This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

Detection of False-data Injection Attacks in Cyber-Physical DC Microgrids

Omar Ali Beg, *Student Member, IEEE,* Taylor T. Johnson, *Member, IEEE,* and Ali Davoudi, *Senior Member, IEEE*

***Abstract*—Power electronics-intensive DC microgrids use increasingly complex software-based controllers and communication networks. They are evolving into cyber-physical systems (CPS) with sophisticated interactions between physical and computational processes, making them vulnerable to cyber attacks. This work presents a framework to detect possible false-data injection attacks (FDIA) in cyber-physical DC microgrids. The detection problem is formalized as identifying a change in sets of inferred candidate invariants. Invariants are microgrids properties that do not change over time. Both the physical plant and the software controller of CPS can be described as Simulink/Stateflow (SLSF) diagrams. The dynamic analysis infers the candidate invariants over the input/output variables of SLSF components. The reachability analysis generates the sets of reachable states (reach sets) for the CPS modeled as hybrid automata. The candidate invariants that contain the reach sets are called the actual invariants. The candidate invariants are then compared with the actual invariants, and any mismatch indicates the presence of FDIA. To evaluate the proposed methodology, the hybrid automaton of a DC microgrid, with a distributed cooperative control scheme, is presented. The reachability analysis is performed to obtain the reach sets and, hence, the actual invariants. Moreover, a prototype tool, HYbrid iNvariant GEneratoR (Hynger), is extended to instrument SLSF models, obtain candidate invariants, and identify FDIA.**

***Index Terms*—Cyber-physical systems, dc microgrid, distributed control, false-data injection attack, hybrid automaton.**

I. Introduction

ISLANDED multi-converter DC microgrids have advantages over their AC counterparts, including higher reliability, simpler control, and more efficient interfacing with naturally-DC renewable energy sources, electronics loads, and energy storage units [1], [2]. Therefore, DC microgrids have emerged as a key technology for the future, and their related control methodologies are also evolving. Given the well-established advantages of distributed control schemes over centralized control methodologies, the migration from current central controllers to future distributed schemes is inevitable [3]–[8]. The centralized control systems require two-way, high bandwidth communication links between the central controller and every other agent, and expose a single point-of-failure. Moreover, sparsity of communication networks utilized in distributed control schemes reduces the infrastructure cost, and improves solution scalability compared to a fully-connected communication network.

These DC microgrids are evolving into cyber-physical systems (CPS) with sophisticated software-based control and communication networks. Such CPS are, however, vulnerable to cyber attacks, as there is no central entity to monitor activities of all DC-DC converters leading to a limited global situational awareness. This vulnerability is analogous to the situation in cyber-physical power systems that have faced various types of cyber attacks, e.g., false-data injection attack (FDIA) [9], denial of service [10], [11], jamming [12], and random attacks [13]. Some prevention strategies for jamming include frequency hopping, direct-sequence spread spectrum technique, channel surfing, and protocol hopping [14]. In this work, detection of FDIA in power electroncis-intensive DC microgrids is considered that involves spoofing a signal, either in sensors or the communication network, through an attack vector that aims to disrupt the steady-state operation [9]. The attack vector formulation is a sophisticated process, and requires expert knowledge of the entire system. The intruder should have either physical access to a specified number of meters, or a complete knowledge of the infrastructure and the communication network [9].

The preventive measures against FDIA include physical security, information security, and communication security. With regards to the physical security, a minimum number of strategically selected set of sensor measurements (called as basic measurements) that need to be protected to thwart FDIA has been proposed [15]. Moreover, phasor measurement units (PMUs) can be strategically placed to protect power grids against such attacks [16]. However, PMUs are also vulnerable due to their use of global positioning systems [17]. With regards to information security, a prevention strategy against FDIA involves dynamically changing the information structure of microgrids [18]. In general, the communication security can be improved using stringent cryptographic techniques, i.e., encryption, authentication, and key management for power systems [19]. For example, a communication security architecture for distributed microgrid control [20] exchanges encrypted information. A trusted sensing base is proposed in the form of a current transformer that encrypts the AC power signal before sending it to PMUs [21].

