Chapter 16

ENABLING THE EXPLORATION OF OPERATING PROCEDURES IN CRITICAL INFRASTRUCTURES

Christos Siaterlis, Bela Genge, Marc Hohenadel and Marco Del Pra

Abstract Modern testbeds for the experimental analysis of critical infrastructures either totally ignore the human factor or incorporate real human-machine interfaces (HMIs) and software that require the presence of human operators during an experiment. Although experimentation with humans in the loop can provide invaluable experimental data about human decision making and reactions, it is infeasible to conduct a systematic exploration of the vast parameter space of possible human operator decisions, reasoning and actions. This paper argues that testbeds should incorporate simulated human decision making capabilities in order to engage humans in the loop, especially because humans play crucial roles in cyber security experiments involving critical infrastructures. An extension of a previously developed experimentation framework is also described; the extension provides generic "human decision" units that enable the integration of human operator and HMI models. The utility of the approach is demonstrated by assessing the impact of human operator reactions during an attack on a cyber-physical infrastructure incorporating the IEEE 30-bus power grid model.

Keywords: Critical infrastructures, operating procedures, security, simulation

1. Introduction

Most investigations that focus on critical infrastructures (e.g., power plants, smart grids and water treatment facilities) highlight the fact that human operators play a crucial role in the resolution of cyber security incidents [7]. Simple configuration mistakes that leave systems unprotected are often uncovered only after security incidents. On the other hand, human decisions can make the difference between a complete breakdown and system survival.

The interaction of human operators with critical infrastructures is mostly implemented using information and communications technologies, largely due to reduced costs and greater efficiency, flexibility and interoperability. Consequently, several approaches have focused on the design of (graphical) interfaces, also known as human-machine interfaces (HMIs), that assist decision making processes and reduce the reaction times of human operators [12, 18]. On the other hand, human operators can also interact with a system independently of HMIs, for example, by switching a device on or off. Therefore, testbeds that focus on the analysis of critical infrastructures should also take into account the presence of human operators.

This paper argues that the presence of human operators and HMIs dramatically changes system behavior and should be taken into account when designing testbeds for analyzing critical infrastructures. Existing testbeds (see, e.g., [5, 11, 13, 21]) may engage human operators and real HMIs, but they do not include software simulations of these components. Although testbeds with human operators and real HMIs provide reliable experimental data, they are unable to support exhaustive parameter testing. This is mainly due to the costs involved in acquiring and training human operators, and the costs of customizing proprietary HMI software.

Recognizing the importance of human operators and, especially, operating procedures in security experiments, we propose an extension to our previously developed experimentation framework [6]. The extension incorporates "human decision" units that can run human operator and HMI models in real time. Actions issued by the models are translated into commands that are executed in the cyber and physical realms. This way, the extended framework enables the recreation of realistic scenarios in which operators interact with the cyber realm and also execute actions in the physical realm. The approach is evaluated by assessing the impact of human operator reactions during an attack on a cyber-physical infrastructure incorporating the IEEE 30-bus power grid model.

2. Related Work

Most of the critical infrastructure experimentation testbeds in use today do not take into account human operator or HMI models. In contrast, several testbeds have been developed that model the interactions of human operators with a physical process through real HMIs. The most relevant approaches from both these categories are discussed in this section.

Chabukswar, *et al.* [3] used the Command and Control WindTunnel [14] multi-model simulation environment based on the IEEE high-level architecture standard [19] to facilitate interactions between simulation engines. They used OMNeT++ to simulate a network, and Matlab Simulink to build and run the physical process model. In this approach, neither the human nor the HMI were considered because the main focus of the testbed was to recreate critical infrastructures and hardware control loops.

Hopkinson, *et al.* [9] adopted a similar approach, in which a PowerWorld server (a high-voltage power system simulation and analysis package) [15], was used to provide a simulation environment for electrical power systems and the ns-2 network simulator was used to simulate other system components (e.g., programmable logic controllers (PLCs) and malware). Like the work of Chabukswar, *et al.*, this approach also does not consider human operators or HMI models. Nevertheless, it provides a generic interface for extending the testbed with additional ns-2 modules. Although the approach could support the integration of external modules, it was not designed to run complex mathematical models such as those developed using dedicated modeling tools like Simulink. Furthermore, designers would have to define "glue" code to enable the integration of a wide range of models with ns-2. These aspects are the main focus of the work described in this paper – among other things, glue code consisting of several modules and interfaces is proposed to integrate human operator and HMI models in a previously developed critical infrastructure experimentation testbed.

In contrast with the two approaches mentioned above, several testbeds (e.g., [5, 11, 13, 21]) incorporate "real" humans and HMIs – specifically, human operators interact with real proprietary HMIs. Although such testbeds provide reliable experimental data, they are unable to support exhaustive parameter testing involving human operators and HMIs. Queiroz, *et al.* [16] have implemented just such an approach. In their testbed, HMIs are simulated as OMNeT++ modules and human operators interact with the simulated HMIs. Although this approach represents an advancement over the approaches described above, it still requires human operators to be present and interact with the system. Ultimately, the fidelity of such an approach is counterbalanced by its higher costs and lower efficiency because experiments might require the presence of multiple humans to perform repetitive tasks in order to cover the entire parameter space.

