Mobile Communications-Enabled Smart Grid Cosimulator System Design

Kevin Shimotakahara, *Student Member, IEEE*, Medhat Elsayed, *Student Member, IEEE*, Melike Erol-Kantarci, *Senior Member, IEEE*, and Karin Hinzer, *Senior Member, IEEE*

***Abstract*—In this article, we present a system design approach for a simulator constructed in MATLAB/Simulink used to simulate power systems and the underlying communication system, i.e., device-to-device (D2D) communication protocol stack of long-term evolution (LTE). Emphasis is placed on the characterization of the continuous time-discrete event (CT-DE) cosimulation interface used to develop this simulator. We model the mapping of communication links between D2D nodes as a bipartite digraph, and proceed to characterize the dynamics of information exchange between power system signals and LTE nodes using the discrete event specification formalism. The joint CT-DE dynamics were tested with a simple fault location, isolation, and restoration application. The results indicated that the control signal shifts in the power system simulator were synchronized with the appropriate LTE messaging dynamics.**

***Index Terms*—Cosimulation, discrete event specification (DEVS), device-to-device (D2D), long-term evolution (LTE), MATLAB, smart grid, Simulink.**

I. Introduction

POWER systems around the world continue to become modernized with the digital communication network overlays that improve the performance of energy markets, grid stability, and integration of renewable energy sources [1]–[5]. This ongoing technological procession has lead to the emergence of simulation tools that provide designers of smart grid technology with a safe and cost-effective design space for their innovations. Nonetheless, there is little work done in developing a smart grid simulator solely in the MATLAB/Simulink environment despite its powerful software packages for simulating the long-term evolution (LTE) physical layer, discrete event dynamics, and conventional power systems. It has been a common approach in the literature to use Simulink for both power engineering research [6]–[12] and hybrid continuous time-discrete event (CT-DE) cosimulations [13].

Among the few works, Elkhorchani and Grayaa [14] tested their proposed smart grid wireless communication architecture on top of their power-flow model of a renewable-integrated power system in the MATLAB/Simulink environment. The authors modeled ZigBee, WIMAX, and WiFi physical layer communication protocols. Shlebik *et al.* [15] simulates a power line communication (PLC)-enabled advanced metering infrastructure application in MATLAB, modeling the transmission of smart meter data via various modulation schemes. Moreover, despite the wide choice of protocols and architectures that can be simulated with the network simulators used by state-of-the-art smart grid cosimulators [16], such simulators (e.g., ns-3 and OPNET) do not offer a high fidelity characterization of the dynamics present at the physical layer of a protocol stack. More specifically, such network simulators are considered “layer 2 and up” simulators, meaning that they simplify physical layer (“layer 1”) dynamics, e.g., fading effects, physical resource allocation, etc.

We have made a new smart grid cosimulation tool that has the advantage of being capable of simulating the propagation of time-domain electromagnetic waveforms that have been modulated with digital information compliant to LTE specifications, making highly detailed analysis on effects of signal attenuation, fading, and interference possible. This has enabled us to study the impact of the physical layer of the mobile communications network on the performance of smart grid applications in our previous work in [17]. We have further focused on validating our cosimulation interface design in testing a simple fault location, isolation, and restoration (FLISR) application in [18]. This was an essential step for encouraging practitioners to consider utilizing our hybrid simulation approach. We desired to promote the legitimacy of this approach as a simple and effective way for researchers to integrate their custom DE-based simulation code with a CT system built in Simulink. Our approach acts as an alternative to developing elaborate APIs for simulation unit interfacing compliant with leading standards such as high-level architecture and functional mockup interface cosimulation frameworks, whose brief descriptions can be found in [16]. These standards are useful for containerizing simulation units and transferring them to a new cosimulation environment as done by a NASA project shown in [19], or developing a framework that can be used to implement an arbitrarily sophisticated cosimulation orchestration mechanism [20]. On the other hand, our approach performs CT-DE integration in a simpler way. The codes of the simulator are made available online [21]. The

Manuscript received January 16, 2020; revised July 15, 2020 and September 8, 2020; accepted September 10, 2020. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant 201703995, in part by the Collaborative Research and Training Experience Program under Grant 497981, and in part by the U.S. National Science Foundation under Grant CNS-1647135. *(Corresponding author: Melike Erol-Kantarci.)*

The authors are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa K1N 6N5, Canada (e-mail: kshim017@uottawa.ca; melsa034@uottawa.ca; melike.erolkantarci@uottawa.ca; khinzer@uottawa.ca).

Digital Object Identifier 10.1109/JSYST.2020.3025202

key contribution of this article is that it builds on the initial work of [18], and provides a mathematical characterization of our cosimulator's joint CT-DE dynamics, which enables this approach to be generalized beyond the software platforms used. This article also includes a detailed set of simulation results obtained through the cosimulator. This article, along with the provided source codes, makes future contributions from the community possible.

A. Contributions

The main goal of this article is to provide a system design perspective of our cosimulator's CT-DE dynamics and its Simulink–MATLAB interface. The purpose of this approach is twofold. First, being able to characterize the joint dynamics between our power system simulator (CT based) and LTE network simulator (DE based) with mathematical equations that generalizes our solution to the problem of merging such simulation units. The consequence is that future practitioners can refer to our generalized system design as a guide to implementing their own custom cosimulation of some continuous time system coupled with a communication network that uses a time-slot-based medium access control (MAC) scheme. Note that, in our cosimulator, we have chosen LTE as the communication facilitator, however, any communication technology can be integrated using our methodology.

Second, a mathematical characterization of our simulator's Simulink–MATLAB interface unambiguously describes the dynamics of our cosimulation orchestration mechanism, providing clarity and reassurance. The main concept behind our orchestrator is that it combines the ideas of fixed point and event-based synchronization methodologies, where fixed points in time are modeled as "time advance events." In a master–slave architecture, the power system simulator (master) can send time advance events to the communication network simulator (slave) while still being able to asynchronously update message queues when smart grid messages are generated in response to threshold crossing events of CT signals in the power system.

The remaining sections of this article are organized as follows: Section II provides an overview of the MATLAB/Simulink implementation model, and characterizes the power system and communications network mathematically. Section III reviews the FLISR application devised for testing the simulator. Section IV presents the test results, showing successful execution of the FLISR application on the power system side, with its control signal actuation timing being consistent with that of the messaging dynamics observed in the LTE network. Finally, Section V concludes this article.

II. Integrated Simulator Model

A. Software Architecture

The integrated simulator combines the LTE System Toolbox, Simscape Power Systems, and SimEvents software packages that are available in the MATLAB/Simulink environment. In addition to using these software packages, original code was developed to model the relevant 4G LTE protocol layers that the LTE System Toolbox does not support.