Recent work on FDIA detection, albeit in power systems [11], [13], [22]–[29], broadly employs state estimation processes, e.g., using Kalman filters [13], sparse op-

The material presented in this paper is based upon work supported by the National Science Foundation (NSF) under grant numbers ECCS-1405173, CNS 1464311, and SHF 1527398, the Air Force Research Laboratory through contract number FA8750-15-1-0105, and the Air Force Office of Scientific Research under contract numbers FA9550-15-1-0258 and FA9550-16-1-0246. Omar A. Beg, and A. Davoudi are with the University of Texas, Arlington, TX 76019, USA. T. Johnson is with Vanderbilt University, Nashville, TN 37240, USA. (e-mail: omar.beg@mavs.uta.edu; taylor.johnson@vanderbilt.edu; davoudi@uta.edu).

Manuscript received August 29, 2016; revised December 22, 2016; accepted January 12, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

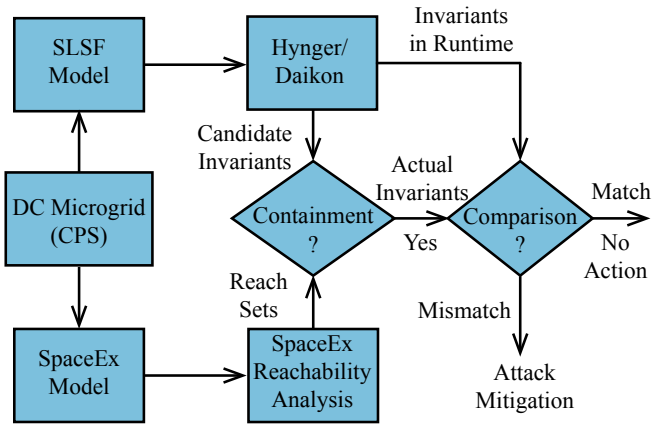

Fig. 1. The proposed FDIA detection framework bridges the gap between the software-based anomaly detection techniques and power electronics-intensive DC microgrids modeled as hybrid automata.

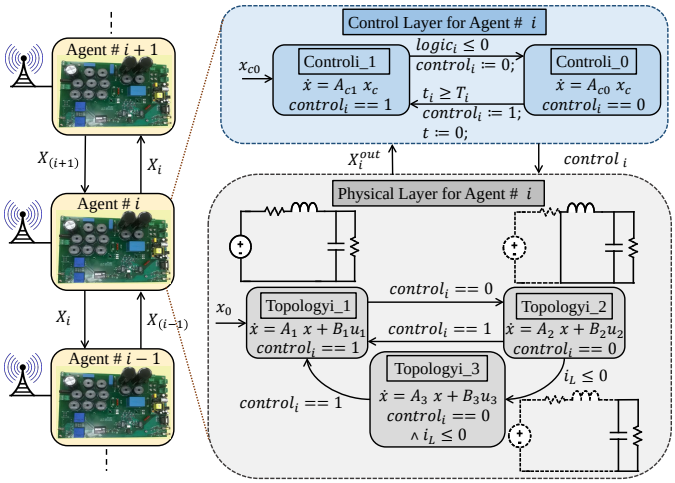

Fig. 2. Hybrid automaton of a cyber-physical DC microgrid showing converter and controller interactions. Each converter, its corresponding controller, and its communication links are, altogether, considered as an agent. Each agent shares its information with the neighboring agents on the communication graph.

timization [22], generalized likelihood ratio [23], Kullback-Leibler distance [24], Chi-square detector and similarity matching [25], state forecasting [26], and machine-learning techniques [27]. However, to the best of authors' knowledge, FDIA detection in software-intensive DC microgrids is not systematically studied yet. This work aims to formalize the FDIA detection problem as a change in sets of inferred *invariants*; system properties that do not change over time. Here, invariants are defined in terms of bounds over the output voltage and current of individual converters.