3. Problem Statement

In many fields, there is an increasing trend to replace humans with automated control loops. Nevertheless, human operators continue to play a significant role in the operation of critical infrastructures, especially during abnormal situations and contingencies.

Most critical infrastructure experimentation testbeds available today (see, e.g., [5, 11, 13, 21]) engage real human operators and HMIs, not simulations of human operators and HMIs. Human operator modeling (see, e.g., [1, 4, 8, 10, 17, 23]) is a complex task; in fact, research on designing human operator models has been around since the beginning of the 20th century [1, 4]. Recent research has demonstrated the applicability of linear and nonlinear control theories [10] and belief-desire-intention paradigms combined with agent-based platforms [17, 23] in the human operator modeling process. Each approach comes with its own advantages and disadvantages, complicating the task of selecting an approach. Therefore, the design of a generic experimentation platform with human operator and HMI models that could be applied to a wide range of critical infrastructures is not a trivial task.

Taking into account the complexity and diversity of critical infrastructures and the operator models needed for each of these systems, a single operator model that is applicable to all systems may not be possible, let alone feasible. A solution that would require the integration of every operator model from scratch would also be infeasible. A more reasonable solution would provide designers with several interfaces that allow a wide variety of operator models to be adapted, coupled and integrated into a testbed. The solution should also enable the coupling and integration of various HMI models because HMI software has a major impact on the state of a critical infrastructure.

It is also important to take into account the fact that large critical infrastructures have many human operators who supervise and control systems at any given time. A generic representation of the human decision making process should include hierarchical and graph-based information flows. These complex interactions should not be ignored or the applicability of the approach would be greatly limited.

Another important issue to be considered is the translation of simulated operator decisions to actions in the cyber-physical realm. These actions include interactions with the physical process as well as interactions in the cyber realm (e.g., configuring a firewall, launching an external script and shutting down a computer). The translation process should be flexible and should rely on generic modules that are easily replaceable. In this way, the implemented solution would support experimentation with a wide range of physical processes and networked industrial control systems.

Extending existing critical infrastructure experimentation testbeds with human operator and HMI models is a complex task. But it is important because, when human operator and HMI models are added, the experimentation environment is able to support the human-in-the-loop paradigm that plays a crucial role in the outcome of any cyber security experiment involving critical infrastructures.

4. Proposed Approach

This section presents our approach for integrating human decision making into our previously developed critical infrastructure experimentation framework [6]. The section begins with a brief overview of the experimentation framework and proceeds to describe the extension.

4.1 Experimentation Framework

The experimentation framework developed in our previous work [6] follows a hybrid approach, where the Emulab-based testbed recreates the control and process network of a networked industrial control system, including PLCs and SCADA servers, and a software simulation reproduces the industrial process. The architecture shown in Figure 1 has three layers: (i) cyber layer; (ii) physical layer; and (iii) link layer, which lies in between the cyber and physical layers. The cyber layer includes various information and communication technology

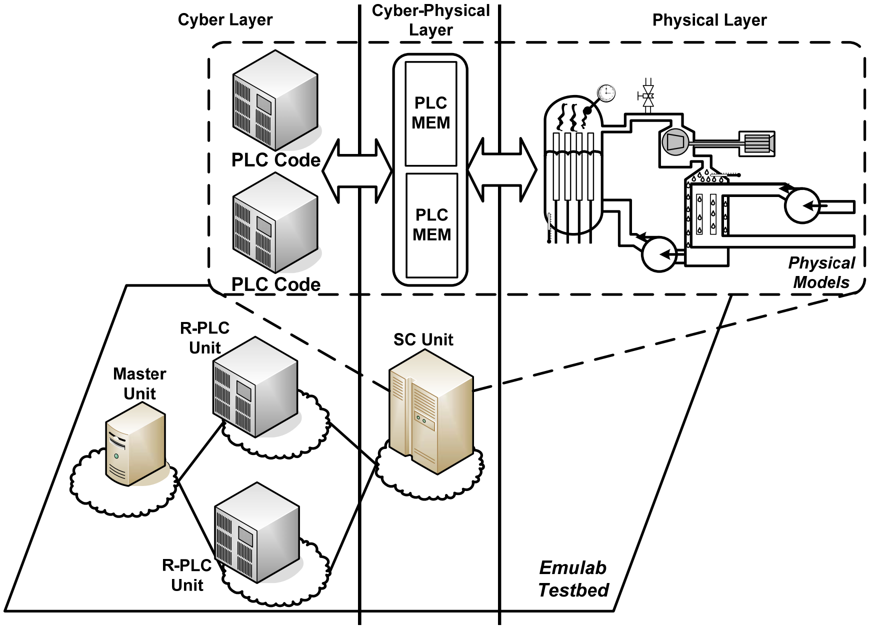


Figure 1. Experimentation framework.

components that are used in SCADA systems, while the physical layer incorporates simulations of physical processes and devices. The link layer (cyber-physical layer) provides the glue between the two layers through the use of a shared memory region.