Fig. 1 provides a high-level system diagram of the simulation platform. In the Simulink environment, both Simscape Power Systems and SimEvents discrete event simulation tools are used. Simscape provides easy to use block components for power system devices, modeling their dynamics and providing the user with a simple interface to modify their characteristics. Depending on the state of the Simscape power system, the Simulink solver and/or sensors monitoring state variables may trigger an event in the SimEvents domain (Steps 1(a) and 1(b) in Fig. 1). SimEvents allows to simulate system behavior that responds to intermittently occurring event inputs. For example, such an event could be a timer within a smart meter indicating it is time for the device to report its measurements to another device in the smart grid via a communications message. These events produced in SimEvents proceed to call Simulink functions that in turn evoke the applications of smart grid communication devices, which are implemented as MATLAB scripts (Step 2 in Fig. 1).

In the simulator, smart grid device application scripts construct data packets to send through a communication network in the form of a binary array. This data are passed to the LTE protocol stack (Step 3 in Fig. 1). The LTE protocol stack is simplified in our implementation, only capturing the features relevant to the flow of data through the communication system. As such the packet data convergence protocol (PDCP) layer, which manages header compression and encryption, was ignored for simplicity. However, the radio link control (RLC) layer is implemented in our simulation platform, as it fulfills the key role of concatenating and segmenting internet protocol (IP) packets (technically PDCP packets) buffered by the application into RLC protocol data units (PDUs). The sizes of the RLC PDUs are determined independently from the sizes of the IP packets that are being used to fill the PDU. This process ensures that the data packets being sent over the air interface can be encoded within the interval of frequency spectrum allocated to it for the given transmission time interval it is being sent at.

These RLC PDUs are then given a MAC layer header, which specifies information such as source and destination IDs of sender and receiver in the case of D2D MAC headers, in addition to other pieces of information like the logical channel ID that specifies the logical channel of the data for data plane messages, or the type of control element (CE) being sent for control plane messages. Once MAC headers are added, the data have been refined into transport blocks, and are ready to be input into LTE Toolbox functions, which modulate the binary data into a time-domain waveform (Step 4 in Fig. 1). On the receiving end, the reverse of this process occurs (Steps 5–7 in Fig. 1), ultimately resulting in a different smart grid entity application script receiving a message that results in the occurrence of an event in SimEvents, which in turn causes some form of actuation to be performed within the power system, e.g., a circuit breaker (CB) opening, or the inputs to some power controller changing.

B. Discrete Event Specification (DEVS) Model of ICT-Power System Interface and Its Implementation

The interface between the power system and communication system simulator dynamics can be modeled with the collection of DEVS atomic models [22] characterized in this section. Fig. 2

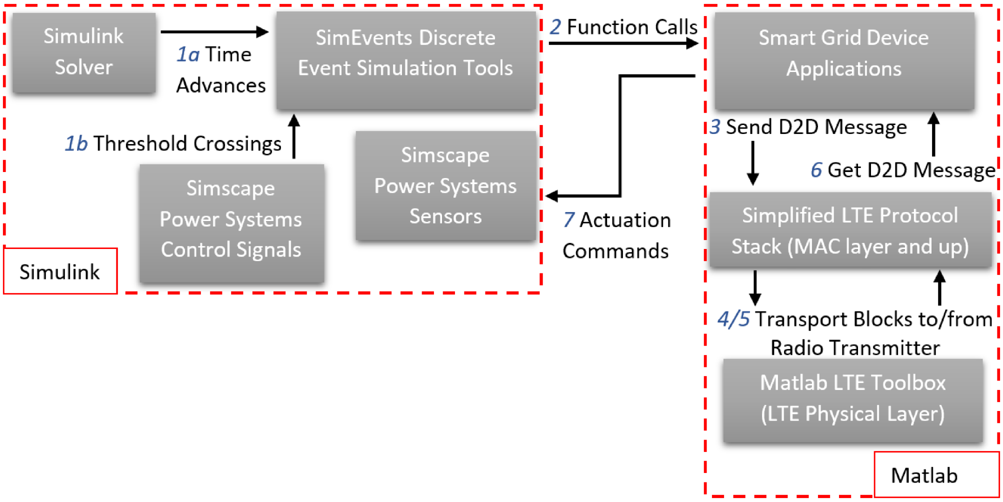

Simulink Solver
1a Time Advances
SimEvents Discrete Event Simulation Tools
2 Function Calls
Smart Grid Device Applications
1b Threshold Crossings
Simscape Power Systems Sensors
Simscape Power Systems Control Signals
Simulink
3 Send D2D Message
6 Get D2D Message
7 Actuation Commands
Simplified LTE Protocol Stack (MAC layer and up)
4/5 Transport Blocks to/from Radio Transmitter
Matlab LTE Toolbox (LTE Physical Layer)
Matlab

Fig. 1. High-level system diagram of a smart grid simulator.

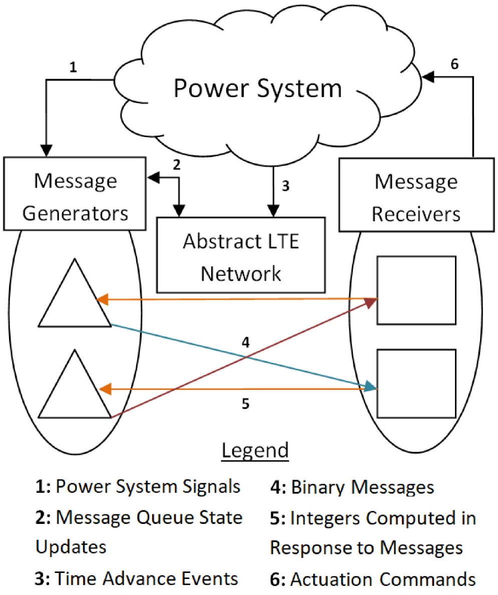


Fig. 2. Input–output coupling diagram of DEVS atomic models.

is a diagram of the interacting DEVS models. For any given D2D-enabled smart grid application, it can be modeled as having N "message generator" nodes that accept power system sensor readings and/or commands from "message receiver" nodes, then generate appropriate communications messages that get queued within the atomic model in response. An "abstract LTE network" block's dynamics is synchronized with the power system, and manages the N message generators, making forwarding and message dropping decisions every subframe the simulation advances. The N generators forward their messages to the appropriate subcollection of N message receiver blocks, which process the incoming message(s) and send outputs to its corresponding message generator and/or power system actuation mechanism as required. The input/output coupling of the message generator and receiver can be modeled as a bipartite digraph, where the message generator nodes have one or multiple unidirectional connections to receiver nodes, and every receiver node has a single unidirectional connection to a single generator node. This graph structure captures how D2D nodes transmit by multicasting messages to a group of receivers, while a D2D node's receiver can only send information to its own transmission entity.

1) Message Generator Atomic Model: The message generator atomic model accepts either instantaneous measurements of power system state variables; signals from a message receiver block; or new queue state and data forwarding instructions from the abstract LTE network atomic model as potential inputs.

Should the input be from a receiver entity or the power system, the message generator generates appropriate binary messages, and stores them in a queue that is expressed as part of the atomic model's state variable. After enqueuing the appropriate messages, the new state of the queue is shared with abstract LTE network atomic model.

Should the input be from the abstract LTE network, the message generator updates the state of its queue with the new input information, and then outputs the first f messages in the updated queue. f is included in the input from the abstract LTE network.

The DEVS tuple is characterized as follows.