The overall block diagram of the proposed FDIA detection framework is shown in Fig. 1. The candidate invariants are inferred from the Simulink/Stateflow (SLSF) model of the DC microgrid. Hynger (HYbrid iNvariant GEneratoR) [30] tool is used to provide an interface between the SLSF model and the Daikon tool [31], [32]. Daikon is a software-based invariant inference tool. Hynger takes the SLSF model as an input, executes it to generate time traces, and transforms them into a format compatible with Daikon to generate candidate invariants. Moreover, the cyber-physical DC microgrid is formally modeled as multi-agent hybrid automata, and the reachability analysis is performed using SpaceEx [33] to obtain the reachable set of states (called the reach sets). The Hynger/Daikon combination provides only the candidate invariants. The SpaceEx tool is used concurrently in the proposed framework to obtain the actual invariants. The candidate invariants that contain the reach sets are called the actual invariants. The candidate invariants are then compared with the actual invariants, and any mismatch indicates the presence of FDIA. A mitigation strategy can then disconnect the affected converter and prevent the microgrid's instability.

The remainder of this paper is organized as follows: The hybrid automaton modeling of DC microgrids that includes both physical and cyber layers is discussed in Section II. The FDIA detection framework for DC microgrids is discussed in Section III. Section IV studies a DC microgrid prototype, with an analysis of FDIA effects, detection using the proposed framework, and mitigation. Section V concludes the paper.

II. Cyber-physical DC Microgrid as Multi-agent Hybrid Automata

The proposed FDIA detection framework requires the CPS modeled as SLSF diagrams and as hybrid automata to obtain the candidate invariants and the reach sets, respectively. A hybrid automaton [34] is a formal model, essentially a finite-state machine with additional continuous dynamic variables. Cyber-physical DC microgrids can be modeled as multi-agent hybrid automata, where power electronics DC-DC converters (referred to as converters) form the physical layer, and the software-based controller with the communication network among converters, altogether, form the cyber layer. Each converter, with its corresponding controller and communication links, is considered an agent, and its hybrid automaton is shown in Fig. 2. This hybrid automaton exchanges information with its two immediate neighbors, e.g., $(i+1)$th and $(i-1)$th agents in Fig. 2, through *global variables* to implement a cooperative control protocol.

A. Modeling the Physical Layer

The output voltage v_i^{out} and output current i_i^{out} of the ith converter are regulated by controlling the MOSFET switch through the corresponding control layer. The switching state of the MOSFET switch leads three different topologies (switching sub-interval) as shown in Fig. 2. The state of a hybrid automaton may change either through a continuous flow trajectory within a given topology, or through a discrete transition between two given topologies.

1) Formal Hybrid Automaton: Let $\mathbb{R}^n$ be the set of n-dimensional reals, and 2^X be the power set of a given set X, i.e., the set of all the subsets of X.

Definition 2.1: A *hybrid automaton* is defined by a tuple $\mathcal{H} = \langle Q, X, \Theta, U, F, \mathcal{T}, E, G, inv \rangle$:

- $Q = \{q_1, q_2, ..., q_N\}$ is a finite set of topologies.
- X is a finite set of continuous variables, with $\forall\ x \in X\ \exists\ val(x) \in \mathbb{R}$, where $val(x)$ is a valuation of x as a result of a function mapping. $X = X_g \cup X_l$, such that X_g is the set of global variables and X_l is the set of local

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

variables. Further, $X_g = In \cup O$, where In is the set of global input variables and O is the set of global output variables. A *state* is defined by $s = (q, val(x)) \in Q \times \mathbb{R}$.

- $\Theta \subseteq Q \times \mathbb{R}^n$ is a set of initial conditions.
- $U = \{u_1, u_2, ...u_N\}$ is the set of inputs for each topology.
- F is a finite set of ODEs defined for each $q \in Q$ over the continuous variables $x \in X$. $F(q, x)$ defines the continuous dynamics for each $q \in Q$ over a time period T, and assigns a Lipschitz continuous vector space in $\mathbb{R}^n$.
- $\mathcal{T}$ is a finite set of continuous flow trajectories that define $val(x)$ over $[0\ T]$ from given initial conditions $(q, x_0) \in \Theta$, such that $\forall\ \tau(q,x) \in \mathcal{T}$, $\exists\ s \in \tau(q,x)$ that satisfies $inv(q)$ (i.e., $s \in \tau(q,x) \vDash inv(q)$). A continuous flow trajectory is given by