The physical layer is recreated using a soft real-time simulator that runs on the simulation core (SC unit) and executes a model of the industrial process. The cyber layer is recreated by an emulation testbed that uses the Emulab architecture and software [22] to automatically and dynamically map physical components (e.g., servers and switches) to a virtual topology.

In addition to the process network, the cyber layer also includes control logic code that is executed by PLCs in a real-world system. The control code can be run sequentially or in parallel with the physical model. In the sequential case, tightly coupled code (TCC) is used; this code runs on the SC unit in the same memory space as the model. In the parallel case, loosely coupled code (LCC) is used; this code runs in another address space, possibly on another host in the remote PLC (R-PLC) unit. The main advantage of TCC is that it does not miss values generated by the model between executions. On the other hand, LCC allows the remote execution of PLC code, the injection of malicious code without stopping model execution, and the execution of complex PLC emulators. A master unit implements global decision algorithms based on the sensor values received from R-PLC units.

The cyber-physical layer incorporates PLC memory (as a set of registers typical of PLCs) and the communications interfaces that glue the other two layers. Memory registers provide links to inputs (e.g., valve position) and outputs (e.g., sensor values) of the physical model.

Prototypes of SC, R-PLC and master units were written in C# (Windows) code, which was ported and tested on Unix-based systems (FreeBSD, Fedora and Ubuntu) using the Mono platform. Matlab Simulink is used as the industrial process simulator (physical layer). Matlab Real Time Workshop is used to generate C code from the Simulink models. Communications between SC and R-PLC units are handled by a .NET binary implementation of RPC over TCP (referred to as "remoting"). Communications between the master and R-PLC units use the Modbus TCP protocol.

4.2 Extended Architecture

As mentioned above, adding human decision making to an experimentation testbed offers several advantages. However, the complexity of modeling human decision making requires an approach that does not limit the testbed to one specific model. Therefore, we propose a generic approach for integrating models in cyber-physical testbeds. In this approach, models are seen as "black boxes" that must implement a standard interface in order to interact with other system components. The required interface includes a set of inputs and a set of outputs that are connected to "action scripts" at run time. All the actions issued or received by models are sent through action scripts that include the code necessary for processing actions and communicating with other software components.

To implement the proposed approach, we extended the framework developed in our previous work with a generic human decision (HD) unit. In the remainder of this section, we describe the prototype implementation in detail and provide insight into some of the key aspects of the implementation.

The design of the HD unit started with the assumption that human operators interact with cyber-physical systems in several ways. Human operators certainly rely on information and communications hardware and software present in networked industrial control system installations (e.g., HMIs). However, they also interact independently of these components via installation-specific components (e.g., customized script for configuring a firewall) and physical actions (e.g., shutting down a server). The HD unit architecture takes all these aspects into account through the actions (glue) layer and through action scripts that implement the specifics of each experiment conducted using the testbed.

Figure 2 presents the extended architecture, which includes an HD unit and other components typical of networked industrial control systems. Within the HD unit, human operator and HMI models interact with the cyber-physical realm through an action module that enables bidirectional communications and command execution (i.e., actions). The human operator and HMI models shown in Figure 2 interact in real time using direct connections that are included in the architecture. Although the human operator and HMI models are con-

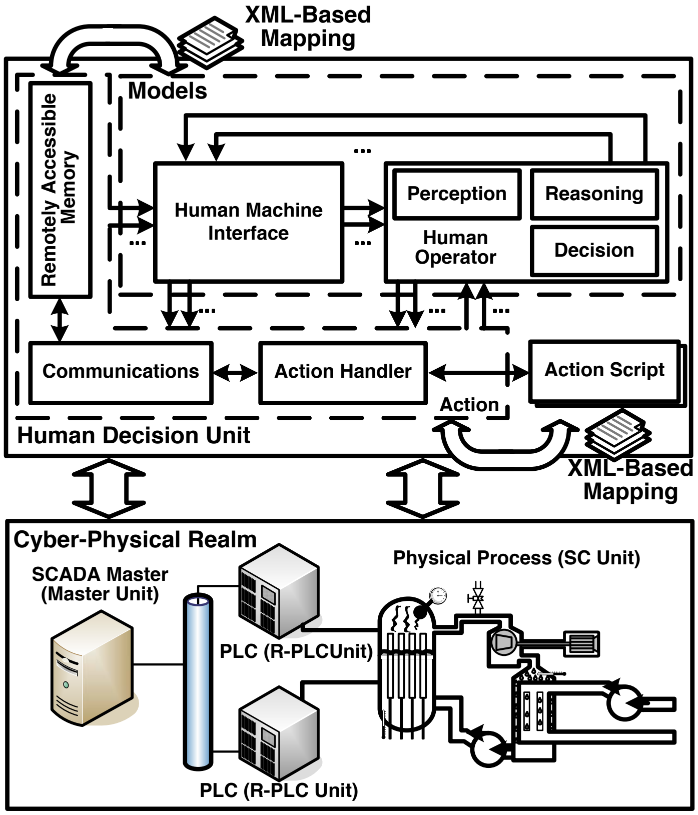


Figure 2. Extended architecture with a human decision unit.

structed according to the specifications of an experiment, three building blocks are included in the human operator model for completeness: (i) perception; (ii) reasoning; and (iii) decision. These basic human operator functionalities can be taken into account when building similar models. Also, depending on the requirements of an experiment, they can be replaced with other blocks if necessary.

