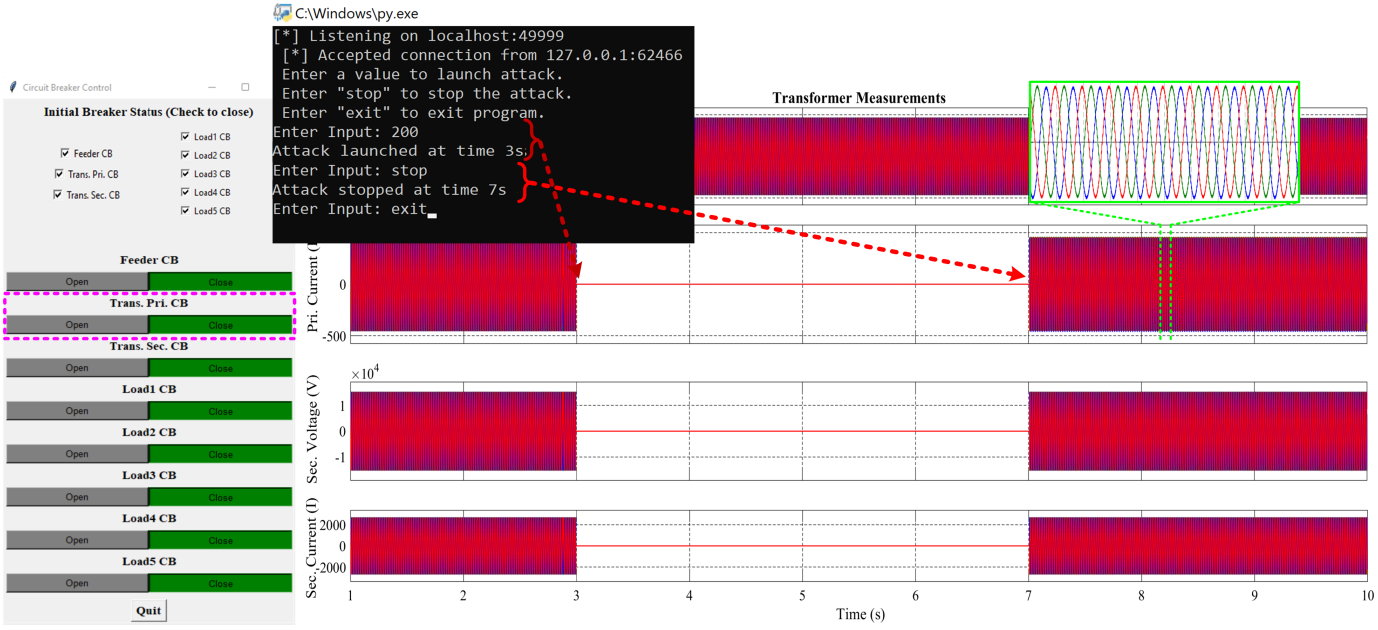

Fig. 8. Over-current attack on Transformer using attacker script with no changes in GUI.

V. Attack Experiments

Achieving real-time simulation is ideal for attack experiments, especially attacks using an external Python TCP server. Hence, even though simulation time can be much faster than real-time (i.e. simulation-time to real-time ratio of greater than 1), a real-time pacer block is used to match the simulation time with the real-time. From the aforementioned challenges and solutions, a real-time simulation based attack environment is developed. The implementation and the attack evaluation on this model are detailed in the following sections.

One of the objectives of this work is to evaluate cyber-attack experiments on the developed cyber range. Therefore in this section, the compatibility of the integrated system for cyber security research is demonstrated by emulating an attack using an attacker script. The script can be used to modify the status of any CB or manipulate the data such that false command is triggered to destabilise the system. In the real world or in the smart grid cyber range [9], this type of manipulation attack can be mounted through Man-in-the-middle or False Data Injection attacks. As this work includes only the simulator and the database, the attacker scripts are included in the Simulink. However, the attacks are not pre-scripted to ensure the real-time testing of the attack scenarios. A block diagram of the system with the attacker script is portrayed in Fig. 7.