1) X: The set of possible inputs is defined in terms of three sets and an "enqueue" label. The first set contains all possible collections of measurements/signals receivable from the power system or message receiver blocks, expressed as Γ, a set of integer m-tuples with n or fewer digits

$$\Gamma = \bigcup_{m=1}^{M} \{x_a | x_a \in \mathbb{Z}, \log(|x|) + 1 \le n\}^m. \quad (1)$$

The second set represents all possible states of a data queue, expressed as Δ, a set of binary word q-tuples, where each word has k or fewer bits

$$\Delta = \bigcup_{q=1}^{Q} \{x_b | x_b = a_1 a_2 ... a_k, a_i \in \{0, 1\}, 1 \le i \le k\}^q \quad (2)$$

m and q represent the number of incoming measurements/signals and the number of messages in the queue, respectively. M and Q are the largest possible values for m and q, respectively.

The third set is the set Θ of natural numbers from 0 to F, where F is the maximum number of messages that the message generator can output to receiver blocks in a single output

$$\Theta = \{x_c | x_c \in \mathbb{N}, x_c \leq F\} \tag{3}$$

X is defined as the union of the Cartesian product of all possible m-tuples and an enqueue label and the Cartesian product of all possible q-tuples and the set of natural numbers that can represent how many messages shall be output

$$X = \Gamma \times \text{enqueue} \quad \bigcup \quad \Delta \times \Theta. \tag{4}$$

2) Y: The set of all possible outputs is defined in terms of three sets and two labels, namely $toLTE$ and $toRXers$. The first set $\Lambda \subset \Delta$ represents all possible f-tuples of messages that can be forwarded to a receiver block

$$\Lambda = \bigcup_{f=1}^{F} \{y_a | y_a = a_1 a_2 ... a_k, a_i \in \{0,1\}, 1 \leq i \leq k\}^f. \tag{5}$$

The second set is all possible values that can represent a message generator's data queue, namely Δ as previously defined. The third set is the set Π of natural numbers from 1 to N, where N is the number of D2D-enabled smart grid nodes in the network

$$\Pi = \{y_b | y_b \in \mathbb{N}, 1 \leq y_b \leq N\}. \tag{6}$$

Y is defined as the union of the Cartesian product of Λ and $toRXers$ and the Cartesian product of Π, Δ, and $toLTE$ as

$$Y = \Lambda \times toRXers \quad \bigcup \quad \Pi \times \Delta \times toLTE. \tag{7}$$

3) S: The state has four substates $s_{\text{ied}} \in S_{\text{ied}}$, $s_q \in S_q$, $s_{\text{out}} \in S_{\text{out}}$, and $s_{\text{app}} \in S_{\text{app}}$. The set of possible states is defined in terms of four sets. The first set consists of three substates that dictate whether the message generator is either idle, enqueuing new data messages, or dequeuing data messages

$$S_{\text{ied}} = \{\text{idle}, \text{enqueuing}, \text{dequeuing}\}. \tag{8}$$

The second set is the set of possible queue states of the message generator, namely Δ

$$S_q = \Delta. \tag{9}$$

The third set is Π, the set of possible numbers of messages that can be forwarded in a single output

$$S_{\text{out}} = \Pi. \tag{10}$$

The fourth set is S_{app}, which is the set of all combinations of smart grid application-specific state variables that characterize the mapping from input signals and measurements to binary messages. Naturally, there is no general way to characterize S_{app}. S is defined as the Cartesian product of all defined substate sets

$$S = S_{\text{ied}} \times S_q \times S_{\text{out}} \times S_{\text{app}}. \tag{11}$$

4) δ_{ext}: The external transition function is defined when the input substate $s_{\text{ied}} = \text{idle}$. This function also makes use of $\text{runapp}_{\text{mode}}()$, mode $\in \{0,1\}$, which models some smart grid application that accepts measurements/signals and returns a set of binary messages to be enqueued for transmission while updating the state variables of S_{app} as required. The details of the smart grid application are the choice of the user of the simulator. For the ease of the characterization of δ_{ext}, a mode subscript is included for this function that allows one to choose whether $\text{runapp}_{\text{mode}}()$ returns new S_{app} values, or the new queue state of the message generator. In reality, this function would do both simultaneously. $\text{runapp}_{\text{mode}}()$ can be characterized in general terms as a mapping from the Cartesian product of X and S to Δ as

$$\text{runapp}_{\text{mode}} : \begin{cases} X \times S \mapsto \Delta, & \text{mode} = 0 \\ X \times S \mapsto S_{\text{app}}, & \text{mode} = 1. \end{cases} \tag{12}$$

$\delta_{\text{ext}}(s, x)$ is a piecewise function that either enqueues data messages or updates the queue state and the number of messages that are to be output as per the instructions of the abstract LTE network

$$\delta_{\text{ext}}([\text{idle}, s_q, s_{\text{out}}, s_{\text{app}}], x)$$

$$= \begin{cases} [\text{enqueuing}, \text{runapp}_0(x, s), 0, \text{runapp}_1(x, s)], \\ \qquad x_2 = \text{enqueue} \\ [\text{dequeuing}, x_1, x_2, \text{runapp}_1(x, s)], \quad \text{otherwise}. \end{cases} \tag{13}$$

5) δ_{int}: The internal transition sets s_{ied} to idle, removes the first s_{out} messages (i.e., any messages that may have been output from the queue by λ), sets s_{out} to 0, and leaves s_{app} unchanged

$$\delta_{\text{int}}(s) = [\text{idle}, s_{q-D}, 0, s_{\text{app}}] \tag{14}$$

$D = \{1, 2, ..., s_{\text{out}}\}$ is a set of indices of queue s_q, and s_{q-D} denotes the vector containing the set of elements of s_q (i.e., the set of binary messages of s_q) that do not have the indices present in the index set D.

6) λ: The output function either returns a vector whose elements consist of the node identifier $id \in \Pi$ of the message generator; the queue of the message generator; and a $toLTE$ label, or a vector whose elements consist of a vector of output binary messages and a $toRXers$ label

$$\lambda(s) = \begin{cases} [\text{id}, s_q, toLTE], & s_{\text{ied}} = \text{enqueuing} \\ [s_{qD}, toRXers], & s_{\text{ied}} = \text{dequeuing} \end{cases} \tag{15}$$

$D = \{1, 2, ..., s_{\text{out}}\}$ is a set of indices of the queue s_q, and s_{qD} is the vector containing the elements of s_q that have the indices present in index set D.

7) t_a: The time advance function is defined as follows:

$$t_a(s) = \begin{cases} \infty, & s_{\text{ied}} = \text{idle} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

This way, the message generator waits in an idle state until an input arrives, upon which it will instantaneously call the output function λ.

2) Abstract LTE Network Atomic Model: This atomic model treats the discrete advances in time of the power system simulator as event inputs to synchronize the timing of events in the ICT side of the simulator to the time advancement of the power system dynamics. Moreover, this atomic model dictates how long the queuing delay will be for each message moving from one device to another, and whether or not a message will make it to the receiver, or be lost due to communication failure.

For the LTE network simulator to have access to the messages being generated by the smart grid applications, this atomic model also accepts queue information updates from the message generator atomic models as inputs. The DEVS tuple is characterized as follows.