$$\tau(q, x) = x_0 + \int_0^T F(q, x)dt. \tag{1}$$

- E is a finite set of feasible discrete transitions allowed among the topologies. It is defined by a tuple $e = \langle q, q', g, x' \rangle$, such that a discrete transition is allowed from source topology q to the destination topology q' only when the associated guard condition g is satisfied, and the continuous state is updated to x' after the transition. It might not be possible to visit the entire set of topologies from one particular topology.
- $G \subseteq 2^X$ is the guard set such that $\forall\ e\ \exists\ g \in G$. A *guard* must be satisfied by a state to take a discrete transition from a given topology to another. A state $s = (q_k, val(x))$ *satisfies* g (i.e, $s \vDash g$) iff $q_k = q_l \in e = (q_l, q_l', g, x')$ and $val(x) \in g$.
- inv is a finite set of invariants, where an invariant is associated to each given topology, i.e., $\forall\ q \in Q\ \exists\ inv(q) \subseteq \mathbb{R}^n$. An *invariant* is a property of the hybrid automaton that must be satisfied by all the states for a given topology. A state $s \vDash inv(q)$ iff $val(x) \in inv(q)$.

If a state $(q, val(x))$ does not satisfy an invariant $inv(q)$, the continuous state x stops evolving within a topology. The guard function ensures a discrete transition to an appropriate topology once the corresponding guard is satisfied. Here, invariants and guards are defined in the form of bounds over continuous state variables. The semantics of the hybrid automaton $\mathcal{H}$ is defined by its execution, ϵ. An *execution* is defined as a sequence of states, $\epsilon = s_0, s_1, s_2, \ldots$, obtained as a result of continuous flow trajectories and discrete transitions.

2) Instantiation of the Physical Layer: The hybrid automaton of the ith buck converter is considered for instantiation, where v_i^{in} is the DC input voltage. The continuous dynamics, for a given topology, is given by a set of state-space equations

$$\frac{dx}{dt} = A_q x + B_q u. \tag{2}$$

$A_q \in \mathbb{R}^{n \times n}$ and $B_q \in \mathbb{R}^{n \times m}$ are system matrices. Subscript q denotes the appropriate topology. The instantiation of the hybrid automaton for the ith agent, as per Definition 2.1, is

- Three topologies, shown in Fig. 2, are denoted by $Q = \{q_1, q_2, q_3\}$.
- $X = \{i_i^L, v_i^C, i_i^{out}, v_i^{out}, control_i\}$, where $X_l = \{i_i^L, v_i^C\}$ and $X_g = \{i_i^{out}, v_i^{out}, control_i\}$.
- $U = \{[v_i^{in}, 0, 0, 0]', [0, 0, 0, 0]', [0, 0, 0, 0]'\}$ forms the input vector set.
- $E = \{(q_1, q_2, g_{12}, x'), (q_2, q_1, g_{21}, x'), (q_2, q_3, g_{23}, x'), (q_3, q_1, g_{31}, x')\}$ defines the feasible discrete transitions, e.g., (q_2, q_3, g_{23}, x') means that a discrete transition from topology q_2 to q_3 is allowed, if the guard $g_{23} = \{(i_i^L \leq 0)\}$ is satisfied and the continuous state is reset to x'.
- Guard set, for the corresponding elements of E, is defined by $G = \{(control_i == 0), (control_i == 1), \left(i_i^L \leq 0\right), (control_i == 1)\}$. Signals received from the control layer are $control_i == 1$ and $control_i == 0$ to set the MOSFET ON and OFF, respectively.
- The continuous flow trajectory is defined by (2), with the corresponding state matrices for each topology.

The evolution of the hybrid automaton model starts with initial conditions from the set $init$, e.g., $(q_1, x_0) \in init$ for a given input $u_1 = [v_i^{in}, 0, 0, 0]'$ and, subsequently, the continuous state evolves according to the flow function. The topology remains the same, i.e., $q(t) = q_1$, as x_0 evolves inside the invariant $inv(q_1)$ and attains a final value $x' \in inv(q_1)$. Once the continuous state x' satisfies the corresponding guard, $g_{12} = \{(control_i == 0)\}$ corresponding to the topology q_1, the topology may transition from q_1 to q_2, and the continuous state is reset with a new value x'' in the new invariant set $inv(q_2)$ with a new input $u_2 = [0, 0, 0, 0]'$.