In Simulink, the voltage and the current measurement blocks (sensors) are used to measure the instantaneous values of voltages and currents in the system. These data are utilised to provide feedback control to the system to implement the protection logic and also stored in the historian database. The output signal during the run time can be visualised using the scopes. An attacker can take control of the mechanism by spoofing or modifying the signals received and there upon triggering the protection mechanisms. To demonstrate the

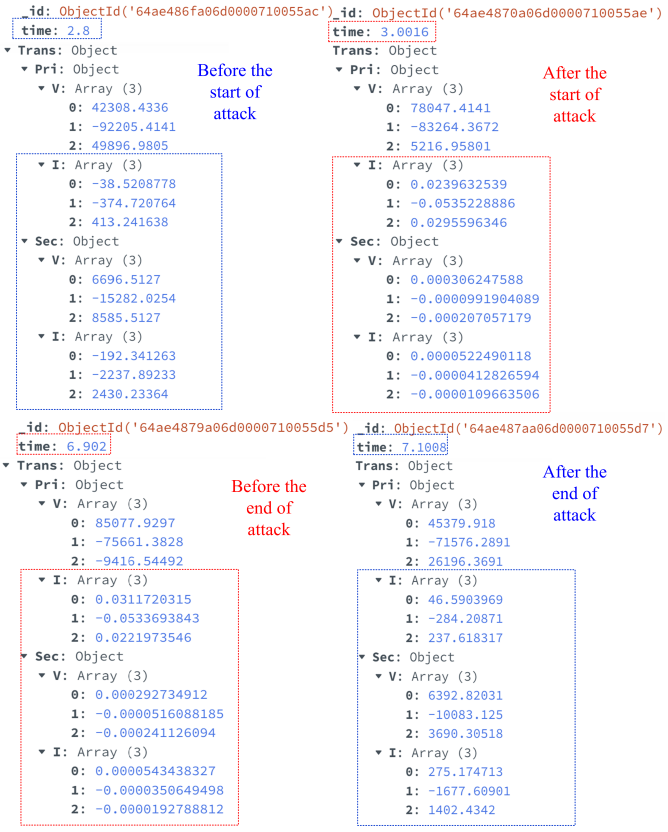

Fig. 9. Consequence of the attack reflected in the database.

attack, an attacker script is created using Python TCP Server. This script allows the attacker to modify the measurement data, thereby, the data from the sensors to the command logic and the historian database is modified. In this work, the attacker script is designed to modify the primary current in the transformer. Once the primary current starts to violate the

threshold limit, the protection logic falsely triggers the CB to open. Hence, the loads that are connected to the transformer are de-energised. Fig. 8 illustrates the result from the attack. As depicted, false data of threshold violating value is inserted through the attacker script (command prompt window in the figure) at 3s. As such, the protection logic trips the CB in the primary side of the transformer. Due to the inter-trip mechanism implemented on the CBs on either side of the transformer, the secondary side voltage and current of the transformer becomes zero. This leads to the loss of loads that are fed through the transformer. However, during this period of disconnectivity, the status of the CB in the user interface is still in closed condition. This proves that the normal data communication is successfully interfered by the attacker script. The attack is stopped at 7s. Once the attack is stopped, the transformer values are back to normal. A portion of the waveform is expanded to show the sinusoidal waveform and the values that are stored in the database during the attack period is illustrated in Fig. 9. In cases where the operator relies solely on the CB user interface to monitor the system, then the attack launched and the transformer isolation is totally masked from the view of the operator.

VI. Conclusion

In this paper, the usage of MATLAB Simulink instead of Pandapower in a cyber range is evaluated. Consequently, a substation model that can be controlled/manipulated in real-time is developed. The integration of different platforms/software is explained in detail. Subsequently, the developed model was used to implement attacks on the CB control. From the results, it is evident that the manipulating the safety mechanism of the CBs can disrupt the stable operation of the grid and in worst case, de-energising the loads.

Though the work confirms the power simulator (Simulink) to be a potential alternative for Pandapower, the cyber range model contains additional components that needs to be integrated. As a result, the future direction would be to integrate the Simulink model with the virtual Intelligent Electronic Devices (IEDs), PLCs and SCADA. This allows the creation of a full-fledged cyber range that includes both the physical and the network characteristics of the system. This completely virtual cyber range can be incorporated with the real-time target computers/PLCs/IEDs to conduct hardware-in-the-loop simulations. Furthermore, the feature to control the CB and measurements of the Simulink will be incorporated to support enhanced cyber-attacks.

Acknowledgment

This research is supported in part by the National Research Foundation, Singapore, under its National Satellite of Excellence Programme “Design Science and Technology for Secure Critical Infrastructure: Phase II”, and also supported in part by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] iea, “Smart Grids,” https://www.iea.org/energy-system/electricity/smart-grids, Accessed: 05/08/2023.

[2] V. D. Savin, “Cybersecurity threats and vulnerabilities in energy transition to smart electricity grids,” in *Navigating Through the Crisis: Business, Technological and Ethical Considerations: The 2020 Annual Griffiths School of Management and IT Conference (GSMAC) Vol 2 11*. Springer, 2022, pp. 71–83.