1) X: The set of all input events is the union of the subframe advance event and the set of possible outputs from the message generator atomic model

$$X = \text{advance\_subframe} \bigcup \Lambda \times toRXers \bigcup \Pi \times \Delta \times toLTE. \tag{17}$$

2) Y: There are N outputs to this atomic model, each one coupling to each of the N message generators implemented in the simulation. As such, Y can be modeled as a set of N-dimensional ordered pairs, where each element is either a two-dimensional vector, or a "null" output. The two-dimensional vector space enumerates all combinations of queue states and the possible numbers of messages that could be forwarded at the time of a subframe advance event. The "null" element is used for particular queues if they are empty, in which case no output is required

$$Y = (\Delta \times \Theta \quad \bigcup \quad \text{null})^N. \tag{18}$$

3) S: The state has four substates $s_t \in S_t$, $s_{iju} \in S_{iju}$, $s_Q \in S_Q$, and $s_{\text{LTE}} \in S_{\text{LTE}}$. s_t is the number of advance_subframe inputs received since the beginning of the simulation, representing the time of simulation (for both the power and communications systems) at the resolution of milliseconds. As such, S_t is the set of natural numbers ranging from 0 to the time duration of the simulation in milliseconds t_{sim} as

$$S_t = \{s_t | s_t \in \mathbb{N}, 0 \leq s_t \leq t_{\text{sim}}\}. \tag{19}$$

S_{iju} consists of "idle", "judgment", and "queue update" states

$$S_{iju} = \{\text{idle}, \text{judgment}, \text{queue\_update}\}. \tag{20}$$

s_Q is a N-tuple containing the states of all message generator atomic model queues in the simulation, which includes both the message queues and the number of messages to be forwarded upon the advancement of a subframe. Such information is needed to determine the simulation trace of the LTE network simulator, which in turn determines when messages can be forwarded to receiving smart grid application nodes. S_Q is defined as follows:

$$S_Q = (\Delta \times \Theta)^N. \tag{21}$$

Finally, S_{LTE} is the set of all combinations of LTE network simulator-specific state variables that characterize its discrete event dynamics. S is defined as the Cartesian product of S_t, S_{iju}, S_Q, and S_{LTE} as

$$S = S_t \times S_{iju} \times S_Q \times S_{\text{LTE}}. \tag{22}$$

4) δ_{ext}: The external transition function increments s_t, changes s_{iju} to judgment, and updates the state of all message generator queues if the input is a subframe advance event. The message generator queues and LTE simulator state variables get updated by the LTE network simulator, which also has access to the current queue states of all the message generators. This functionality is modeled by a generic function $\text{runLTE}_{\text{mode}}()$, which maps the elements of the Cartesian product of X and S to the set of queue state N-tuples if mode $= 0$; if mode $= 1$, $X \times S$ is mapped to S_{LTE} instead

$$\text{runLTE}_{\text{mode}} : \begin{cases} X \times S \mapsto S_Q, & \text{mode} = 0 \\ X \times S \mapsto S_{\text{LTE}}, & \text{mode} = 1. \end{cases} \tag{23}$$

The design of $\text{runLTE}_{\text{mode}}()$ is independent of the cosimulation interface characterized in this section, and is in reality instantiated by the ICT simulation unit chosen for the cosimulator.

In the case that the input is a queue information update from one of the message generators, the external transition function changes s_{iju} to queue_update, and replaces the ith element of s_Q. Thus, the characterization of δ_{ext} is as follows:

$$\delta_{\text{ext}}([s_t, \text{idle}, s_Q, s_{\text{LTE}}], x) =$$

$$\begin{cases} [s_t + 1, \text{judgment}, \text{runLTE}_0(X, S), \text{runLTE}_1(X, S)], \\ \qquad x = \text{advance\_subframe} \\ [s_t, \text{queue\_update}, s_Q^{'}, s_{\text{LTE}}], \quad x_3 = toLTE \end{cases} \tag{24}$$

where

$$s_Q^{'} = (s_{Q1}, s_{Q2}, ..., s_{Qi}, ..., s_{QN}), \quad i = x_1. \tag{25}$$

5) δ_{int}: The internal transition function changes s_{iju} to "idle"

$$\delta_{\text{int}}([s_t, s_{iju}, s_Q, s_{\text{LTE}}]) = [s_t, \text{idle}, s_Q, s_{\text{LTE}}]. \tag{26}$$

6) λ: The output function is defined for when $s_{iju} =$ judgment, and outputs s_Q as

$$\lambda([s_t, \text{judgement}, s_Q, s_{\text{LTE}}]) = s_Q. \tag{27}$$

7) t_a: The time advance function is defined as follows:

$$t_a([s_t, s_{iju}, s_Q, s_{\text{LTE}}]) = \begin{cases} \infty, & s_{iju} = \text{idle} \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

This way, the LTE network increments its state and dictates the advancement, delay, or destruction of messages every subframe at the same instant the power system advances a millisecond.

3) Message Receiver Atomic Model: The message receiver atomic model accepts sets of one or more binary messages as inputs, and generates appropriate numerical signals to their associated generator entity. The DEVS tuple is characterized as follows.

1) X: The set of all possible inputs consists of the set of all possible outputs from a message generator atomic model (although the inputs with $toLTE$ tags are ignored)

$$X = \Lambda \times toRXers \quad \bigcup \quad \Pi \times \Delta \times toLTE. \tag{29}$$

2) Y: The set of possible outputs consists of the Cartesian product of the set of all m-tuples of integers with n or fewer digits, where $m \leq M$ and an enqueue tag, which is exactly $\Gamma \times$ enqueue

$$Y = \Gamma \times \text{enqueue}. \tag{30}$$

3) S: The state has two substates $s_{ia} \in S_{ia}$ and $s_{\text{app}} \in S_{\text{app}}$. S_{ia} consists of idle and active states

$$S_{ia} = \{\text{idle}, \text{active}\}. \tag{31}$$

S_{app} is the collection of all possible combinations of smart grid application-specific state variables that are used to determine the appropriate mapping from input messages to output signals to be sent to the message generation atomic model. Naturally, there is no general way to characterize S_{app}. S is defined as the Cartesian product of S_{ia} and S_{app} as

$$S = S_{ia} \times S_{\text{app}}. \tag{32}$$

4) δ_{ext}: The external transition function is defined when the input substate $s_{ia} = \text{idle}$, and the second element of $x = toRXers$. When the message receiver atomic model receives an input, s_{ia} gets set to active, and the application specific state variables are updated based on the input and current state of the message receiver. Updating the application specific state variables in reality is done by algorithms coded by the smart grid application developer, and the process varies from application to application. However, they can all be modeled in general as some function updateVariables() that maps the elements of X and S_{app} to the elements of S_{app} as

$$\text{updateVariables} : X \times S_{\text{app}} \mapsto S_{\text{app}}. \tag{33}$$

By making use of updateVariables, the external transition function can be defined as follows:

$$\begin{aligned} &\delta_{\text{ext}}([\text{idle}, s_{\text{app}}], x) \\ &\quad = [\text{active}, \text{updateVariables}(s_{\text{app}}, x)], \\ &\quad = x_2 = toRXers. \end{aligned} \tag{34}$$

5) δ_{int}: The internal transition function changes s_{ia} from active to idle

$$\delta_{\text{int}}([\text{active}, s_{\text{app}}]) = [\text{idle}, s_{\text{app}}]. \tag{35}$$

6) λ: The external transition function is up to the smart grid application developer to design, as it specifies the mapping from the state variables of the smart grid application to the set of possible outputs to be sent to the smart grid device's message generator

$$\lambda([\text{active}, s_{\text{app}}]) : S_{\text{app}} \mapsto Y. \tag{36}$$

7) t_a: The time advance function is defined as follows:

$$t_a([s_{ia}, s_{\text{app}}]) = \begin{cases} \infty, & s_{ia} = \text{idle} \\ 0, & \text{otherwise.} \end{cases} \tag{37}$$

This way, the message generator waits in an idle state until an input arrives, upon which it will instantaneously call the output function λ.