B. Modeling the Cyber Layer

Microgrid control hierarchy is divided into three levels, i.e., primary, secondary, and tertiary [35]. Primary control features the fastest response, and is based entirely on local measurements with no communication. Secondary control operates on a slower time scale, often with reduced communication bandwidth by using sampled measurements. In this work, we consdier two control objectives: proportional load sharing among converters, according to their power ratings, and global voltage regulation of the distribution bus. These objectives are implemented in the secondary control layer through proportional load sharing sub-layer and global voltage regulation sub-layer (which includes a voltage observer and a noise cancellation module), as shown in Fig. 3. We use a distributed cooperative control scheme, i.e., the output of a particular agent depends only on its information and its N_i neighbors on the communication graph [3]. A *graph* $\mathbb{G}$ is defined as a pair (tuple) of a set of vertices and edges, i.e., $\mathbb{G} = (\Lambda, \varepsilon)$. Let $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_N\}$ define a set of N vertices (nodes), and $\varepsilon \subseteq \Lambda \times \Lambda$ a set of edges. An edge from node λ_i to λ_j is a pair $(\lambda_i, \lambda_j) \in \varepsilon$. The graph is said to be *bi-directional* if $(\lambda_i, \lambda_j) \in \varepsilon \implies (\lambda_j, \lambda_i) \in \varepsilon,\ \forall\ i, j \in \Lambda$.

A graph may be represented by an *adjacency matrix* $\mathbb{A} = [a_{ij}]$ with weights $a_{ij} > 0$ if $(\lambda_j, \lambda_i) \in \varepsilon$, and $a_{ij} = 0$ otherwise. The local control protocol, u_i for each agent i is

$$u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i), \tag{3}$$

such that the control of each agent depends only on the difference between its state and those of its neighbors. This protocol ensures that all agents reach a consensus.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

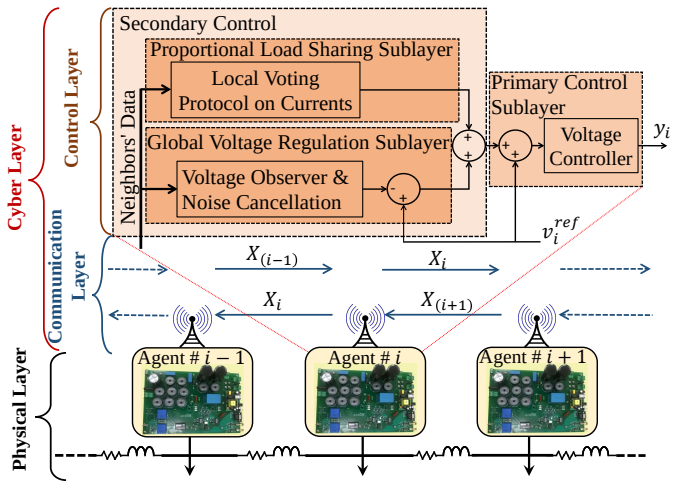

Fig. 3. Structure of the cyber-physical DC microgrid showing the cyber and physical layers and the control sub-layers.

The global voltage reference for N agents is defined as $V_{ref} = [v_1^{ref}, v_2^{ref}, ..., v_N^{ref}]^T$, input DC voltage as $V_{in} = [v_1^{in}, v_2^{in}, ..., v_N^{in}]^T$, output DC voltage vector as $V_{out} = [v_1^{out}, v_2^{out}, ..., v_N^{out}]^T$, output current vector as $I_{out} = [i_1^{out}, i_2^{out}, ..., i_N^{out}]^T$, the voltage estimation vector for the voltage observer module as $V_{est} = [v_1^{est}, v_2^{est}, ..., v_N^{est}]^T$, the per-unit current vector as $X_{pu} = [x_1^{pu}, x_2^{pu}, ..., x_N^{pu}]^T$, and the estimate of the voltage deviation vector for the noise cancellation module as $W_{est} = [w_1^{est}, w_2^{est}, ..., w_N^{est}]^T$. Here, x_i^{pu} refers to the loading percentage of the ith agent. As shown in Fig. 3, X_i depicts the information vector communicated from the ith agent to the $(i-1)$th and $(i+1)$th agents, such that $X_i = [x_i^{pu}, v_i^{est}, w_i^{est}]^T$. Moreover, X_{i-1}, and X_{i+1} of Fig. 3 are defined similarly. Communication links are modeled as low-pass filters ($\mathbb{T}_1$ and $\mathbb{T}_2$ in Appendix) to emulate delays inherent in the data exchange process, as in [3], [36], [37]. Here, i_i^{out} and v_i^{out} are passed through $\mathbb{T}_1$ to get the per-unit current x_i^{pu} and the voltage y_i^{vo}, respectively.