[3] M. M. Roomi, S. M. S. Hussain, D. Mashima, E.-C. Chang, and T. S. Ustun, “Analysis of false data injection attacks against automated control for parallel generators in iec 61850-based smart grid systems,” *IEEE Systems Journal*, pp. 1–12, 2023.

[4] J. Fruhlinger, “Stuxnet explained: The first known cyberweapon,” https://tinyurl.com/wjb9p79d, Accessed: 10/08/2023.

[5] K. Zetter, “Inside the Cunning, Unprecedented Hack of Ukraine’s Power Grid,” https://tinyurl.com/484a34ee, Accessed: 12/08/2023.

[6] A. S. Gillis, “WannaCry ransomware,” https://tinyurl.com/ysebda9a, Accessed: 10/08/2023.

[7] ESET, “Industroyer2: Industroyer reloaded,” https://tinyurl.com/4dhhdfn, Accessed: 10/08/2023.

[8] Cyberscoop, “Threats,” https://tinyurl.com/52hs4jsp, Accessed: 10/08/2023.

[9] M. M. Roomi, P. P. Biswas, D. Mashima, Y. Fan, and E.-C. Chang, “False Data Injection Cyber Range of Modernized Substation System,” in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*. IEEE, 2020, pp. 1–7.

[10] pandapower, “pandapower,” http://www.pandapower.org/, Accessed: 30/07/2023.

[11] python, “pyModbusTCP,” https://pypi.org/project/pyModbusTCP/, Accessed: 29/07/2023.

[12] “MySQL,” https://www.mysql.com/, Accessed: 25/07/2023.

[13] D. Mashima, M. M. Roomi, B. Ng, Z. Kalbarczyk, S. Hussain, and E.-C. Chang, “Towards automated generation of smart grid cyber range for cybersecurity experiments and training,” in *Proceedings of Dependable Systems and Networks 2023 (Industry Track)*, 2023.

[14] M. M. Roomi, W. S. Ong, D. Mashima, and S. S. Hussain, “Openplc61850: An iec 61850 mms compatible open source plc for smart grid research,” *SoftwareX*, vol. 17, p. 100917, 2022.

[15] M. M. Roomi, W. S. Ong, S. M. S. Hussain, and D. Mashima, “Iec 61850 compatible openplc for cyber attack case studies on smart substation systems,” *IEEE Access*, vol. 10, pp. 9164–9173, 2022.

[16] Mathworks, “Simulink,” https://ww2.mathworks.cn/en/products/simulink.html, Accessed: 05/08/2023.

[17] I. N. Fovino, M. Masera, L. Guidi, and G. Carpi, “An experimental platform for assessing scada vulnerabilities and countermeasures in power plants,” in *3rd International Conference on Human System Interaction*, 2010, pp. 679–686.

[18] I. N. Laboratory, “National SCADA Test Bed Substation Automation Evaluation Report,” https://inldigitallibrary.inl.gov/sites/sti/sti/4374057.pdf, 2009, Accessed: 07/08/2023.

[19] H. Albunashee, C. Farnell, A. Suchanek, K. Haulmark, R. McCann, J. Di, and A. Mantooth, “A testbed for detecting false data injection attacks in systems with distributed energy resources,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pp. 1–1, 2019.

[20] iTrust, “Electric power and intelligent control,” https://itrust.sutd.edu.sg/testbeds/electric-power-intelligent-control-epic/, Accessed: 12/08/2023.

[21] M. McDonald *et al.*, “Modeling and simulation for cyber-physical system security research, development and applications,” *Sandia National Laboratories, Tech. Rep. Sandia Report SAND2010-0568*, 2010.

[22] D. C. Bergman, D. K. Jin, D. M. Nicol, and T. Yardley, “The virtual power system testbed and inter-testbed integration.” *CSET*, vol. 9, pp. 5–5, 2009.

[23] S. Poudel, Z. Ni, and N. Malla, “Real-time cyber physical system testbed for power system security and control,” *International Journal of Electrical Power & Energy Systems*, vol. 90, pp. 124–133, 2017.

[24] N. K. Kandasamy, S. Venugopalan, T. K. Wong, and L. J. Nicholas, “Epictwin: an electric power digital twin for cyber security testing, research and education,” *arXiv preprint arXiv:2105.04260*, 2021.

[25] P. Gunathilaka, D. Mashima, and B. Chen, “Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions,” in *Proceedings of the 2nd ACM workshop on cyber-physical systems security and privacy*, 2016, pp. 113–124.

[26] “Mongodb,” https://www.mongodb.com/, Accessed: 30/07/2023.

[27] Mathworks, “double,” https://ww2.mathworks.cn/help/matlab/ref/double.html, Accessed: 12/08/2023.