4) Remark on Generalization of Cosimulation Interface and Cosimulation Computational Complexity: As mentioned in Section I, future practitioners can refer to the equations in this section as a guide to implementing their own custom cosimulation of a continuous time system coupled with a communication network that uses a time-slot-based MAC scheme. For example, any CT simulation unit that can be wrapped in the augmented DEVS tuple presented in [23, Eq. (3)] can have its inputs and outputs seamlessly connected to the outputs and inputs of the DEVS tuples constructed in this article.

The implementation of the cosimulation interface requires constant time $O(1)$ computational complexity. State information is simply being copied from one simulation environment to the other as needed. The computational complexity of the power system and communication network simulators depend on the simulation unit design, but generally do not exceed $O(N^2)$ complexity. Note that, algorithms used in various levels of the communication system or the power system may contribute to additional complexity. This complexity analysis only refers to the cosimulation interface.

III. Microgrid Simulation: FLISR Application

To test the simulator, an FLISR application is used. As depicted in Fig. 3, we model a 1-km, 5-kV loop distribution feeder with three 5-kW spot loads separated by "smart switches" that relay information to a main feeder CB. Let the smart switch between Loads 1 and 2 be named Switch A, and the one between Loads 2 and 3 be named Switch B. Also, let the smart switch between Load 3 and the backup feeder (supply #2) be named the Tie switch. The breaker can exchange information from these switches and perform FLISR operations, ultimately facilitating remote control over the actuation of such switches to reconfigure the feeder topology in response to faults occurring. For example, Fig. 3 shows that in the event of a fault occurring at Load 2, the FLISR application would reconfigure the circuit topology to disconnect the portion of the power line in which the fault

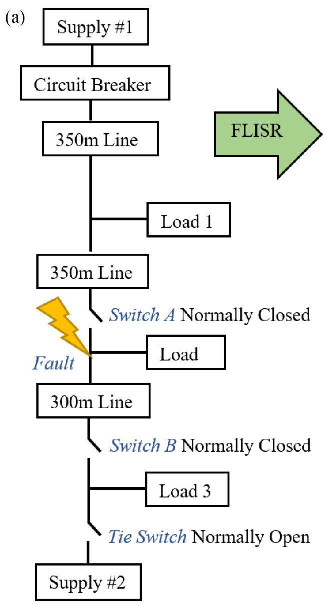


Fig. 3. Depiction of loop feeder topology and its switch configurations in (a) prefault event and (b) postfault event states.

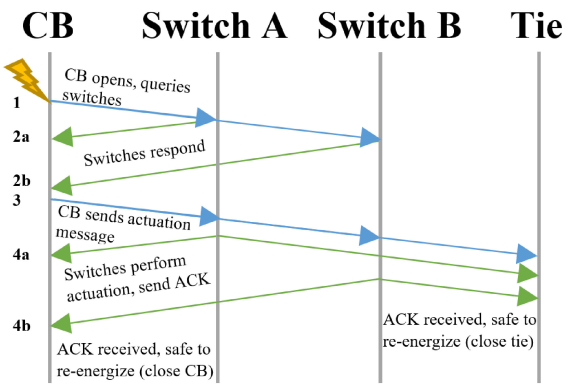

CB
Switch A
Switch B
Tie
1
2a
2b
3
4a
4b
CB opens, queries switches
Switches respond
CB sends actuation message
Switches perform actuation, send ACK
ACK received, safe to re-energize (close CB)
ACK received, safe to re-energize (close tie)

Fig. 4. FLISR application message sequence timing diagram.

is located in a manner that keeps as many loads energized as possible.

Fig. 4 depicts the messages that the various smart grid devices need to send to one another. These messages have been numbered on the left-hand side of the diagram so they can be referenced by the labeling in Fig. 6(d). When the fault is detected by the CB, it opens to de-energize the feeder, and queries Switches A and B to see if they detected the fault. Based on the response of the switches, the CB can then deduce the segment of line in which the fault occurred. Based on where the fault occurred, the CB then sends appropriate actuation commands to Switches A and B in addition to the Tie switch. To assure the line is only re-energized after the fault has been isolated, Switches A and B must execute their actuation commands first, then send an acknowledgement (ACK) message to the CB and Tie switch to notify them that it is safe to carry out their actuation commands.

IV. Results

The FLISR application defined in Section III was put through a series of tests by applying single phase to ground faults at the various loads of the aforementioned power feeder topology, and then, observing the resultant power system waveforms and communication messaging. Further tests were conducted to observe the simulation traces produced by the cosimulator when communication link failures occurred between the various devices of the FLISR application.

In the LTE simulator, a 5-MHz bandwidth scheduling access period was used, within which contains the sidelink control and data resource pools used for D2D communication. The transmission power and antenna gains of each device in the network were set to 20 dBm and 10 dB, respectively. Concerning fundamental large scale fading parameters, the simulator's pathloss model that characterizes the frequency-dependent attenuation of a propagating signal as a function of distance was set to

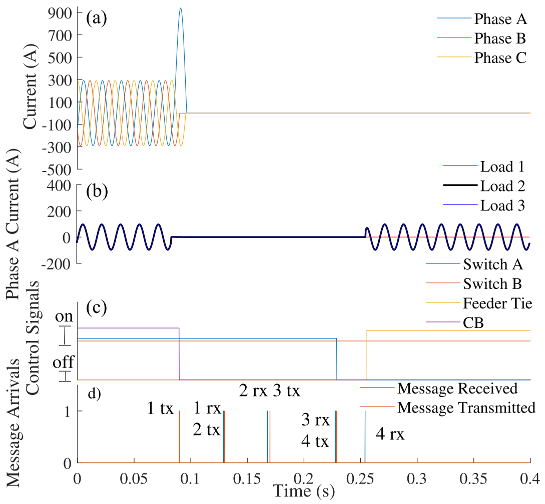


Fig. 5. Cosimulation traces for FLISR application when a fault occurs at Load 1. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