The control sub-layers are discussed next.

1) Proportional load sharing sub-layer: The ith agent shares per-unit current information with its immediate neighbors, i.e., $(i-1)$th and $(i+1)$th agents. This sub-layer has a PI controller with parameters $P(i,i,)$ and $I(i,i)$, where P and I are $N \times N$ matrices that contain the proportional and integral terms, respectively. If $\mathbb{C}$ is the adjacency matrix for the cooperative control strategy, the per-unit current information from $(i-1)$th and $(i+1)$th agents communicated to the PI controller is processed as

$$x_{i-1\to i}^{pu} = \left(x_{i-1}^{pu} - x_i^{pu}\right).\mathbb{C}(i, i-1) \tag{4}$$

and

$$x_{i+1\to i}^{pu} = \left(x_{i+1}^{pu} - x_i^{pu}\right).\mathbb{C}(i, i+1), \tag{5}$$

respectively. Let the state variable of the PI controller be x_i^i, then the corresponding ODE is given by

$$\dot{x}_i^i = \left(x_{i-1\to i}^{pu} + x_{i+1\to i}^{pu}.\right).I(i,i). \tag{6}$$

The output, v_i^i, of this layer is given by

$$v_i^i = \left(x_{i-1\to i}^{pu} + x_{i+1\to i}^{pu}\right).P(i,i) + x_i^i, \tag{7}$$

which is passed to the primary control sub-layer.

2) Global voltage regulation sub-layer: If $\mathbb{A}$ is the adjacency matrix for the cooperative control strategy, the voltage estimation information from $(i-1)$th and $(i+1)$th agents is further processed as

$$v_{i-1\to i}^{est} = \left(v_{i-1}^{est} - v_i^{est}\right).\mathbb{A}(i, i-1) \tag{8}$$

and

$$v_{i+1\to i}^{est} = \left(v_{i+1}^{est} - v_i^{est}\right).\mathbb{A}(i, i+1) \tag{9}$$

respectively. This voltage estimate is then passed through an integrator, with the state variable v_i^{esti}, such that

$$\dot{v}_i^{esti} = \left(v_{i-1\to i}^{est} + v_{i+1\to i}^{est}\right). \tag{10}$$

In the noise-cancellation module, the ith agent shares the estimate information of the voltage deviation w_i^{est} with its immediate neighbors. The actual voltage deviation for the ith agent is

$$w_i = \left(v_i^{est} - v_i^{out}\right). \tag{11}$$

If $\mathbb{B}$ is the adjacency matrix for the cooperative control strategy, the information about the estimate of the voltage deviation from $(i-1)$th and $(i+1)$th agents is

$$w_{i-1\to i}^{est} = \left(w_{i-1}^{est} - w_i^{est}\right).\mathbb{B}(i, i-1) \tag{12}$$

and

$$w_{i+1\to i}^{est} = \left(w_{i+1}^{est} - w_i^{est}\right).\mathbb{B}(i, i+1), \tag{13}$$

respectively. This estimate is passed through an integrator, with the state variable w_i^{esti}, such that

$$\dot{w}_i^{esti} = \left(w_{i-1\to i}^{est} + w_{i+1\to i}^{est}\right). \tag{14}$$