As shown in Figure 2, human operator models receive events from the action module directly and through HMI models. The first case represents the direct interaction of human operators with the cyber-physical realm, while the second case represents interactions through the HMI. In both cases, the inputs are the data used by the models in each time step (e.g., measured voltage) while the outputs are the actions to be executed. Each action includes an identifier and several parameters (e.g., open or close a valve). These are written to the remotely accessible memory module from where they are read by the action handler module. Then, based on external XML configuration files, the action handler module runs a specific action script identified by a numeric identifier.

Action scripts are written in the C programming language and are loaded as external binary libraries. These can be specific to each experiment and may include commands that pass values to other components of the framework (e.g., physical model) or commands that launch additional scripts (e.g., configuring a firewall or turning off a machine). This way, the HD unit provides a flexible approach to translate model-specific outputs to real actions that affect the physical and cyber realms.

The remotely accessible memory module is not limited to passing values between the internal modules of the HD unit; it also serves as a way for external components to interact with the human operator and HMI models. By enabling remote access to this memory region, external software components (e.g., other HD units and SCADA master units) can communicate with the HD unit using simple memory access operations. In these cases, requests are received and processed by the communications module, which passes the received values to the remotely accessible memory module, from where they are read by the models module and provided to the human operator and HMI models. To increase flexibility, external XML configuration files are used to map memory regions to model inputs and to map model outputs to memory regions.

As in our previous work, we implemented a prototype of the HD unit in C# (Windows) and ported it to Unix-based systems using the Mono platform. Currently, communications with other HD and master units are handled by the Modbus protocol, as this protocol was specifically designed for exchanging memory-mapped data between units. However, other protocols can be implemented simply by replacing the Modbus handler units.

Human operator and HMI models were implemented in Matlab Simulink, a general simulation environment for dynamic and embedded systems. Its toolboxes (e.g., control systems and neural networks) provide powerful support for modeling and simulation. Matlab RTW was used to generate C code from the Simulink models. The generated code was then integrated into the extended framework using an XML configuration file. Thus, the model is able to interact in real time with the other system components.

5. Case Study

Human operators play major roles in real critical infrastructure installations and, therefore, cannot be ignored in experimentation testbeds. The HD units address this issue by bringing new elements that implement the human-in-the-loop paradigm in existing testbeds. This section presents experimental results that demonstrate the applicability of the proposed approach and the importance of human operators in a case study involving cyber attacks on a simulated power grid.

The power grid model employed in the experiment was the well-known IEEE 30-bus test system, which includes six power generators and twenty loads distributed on twenty buses. The overall model was divided into four regions,

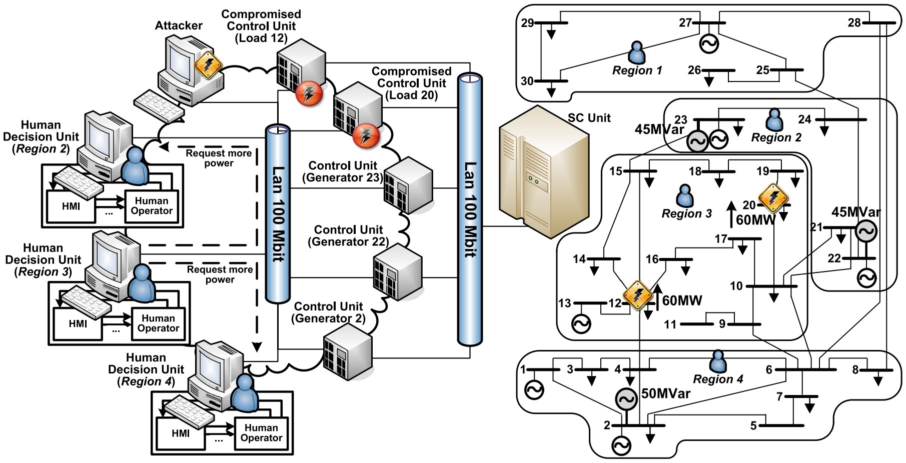


Figure 3. Experimental setup and test system regions.

each controlled by different human operators. One of the main goals of the experiment was to show that disturbances can be balanced by countermeasures taken by power grid operators. The results also confirm the fact that, in an interconnected power grid, operators from different regions must collaborate to restore the grid to normal conditions after a disturbance.

5.1 Experimental Scenarios

Figure 3 presents a schematic diagram of the IEEE 30-bus power grid model. The grid is divided into four regions that minimize the number of connections (i.e., transmission lines) between the regions. In a real-world environment, the grid may be divided into regions based on other factors (e.g., balanced power and loads). However, our focus is not on defining regions, but on recreating a possibly realistic scenario with multiple operators, who have to cooperate in order to keep the power grid within its normal operating limits.