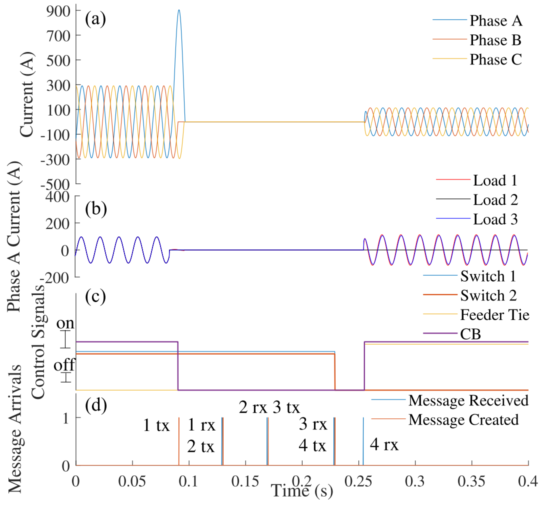


Fig. 6. Cosimulation traces for FLISR application when a fault occurs at Load 2. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

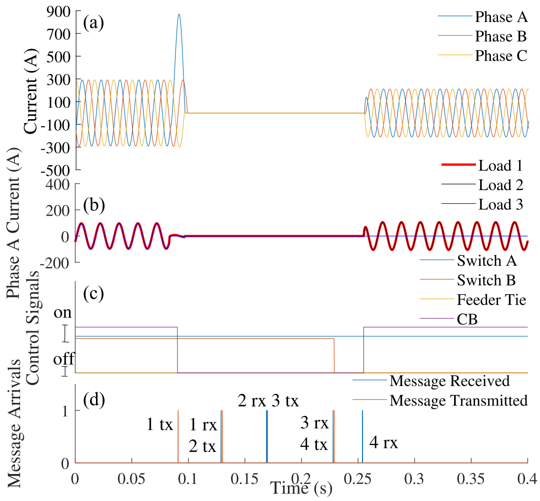


Fig. 7. Cosimulation traces for FLISR application when a fault occurs at Load 3. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

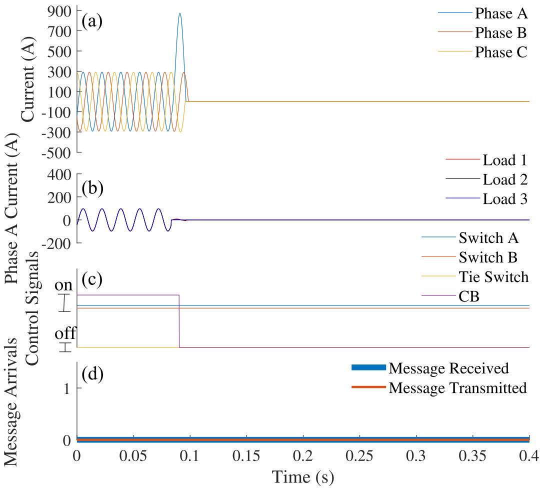


Fig. 8. Cosimulation traces for FLISR application when a fault occurs at Load 2, but a communication failure occurs between the CB and Switches A and B. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

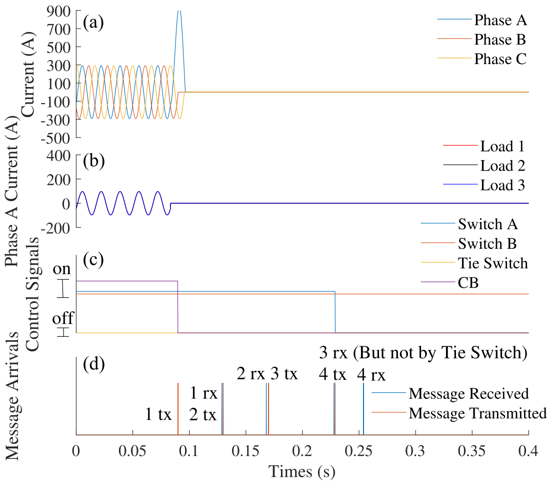


Fig. 9. Cosimulation traces for FLISR application when a fault occurs at Load 1, but a communication failure occurs between the CB and the Tie switch. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

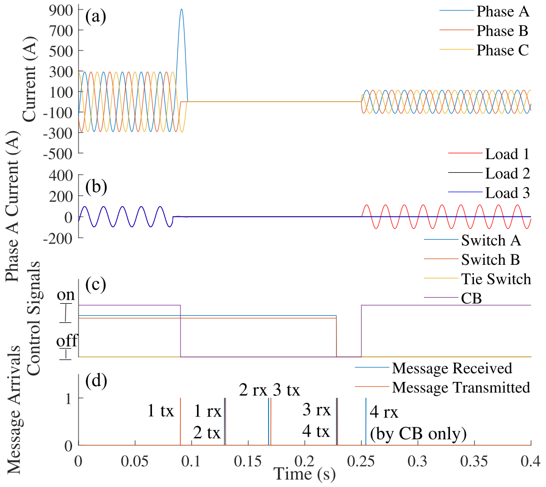


Fig. 10. Cosimulation traces for FLISR application when a fault occurs at Load 2, but a communication failure occurs between the Tie switch and Switches A and B. (a) Three-phase substation egress current. (b) Phase A currents of the three loads on the feeder; Load A is closest to the substation, and Load C is furthest. (c) Control signals of all the switches and main breaker of the feeder. (d) Events where a message has been received by a smart grid device.

a commonly accepted model published in a Third Generation Partnership Project (3GPP) technical report [24]. Moreover, the penetration loss and noise (present in the communication devices' electronics) was set to 5 dB. Finally, shadowing was modeled as a zero mean Gaussian random variable with a 2-dB variance [25].

Figs. 5–7 visualize the power and ICT system dynamics of the FLISR application handling faults occurring at Loads 1–3 on the distribution feeder, respectively. Figs. 8–10 observe the impact of the joint power system ICT dynamics when communication failures occur between the CB and Switches A and B; the CB and the Tie switch; and the Switches A and B and the Tie switch, respectively. It should be noted that a fault is being simulated at Load 2 for Figs. 8 and 10, meaning that their waveforms would look exactly like those in Fig. 6 if no communication failures were to have been introduced. However, for Fig. 9, the fault is being simulated at Load 1.

In Fig. 5(a), one can see how a fault occurring at Load 1 results in the substation egress feeder shutting OFF its current supply indefinitely. This is to be expected, as the feeder breaker and Switch A must open to isolate a fault at Load 1. Moreover, in Fig. 5(b), Loads 2 and 3 get restored by the backup feeder, while Load 1 remains isolated from the power system. This is also expected, as the FLISR application was designed to close the tie switch, reclose Switch 2, and open the CB and Switch 1 in this particular situation, which is observed to be the case in Fig. 5(c).

As can be seen in Fig. 5(d), as well as Figs. 6(d), and 7(d), the message sequencing is consistent with the FLISR application model shown in Fig. 4; the changes in the relay signals occur immediately after actuation commands have been received by Switches A and B, as well as after the ACK messages have been received by the CB and Tie switch.