The estimate for the voltage deviation, w_i^{est}, is

$$w_i^{est} = w_i + w_i^{esti}. \tag{15}$$

This estimate is then passed to a second integrator with a gain K of dimension $N \times N$, and with the state variable w_i^{estii},

$$\dot{w}_i^{estii} = w_i^{est}. \tag{16}$$

The average voltage of the microgrid as estimated by the ith agent, based on the neighbor information, is

$$v_i^{avg} = v_i^{est} = v_i^{esti} + v_i^{out} - w_i^{estii}.K(i,i). \tag{17}$$

This sub-layer has a PI controller with parameters $P(i,i,)$ and $I(i,i)$. The difference between the global reference voltage and the global average voltage as determined by the ith agent is passed through this PI controller. Let the state variable for PI controller be denoted by v_i^{avgi}, then the ODE is given by

$$\dot{v}_i^{avgi} = (v_i^{ref} - v_i^{avg}).I(i,i). \tag{18}$$

The voltage regulation term at the controller output is

$$v_i^{greg} = v_i^{avgi} + (v_i^{ref} - v_i^{avg}).P(i,i). \tag{19}$$

3) Primary control sub-layer: There is a PI controller with parameters P_{mc} and I_{mc}, and a transfer function $\mathbb{T}_2$. The output of $\mathbb{T}_2$ is denoted by y_i^{mc}. The input u_i^{mc} is

$$u_i^{mc} = v_i^{ref} + v_i^i + v_i^{greg}. \tag{20}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

The expression for v_i^i and v_i^{greg} are given by (7) and (19), respectively. The ODE for the state variable x_i^{pi} associated with the PI controller is given by

$$\dot{x}_i^{pi} = (y_i^{mc} - y_i^{vo}) .I_{mc}. \tag{21}$$

The output of this sub-layer, y_i, is given by

$$y_i = P_{mc}.(y_i^{mc} - y_i^{vo}) + x_i^{pi}, \tag{22}$$

that drives the MOSFET of the ith converter. The cyber layer has two topologies, i.e., `controli_1` and `controli_0`, as shown in Fig. 2, to generate control signal, $control_i$. It may evaluate to $controli == 1$ and $controli == 0$, that correspond to 'ON' and 'OFF' pulses for the MOSFET, respectively. The hybrid automaton generates $controli == 1$ in `controli_1`, and $controli == 0$ in `controli_0`. The ODEs developed for the cyber layer of DC microgrid above are used to describe the continuous dynamics for the two topologies. The switching logic, $logic_i$, is formulated using (22), the elapsed time t_i, and the time period T_i of the ith agent. This is implemented in the hybrid automaton model as a guard to enforce the discrete transition from topology `controli_1` to the topology `controli_0`, hence generating control signal $controli == 0$. Whereas, transition from `controli_0` to `controli_1` is entirely dependent upon the time period T_i that forms the corresponding guard to ensure a periodic switching. This transition is enforced by the guard $t_i \geq T_i$, hence generating control signal $controli == 1$.

4) Instantiation of the cyber layer: The instantiation of the hybrid automation model for the cyber layer of the ith agent, as per Definition 2.1, is

- Two topologies are denoted by $Q = \{q_4, q_5\}$.
- The continuous state vector is $X = X_l \cup X_g$, where, $X_l = \{x_i^i, v_i^{esti}, w_i^{esti}, v_i^{avgi}, x_i^{pi}\}$ and $X_g = \{x_i^{pu}, v_i^{est}, w_i^{est}, i_i^{out}, v_i^{out}, x_{i+1}^{pu}, v_{i+1}^{est}, w_{i+1}^{est}, x_{i-1}^{pu}, v_{i-1}^{est}, w_{i-1}^{est}, control_i\}$.
- $E = \{(q_4, q_5, g_{45}, x'), (q_5, q_4, g_{54}, x')\}$ defines the feasible discrete transitions, e.g., (q_5, q_4, g_{54}, x') means a discrete transition from the topology q_5 to q_4 is allowed, if the guard $g_{54} = \{(t_i \geq T_i)\}$ is satisfied and the continuous state is reset to a new value x'.
- Guard set, for the corresponding elements of E, is defined by $G = \{(logic_i \leq 0), (t_i \geq T_i)\}$.
- The continuous flow trajectory is given by ODEs in (6), (10), (14), (16), (18), and (21) for both topologies.