In our scenario, we assume that an adversary can compromise the information and communications technology infrastructure in Region 3 using social engineering and also exploit vulnerabilities in the SCADA protocols [2]. Then, the same adversary employs a coordinated worm-based attack to trigger synchronized events in control devices. The attacks turn the power on to large consumers at the same time at Substations 12 and 20 (120 MW total), which causes a disturbance that drops the voltages below their operating limits of 0.95 p.u. In a real-world environment, these attacks could damage hardware and cause blackouts, and possibly have cascading effects that could spread throughout the power grid.

In our scenario, we also assume that the operators in Region 3 can request assistance from operators in other regions when they lose control of their infrastructure. Upon receiving the request, operators in Regions 2 and 4 inject additional power into the grid with the goal of stabilizing the voltages in Region 3.

The following three cases are considered in our scenario:

- **Case 1:** Operators in Region 3 lose control of their infrastructure and do not request assistance from operators in other regions. The voltages drop well below their normal operating limits and the operators have to react quickly in order to prevent serious damage to physical devices.

- **Case 2:** Operators in Region 3 lose control of their infrastructure and request assistance from operators in Region 4. This case shows that a disturbance originating in Region 3 can be addressed by measures taken in other regions. The operators in Region 3 request their counterparts in Region 4 to increase energy production and the power supplied to the grid. Upon receiving the request, the operators in Region 4 start their back-up generators and increase production by 50 MVar. Although the effects are limited, this case demonstrates the importance of operator collaboration.

- **Case 3:** Operators in Region 3 lose control of their infrastructure and request assistance from operators in Regions 4 and 2. The actions taken in Case 2 produce limited effects, so additional assistance is requested from operators in Region 2. Upon receiving the request, the operators in Region 2 increase the energy production by 90 MVar. This action stabilizes the bus voltages in Region 3.

5.2 Experimental Setup and Models

The critical infrastructure protection scenario was implemented in the Experimental Platform for Internet Contingencies (EPIC) Laboratory at the EU Joint Research Centre [20]. The Emulab testbed included nodes with the following configuration: FreeBSD OS 8, AMD 2.3 GHz Athlon Dual Core CPU and 4 GB RAM. Figure 3 shows the experimental setup.

Models were developed using the "black-box" approach described in the previous sections. A Simulink model was developed for each operator with inputs and outputs connected to action scripts.

The operator model for Region 3 has one input (link status) and three outputs (action identifier and two parameters for the request for assistance). The input action script continuously tests the link between the HD unit and the physical process (SC unit). When the link is "on," the action script sends a value of "1" to the model, and "0" otherwise. The model takes this input and forwards its negated value to the output along with the action identifier. The output action script takes these two values and forwards them to the HD units running in Regions 2 and 4. In the implementation, a value of "1" means that operators in Region 3 need assistance. The implemented functions are different for each case:

$$\begin{aligned}\text{Case 1:}\ & f_1(x) = (ID, 0, 0)\\ \text{Case 2:}\ & f_2(x) = (ID, !x, 0)\\ \text{Case 3:}\ & f_3(x) = (ID, !x, !x)\end{aligned}$$

where x is the model input and ID is the action identifier.

The operator model for Region 4 has one input (status of assistance request ("0" or "1")) and two outputs (action identifier and one parameter denoting the power injected by back-up generators (50 MVar)). The model receives its input value from the HD unit in Region 3 and produces an output that is sent to the SC unit and finally to the physical process. The mathematical function for this model is:

$$f(x) = (ID, (x = 1)?\ 50 : 0)$$

where x is the model input and ID is the action identifier.

The operator model for Region 2 is similar to that for Region 4 in that it takes the same input but produces one additional output. The first output is the action identifier, while the second and the third outputs are the MVars produced by two back-up generators. Our scenario uses two back-up generators in Region 2 to produce a total of 90 MVar (45 MVar per generator). The mathematical function for this model is:

$$f(x) = (ID, (x = 1)?\ 45 : 0, (x = 1)?\ 45 : 0)$$

where x is the model input and ID is the action identifier.