In Fig. 6(a), one can see how a fault occurring at Load 2 results in the substation egress feeder shutting OFF its current supply while the FLISR application operates. The peak egress current is a third of its original value upon restoration. Such a result is as a consequence of Load 2 being isolated from the system, and Load 3 being restored by the backup feeder in lieu of its original feeder, making Load 1 the only remaining demand left on the original feeder. Fig. 6(b) is consistent with the explanation of Fig. 6(a), as Load 2's current output remains zero after the FLISR event, while Loads 1 and 3 get restored. The control signalling in Fig. 6(c) is also consistent with the other subplots, as it shows Switches A and B opening (i.e., their control signals lowering), the Tie switch closing, and the CB reclosing.

In Fig. 7(a), one can see how a fault occurring at Load 3 results in the substation egress feeder shutting OFF its current supply while the FLISR application operates. The peak egress current is two thirds of its original value upon restoration. Such a result is a consequence of Load 3 being isolated from the system, while Loads 1 and 2 are able to be restored by the original feeder since the fault is downstream of these loads. Fig. 7(b) is consistent with the explanation of Fig. 7(a), as Load 3's current output remains zero after the FLISR event, while Loads 1 and 2 get restored. The control signalling in Fig. 7(c) is also consistent with the other subplots, as it shows Switch B opening, the Tie switch remaining open, and Switch A as well as the CB reclosing.

In Fig. 8, one can see that the communication link failure between the CB and Switches A and B results in a complete failure of the FLISR application as expected. Aside from the main feeder breaker shutting OFF due to the local detection of the high current (which does not require any communication messaging), no other control signals change in Fig. 8(c), which is consistent with the absence of the communication network activity observed in Fig. 8(d). Because the CB cannot initiate the FLISR application, clearly none of the loads are restored in 8(b).

In Fig. 9, one can see that the communication link failure between the CB and the Tie switch results in a partial failure of the FLISR application as expected. Despite the control signals of Switches 1 and 2 working properly in Fig. 9(c), the Tie switch fails to close due to communication failure, which is consistent with the communication network activity observed in Fig. 9(d). Because the Tie switch never hears the actuation command sent by the CB, the Tie switch never closes to connect Loads 2 and 3 to the backup feeder, and thus, none of the loads are restored in Fig. 9(b).

In Fig. 10, one can see that the communication link failure between Switches A and B and the Tie switch results in a partial failure of the FLISR application as expected. Despite the control signals of Switches 1 and 2 working properly in Fig. 10(c), the Tie switch fails to close due to communication failure, which is consistent with the communication network activity observed in Fig. 10(d). Because the Tie switch never hears the ACK messages sent by Switches A and B, the Tie switch never closes to connect Load 3 to the backup feeder, and thus, only Load 1 is restored in Fig. 10(b).

V. Conclusion

This article introduced a formal method to combine the existing simulation software packages available in the MATLAB/Simulink environment to create a self-contained LTE D2D-enabled smart grid simulator with detailed ICT and power system dynamics alike. A discrete-event simulation-based interface was mathematically characterized and implemented to facilitate our simulation. The cosimulator code is made available to the research community. The simulator was also tested by using it to simulate an FLISR application on a simple loop feeder for various fault locations and communication failure scenarios.

Acknowledgment

The authors would like to thank P. Tayebi of the SUNLAB research group at the University of Ottawa, Ottawa, ON, Canada for sharing her knowledge of Simscape Power Systems.

References

[1] M. J. B. Reddy, D. V. Rajesh, P. Gopakumar, and D. K. Mohanta, "Smart fault location for smart grid operation using RTUs and computational intelligence techniques," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1260–1271, Dec. 2014.

[2] U. Kumar Debnath, I. Ahmad, D. Habibi, and A. Y. Saber, "Improving battery lifetime of gridable vehicles and system reliability in the smart grid," *IEEE Syst. J.*, vol. 9, no. 3, pp. 989–999, Sep. 2015.

[3] M. Elsayed, M. Erol Kantarci, B. Kantarci, L. Wu, and J. Li, "Low-latency communications for community resilience microgrids: A reinforcement learning approach," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1091–1099, Mar. 2020.

[4] R. Hidalgo-León, C. Sanchez-Zurita, P. Jácome-Ruiz, J. Wu, and Y. Muñoz-Jadan, "Roles, challenges, and approaches of droop control methods for microgrids," in *Proc. IEEE PES Innovative Smart Grid Technol. Conf.-Latin Amer.*, Sep. 2017, pp. 1–6.

[5] F. Lemercier, G. Habault, G. Z. Papadopoulos, P. Maillé, N. Montavont, and P. Chatzimisios, "Communication architectures and technologies for advanced smart grid services," in *Transportation and Power Grid in Smart Cities: Communication Networks and Services* pp. 217–245, eds. H. T., M. Erol-Kantarci, and M. H. Rehmani, Eds. New York, NY, USA: Wiley, 2019, pp. 217–245. [Online]. Available: https://ieeexplore.ieee.org/document/8653942

[6] D. Verma, S. Nema, and R. K. Nema, "Implementation of perturb and observe method of maximum power point tracking in SIMSCAPE/MATLAB," in *Proc. Int. Conf. Intell. Sustain. Syst.*, 2017, pp. 148–152.

[7] M. Fahmida, M. S. Alam, and M. A. Hoque, "MATLAB SIMSCAPE simulation of solar photovoltaic array fed BLDC motor using maximum power point tracker," in *Proc. Int. Conf. Robot., Elect. Signal Process. Techn.*, Jan. 2019, pp. 662–667.

[8] C. S. Salvador and A. A. L. Manalo, "Planned renewable energy usage during power outage," in *Proc. 25th Int. Conf. Syst. Eng.*, Aug. 2017, pp. 377–385.

[9] O. Adeyemo, P. Idowu, A. Asrari, and J. Khazaei, "Reactive power control for multiple batteries connected in parallel using modified power factor method," in *Proc. North Amer. Power Symp.*, Sep. 2018, pp. 1–6.

[10] A. Delavari, I. Kamwa, and P. Brunelle, "Simscape power systems benchmarks for education and research in power grid dynamics and control," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, May 2018, pp. 1–5.

[11] T. D. Castillo and J. R. Castro, "Renewable energy source P-V connected to the grid through shunt active power filter based in p-q theory," in *Proc. IEEE 3rd Ecuador Tech. Chapters Meeting*, Oct. 2018, pp. 1–5.

[12] J. Khazaei, M. Beza, and M. Bongiorno, "Impedance analysis of modular multi-level converters connected to weak ac grids," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 4015–4025, Jul. 2018.

[13] G. Schweiger *et al.*, "An empirical survey on co-simulation: Promising standards, challenges and research needs," *Simul. Model. Pract. Theory*, vol. 95, pp. 148–163, May 2019.

[14] H. Elkhorchani and K. Grayaa, "Smart micro grid power with wireless communication architecture," in *Proc. Int. Conf. Elect. Sci. Technol. Maghreb*, Nov. 2014, pp. 1–10.

[15] T. Shlebik, A. Fadel, M. Mhereeg, and M. Shlebik, "The development of a simulation-based smart grid communication management system using MATLAB," in *Proc. Int. Conf. Green Energy Convers. Syst.*, 2017, pp. 1–7.