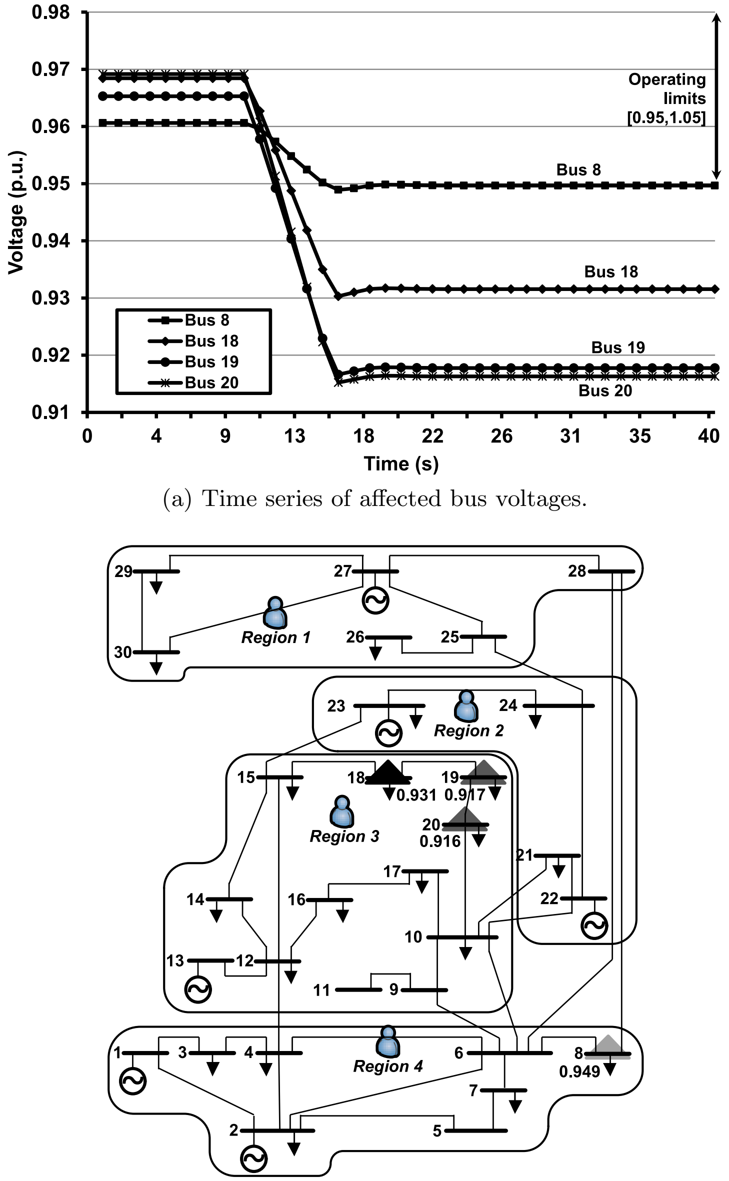
5.3 Experimental Results

This section presents the results obtained for the three cases.

- **Case 1:** Immediately after the attack is launched on Buses 12 and 20, the voltages begin to drop. The voltages of the Buses 18, 19 and 20 fall to almost 0.91 p.u., well below the operating limit of 0.95 p.u. The disturbance also propagates to Region 4, where it causes a voltage drop on Bus 8 to 0.949 p.u., slightly below the operating limit. This effect is also shown in Figures 4(a) and (b) where, without any operator intervention, the disturbance causes severe voltage changes, mostly in Region 3.

- **Case 2:** In this case, we assume that the operators in Region 3 are able to obtain assistance from their counterparts in Region 4, where an additional back-up generator injects 50 MVar into the grid. This action has a significant effect on Bus 8, where the voltage increases above the operating limit. However, the effects are not as significant on the other buses, where the voltages remain below 0.95 p.u. (Figure 5).

- **Case 3:** In Case 3, the operators in Region 3 request the assistance of operators in Regions 4 and 2. The operators in Region 4 inject 50 MVar

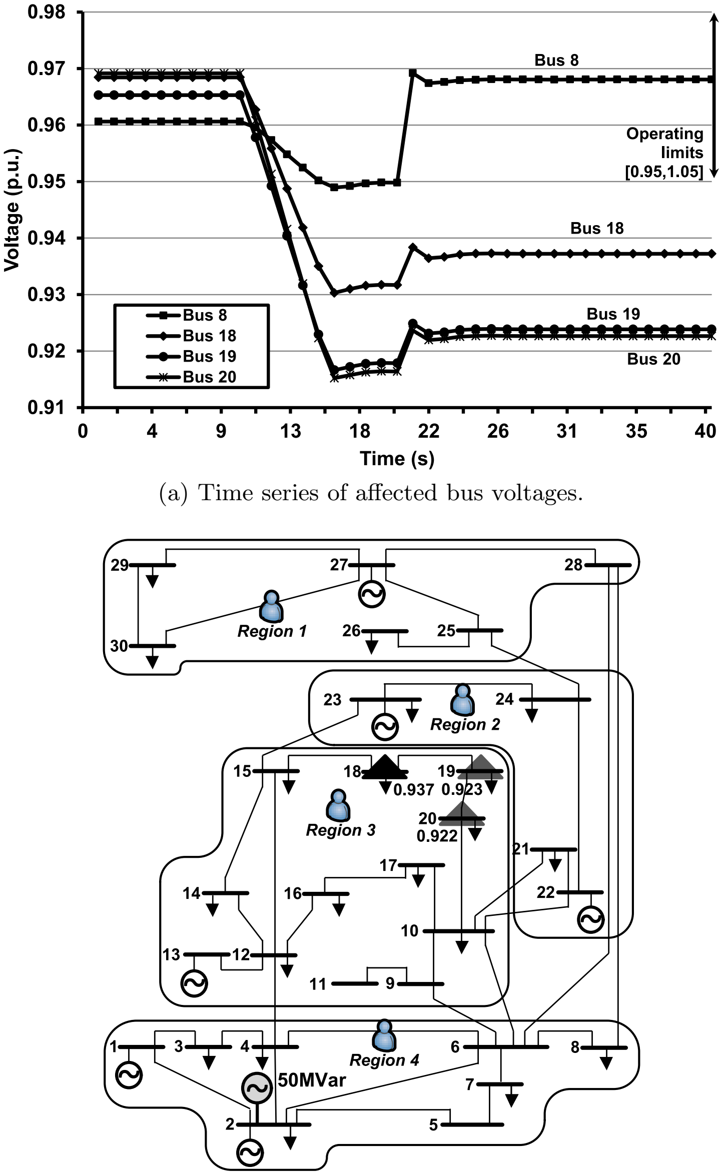


(a) Time series of affected bus voltages.

(b) Regional view.

Figure 4. Effect of the attack in Case 1.

of additional power using a back-up generator and the operators in Region 2 start two back-up generators, which inject an additional 90 MVar of power into the grid. As shown in Figure 6, this immediately increases

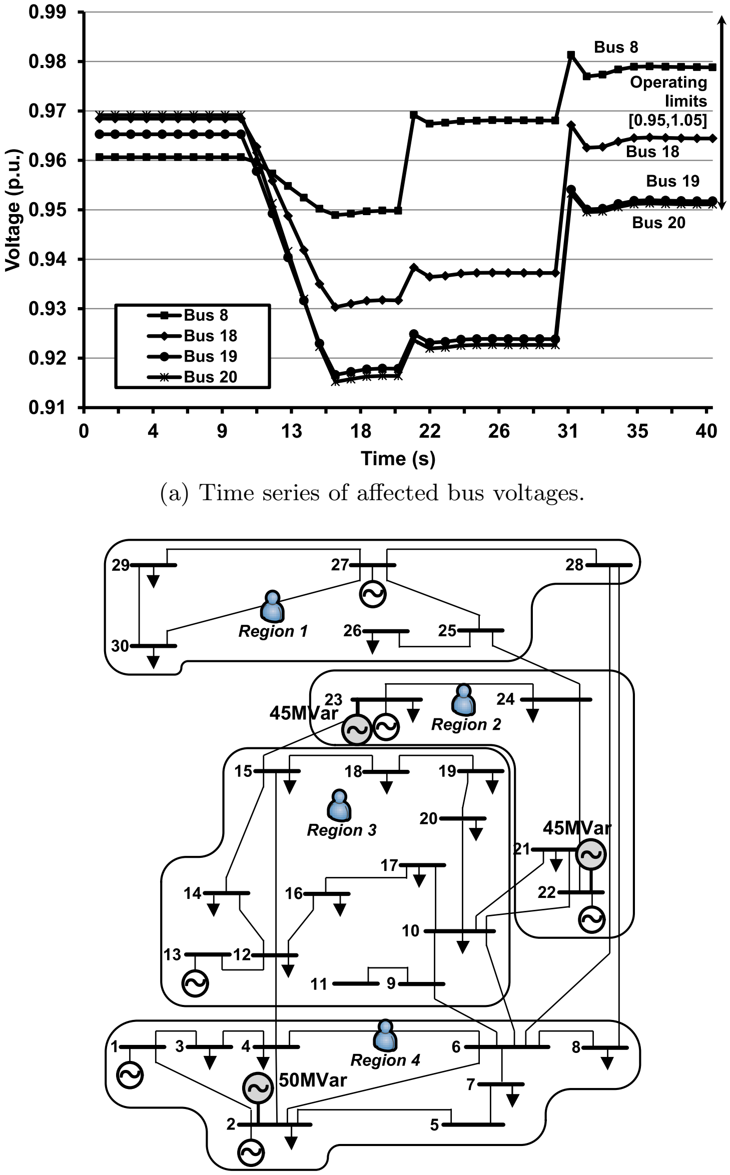


(a) Time series of affected bus voltages.

(b) Regional view.

Figure 5. Effect of the attack in Case 2.

the voltages above the operating limits. Consequently, although the operators in Region 3 lose control of their infrastructure, they are eventually



(a) Time series of affected bus voltages.

(b) Regional view.

Figure 6. Effect of the attack in Case 3.

able to ensure stability in their region by cooperating with operators in neighboring regions.

The results confirm the fact that human operators are indispensable elements of security studies involving critical infrastructures. Upon comparing the results for Case 3 with those for Cases 1 and 2, it is clear that operators in a highly interconnected power grid must collaborate in order to balance regional disturbances and ensure grid stability. The results also demonstrate that the proposed approach can recreate complex scenarios involving the cyber and physical realms, as well as the decision making of human operators. The approach thus implements the important human-in-the-loop paradigm that is a significant aspect of real-world critical infrastructure environments.

6. Conclusions

Human operators play important roles in supervising and controlling modern critical infrastructures. Although industry may be moving towards fully automated control loops, human operators are indispensable during abnormal situations and contingencies. The principal contribution of this paper is an extended experimentation framework that provides generic human decision units that help integrate human operator and HMI models. The integration is based on a "black-box" approach where, as long as generic models implement well-defined interfaces with sets of input and output signals, the models can be integrated in the experimentation framework regardless of their content. The case study involving the IEEE 30-bus power grid model demonstrates the utility of the extended experimentation framework. The results of the case study also confirm that, in large-scale interconnected critical infrastructures involving multiple operators, it is crucial that operators cooperate to ensure the global stability of the infrastructures. Therefore, modern testbeds must provide support for modeling and analyzing the behavior of multiple human operators in complex critical infrastructure scenarios.