[16] IEEE Task Force on Interfacing Techniques for Simulation Tools and S. C. Müller *et al.*, "Interfacing power system and ICT simulators: Challenges, state-of-the-art, and case studies," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 14–24, Jan. 2018.

[17] K. Shimotakahara, M. Elsayed, K. Hinzer, and M. Erol Kantarci, "High-reliability multi-agent q-learning-based scheduling for D2D microgrid communications," *IEEE Access*, vol. 7, pp. 74 412–74 421, 2019.

[18] K. Shimotakahara, M. Elsayed, K. Hinzer, and M. Erol Kantarci, "Integrated power and device-to-device (D2D) communications simulator for future power systems," in *Proc. IEEE Global Conf. Signal Inform. Process.*, 2019, pp. 1–5.

[19] A. Falcone and A. Garro, "Distributed co-simulation of complex engineered systems by combining the high level architecture and functional mock-up interface," *Simul. Model. Pract. Theory*, vol. 97, Aug. 2019, Art. no. 101967.

[20] C. Thule, K. Lausdahl, C. Gomes, G. Meisl, and P. Larsen, "Maestro: The INTO-CPS co-simulation framework," *Simul. Model. Pract. Theory*, vol. 92, pp. 45–61, Apr. 2019.

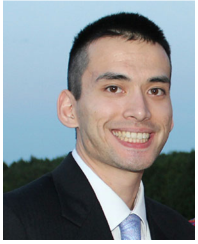
[21] NETCORE, "Netcore projects," 2020. [Online]. Available: https://github.com/KevinShimotakahara/NETCORE-Projects

[22] G. A. Wainer, *Discrete-Event Modeling and Simulation: A Practitioner's Approach*, 1st ed. Boca Raton, FL, USA: CRC Press, 2009.

[23] J. Nutaro, P. T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, "Integrated hybrid-simulation of electric power and communications systems," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jun. 2007, pp. 1–8.

[24] ETSI Report on LTE R9, "Evolved universal terrestrial radio access (E-UTRA); Radio frequency (RF) requirements for LTE Pico Node B (3GPP TR 36.931 version 9.0. 0 Release 9 Section 5.3.2.2.2)," ETSI, Sophia Antipolis, France, Tech. Rep. 136 931, 2011.

[25] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.



Kevin Shimotakahara (Student Member, IEEE) received the B. Eng. degree from Carleton University, Ottawa, ON, Canada, in 2018. He is currently working toward the M.A.Sc. degree with the University of Ottawa, Ottawa, Canada.

He has participated in research activities with energy-related institutions, including the Fraunhofer Institution for Solar Energy Systems, where he developed artificial intelligence (AI)-based electric vehicle charging algorithms. His research interests include smart grids, renewable energy, and AI.

Mr. Shimotakahara was the recipient of the Natural Sciences and Engineering Research Council of Canada (NSERC) CGS-M scholarship in 2019 and also received funding from the NSERC CREATE TOP-SET training program throughout his graduate studies.

Medhat Elsayed (Student Member, IEEE) received the B.Sc. and M.Sc. degrees from Cairo University, Giza, Egypt, in 2009 and 2013, respectively. He is currently working toward the Ph.D. degree with the University of Ottawa, Ottawa, ON, Canada.

He is the author of several publications optimizing wireless networks via machine learning techniques. In addition, he is a coinventor of a provisional patent application. His current research interests include developing innovative machine learning algorithms for 5G wireless networks and beyond.

Mr. Elsayed was the recipient of the NSERC CREATE TOPSET graduate research training scholarship from the University of Ottawa in 2018. He delivered a tutorial at the Second Annual Workshop of the Ottawa AI Alliance held in National Research Council Canada, Ottawa, in 2019. The tutorial covered background on deep learning and reinforcement learning, a survey on the state-of-art work, and his recent work in the field.

Melike Erol-Kantarci (Senior Member, IEEE) received the B.Sc. degree from Istanbul Technical University, Istanbul, Turkey, and the M.Sc. and Ph.D. degrees in computer engineering from Istanbul Technical University, in 2009 and 2004, respectively.

She was a Fulbright Visiting Researcher with the Computer Science Department, University of California Los Angeles (UCLA), Los Angeles, CA, USA, during her Ph. D. studies. She is an Associate Professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada. She is the Founding Director of the Networked Systems and Communications Research (NETCORE) laboratory. Her main research interests include artificial-intelligence (AI)-enabled wireless networks, 5G and 6G wireless communications, smart grid, electric vehicles, Internet of Things, and wireless sensor networks.

Dr. Erol-Kantarci is Tier 2 Canada Research Chair in AI-enabled Next-Generation Wireless Networks. She was selected for the 2019 list of N2Women: Stars in Computer Networking and Communications. She was the recipient of several best paper awards including the IEEE Communication Society Best Tutorial Paper Award in 2017. She is the co-editor of three books on smart grids, smart cities, and intelligent transportation. She has delivered numerous plenary talks, tutorials, and seminars around the globe. She is on the editorial board for the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE INTERNET OF THINGS JOURNAL, IEEE COMMUNICATIONS LETTERS, *IEEE Vehicular Technology Magazine*, and IEEE ACCESS. She has acted as the General Chair and Technical Program Chair for many international conferences and workshops.

Karin Hinzer (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in physics from the University of Ottawa, Ottawa, ON, Canada, in 1996, 1998, and 2002, respectively.

She is Vice-Dean, Research with the Faculty of Engineering and a Professor with the School of Electrical Engineering and Computer Science with a cross-appointment with the Department of Physics, University of Ottawa. She gained extensive experience in the design and fabrication of group III–V semiconductor devices while at the National Research Council Canada, Nortel Networks, and then, Bookham (now Oclaro). Cost reduction strategies and liaison with remote fabrication facilities strongly feature in her industry experience. She joined the University of Ottawa in 2007, where she founded the SUNLAB, the premier Canadian modeling and characterization laboratory for next generation multijunction solar devices and concentrator systems. Her research involves developing new ways to harness the Sun's energy. She has authored and co-authored more than 170 refereed papers, trained over 150 highly qualified personnel and her laboratory has spun-off three Canadian companies in the energy sector. Her research interests include new materials, high-efficiency light sources and light detectors, solar cells, solar modules, new electrical grid architectures/controls, and voltage converters.

Prof. Hinzer was the Tier II Canada Research Chair in Photonic Nanostructures and Integrated Devices, from 2007 to 2017. She was the recipient of the Inaugural Canadian Energy Award with industry partner Morgan Solar for the development of more efficient solar panels in 2010, the Ontario Ministry of Research and Innovation Early Researcher Award for her contributions to the fields of photonic devices and photovoltaic systems in 2015, and the University of Ottawa Young Researcher Award in 2016. She is a Member of the College of New Scholars, Artists, and Scientists of the Royal Society of Canada. She is the Principal Investigator of the Natural Sciences and Engineering Research Council of Canada Collaborative Research and Training Experience Program titled "Training in Optoelectronics for Power: From Science and Engineering to Technology" (NSERC CREATE TOP-SET), a multidisciplinary training program involving three universities and training over 100 students in seven years. She is an Editor for the IEEE JOURNAL OF PHOTOVOLTAICS.