Our future research will explore the complexity of human operator networks and the vulnerabilities of critical infrastructures that rely on information exchange. This will help recreate and analyze complex multi-sector scenarios where one critical infrastructure could have cascading effects on other critical infrastructures.

References

[1] G. Bekey and C. Neal, Identification of sampling intervals in sampled-data models of human operators, *IEEE Transactions on Man-Machine Systems*, vol. 9(4), pp. 138–142, 1968.

[2] E. Bompard, P. Cuccia, M. Masera and I. Nai Fovino, Cyber vulnerability in power systems operation and control, in *Critical Infrastructure Protection (LNCS 7130)*, J. Lopez, R. Setola and S. Wolthusen (Eds.), Springer-Verlag, Berlin Heidelberg, Germany, pp. 197–234, 2012.

[3] R. Chabukswar, B. Sinopoli, G. Karsai, A. Giani, H. Neema and A. Davis, Simulation of network attacks on SCADA systems, presented at the *First Workshop on Secure Control Systems*, 2010.

[4] K. Craik, Theory of the human operator in control systems, *British Journal of Psychology*, vol. 38(2), pp. 56–61, 1947.

[5] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye and D. Nicol, SCADA cyber security testbed development, *Proceedings of the Thirty-Eighth North American Power Symposium*, pp. 483–488, 2006.

[6] B. Genge, I. Nai Fovino, C. Siaterlis and M. Masera, Analyzing cyber-physical attacks on networked industrial control systems, in *Critical Infrastructure Protection V*, J. Butts and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 167–183, 2011.

[7] A. Gheorghe, M. Masera, M. Weijnen and L. De Vries, *Critical Infrastructures at Risk: Securing the European Electric Power System*, Springer, Dordrecht, The Netherlands, 2006.

[8] R. Harris, J. Kaplan, C. Bare, H. Iavecchia, L. Ross, D. Scolaro and D. Wright, Human Operator Simulator (HOS) IV User's Guide, Research Product 89-19, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia, 1989.

[9] K. Hopkinson, K. Birman, R. Giovanini, D. Coury, X. Wang and J. Thorp, EPOCHS: Integrated commercial off-the-shelf software for agent-based electric power and communication simulation, *Proceedings of the 2003 Winter Simulation Conference*, vol. 2, pp. 1158–1166, 2003.

[10] T. Ivancevic and B. Jovanovic, Human operator modeling and Lie-derivative based control (`arxiv.org/pdf/0907.1206.pdf`), 2009.

[11] M. McDonald, G. Conrad, T. Service and R. Cassidy, Cyber Effects Analysis Using VCSE: Promoting Control System Reliability, Technical Report SAND2008-5954, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, 2008.

[12] F. Moussa, C. Kolski and M Riahi, A model based approach to semi-automated user interface generation for process control interactive applications, *Interacting with Computers*, vol. 12(3), pp. 245–279, 2000.

[13] I. Nai Fovino, M. Masera, L. Guidi and G. Carpi, An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants, *Proceedings of the Third Conference on Human System Interaction*, pp. 679–686, 2010.

[14] S. Neema, T. Bapty, X. Koutsoukos, H. Neema, J. Sztipanovits and G. Karsai, Model-based integration and experimentation of information fusion and C2 systems, *Proceedings of the Twelfth International Conference on Information Fusion*, pp. 1958–1965, 2009.

[15] PowerWorld Corporation, Champaign, Illinois (`www.powerworld.com`).

[16] C. Queiroz, A. Mahmood, J. Hu, Z. Tari and X. Yu, Building a SCADA security testbed, *Proceedings of the Third International Conference on Network and System Security*, pp. 357–364, 2009.

[17] A. Rao and M. Georgeff, BDI agents: From theory to practice, *Proceedings of the First International Conference on Multi-Agent Systems*, pp. 312–319, 1995.

[18] R. Reeder and R. Maxion, User interface defect detection by hesitation analysis, *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 61–72, 2006.

[19] X. Shi and Q. Zhong, The introduction of high level architecture (HLA) and run-time infrastructure (RTI), *Proceedings of the Society of Instrument and Control Engineers Annual Conference*, vol. 1, pp. 1136–1139, 2003.

[20] C. Siaterlis, A. Garcia and B. Genge, On the use of Emulab testbeds for scientifically rigorous experiments, to appear in *IEEE Communications Surveys and Tutorials.*

[21] C. Wang, L. Fang and Y. Dai, A simulation environment for SCADA security analysis and assessment, *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, pp. 342–347, 2010.

[22] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb and A. Joglekar, An integrated experimental environment for distributed systems and networks, *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*, pp. 255–270, 2002.

[23] X. Zhao, J. Venkateswaran and Y. Son, Modeling human operator decision-making in manufacturing systems using BDI agent paradigm, presented at the *IIE Annual Conference and Exposition*, 2005.