The control layer and the physical layer both interact with each other and exchange $control_i$ and X_i^{out} as shown in Fig. 2, where $X_i^{out} = [v_i^{out}, i_i^{out}]^T$ and $control_i$ drives the switching in the physical layer. A 50 μs fixed time-step for the numerical solver in the Simulink, and 4 μs sampling time are used in the dSPACE platform.

C. Hybrid Input/Output Automata Conditions

The closed-loop control systems are modeled using hybrid input/output automata (HIOA), to form as a singleton hybrid automaton [38]. Here, the converter and the controller are modeled as two hybrid automata, interacting with each other in a parallel composition, provided that their local variables are disjoint from each other and the two automata are compatible.

Definition 2.2: Let $\mathcal{H}_{ip}$ and $\mathcal{H}_{ic}$ denote the hybrid automata of the converter and the controller for the ith agent, respectively. They are *compatible* if they meet following three conditions

1) $In_{ip} \subseteq O_{ic} \cup O_{(i+1)p} \cup O_{(i-1)p}$,
2) $In_{ic} \subseteq O_{ip} \cup O_{(i+1)c} \cup O_{(i-1)c}$, and
3) $O_{ip} \cap O_{ic} \cap O_{(i+1)c} \cap O_{(i+1)p} \cap O_{(i-1)c} \cap O_{(i-1)p} = \varnothing$.

Subscripts p and c denote the plant (i.e., converter) and the controller, respectively.
The corresponding input and output variables, for the ith agent, are

$$\begin{cases} In_{ip} = \{control_i\}, \\ In_{ic} = \{v_i^{out}, i_i^{out}, X_{i-1}, X_{i+1}\}, \\ O_{ip} = \{v_i^{out}, i_i^{out}\}, \\ O_{ic} = \{control_i, X_i\}. \end{cases} \tag{23}$$

The output variables for the $(i+1)$th and $(i-1)$th agents are

$$\begin{cases} O_{(i+1)p} = \{v_{i+1}^{out}, i_{i+1}^{out}\}, \\ O_{(i+1)c} = \{control_{i+1}, X_{i+1}\}, \\ O_{(i-1)p} = \{v_{i-1}^{out}, i_{i-1}^{out}\}, \\ O_{(i-1)c} = \{control_{i-1}, X_{i-1}\}. \end{cases} \tag{24}$$

It is obvious that the ith agent (comprising converter and controller) meets the compatibility conditions of Definition 2.2, and a parallel composition can be formed. For the ith agent, the parallel composition is $\mathcal{H}_i = \mathcal{H}_{ip} \parallel \mathcal{H}_{ic}$. The DC microgrid is a parallel composition of N agents, i.e, $\mathcal{H}_1 \parallel \mathcal{H}_2 \parallel \cdots \parallel \mathcal{H}_i \parallel \cdots \parallel \mathcal{H}_N \parallel \mathcal{H}_1$, where $\forall\ i\ \mathcal{H}_i = \mathcal{H}_{ip} \parallel \mathcal{H}_{ic}$.

III. FDIA Detection Using Hynger

A. FDIA Scenario Formulation

In cyber-physical DC microgrids, the information among the agents is shared through the global variables (e.g., X_i) that are vulnerable to the FDIA. An FDIA mixes the original data/measurements vector with a malicious vector. The intruder may target the global variables and the sensors data to disturb the consensus procedure, as will be demonstrated in Section IV. In an unconstrained scenario, the intruder has access to all the global variables, and may randomly select some (or all). Under constrained FDIA, the intruder has limited access to one or a few global variables, and formulates the FDIA vector to target these. Let $X_g \in \mathbb{R}^k$ be the vector containing the global variables. FDIA vector $W \in \mathbb{R}^k$ may be formulated to obtain the compromised vector $Z = X_g + \alpha W$, where α is a real valued multiplicative factor that defines the *weight* of the FDIA vector. Each element of the FDIA vector is denoted by w_i, such that a nonzero entry signifies that the corresponding global variable in X_g is targeted. For unconstrained FDIA, all elements of $W \in \mathbb{R}^k$ are nonzero.

B. Hynger - An Overview

Hynger is a MATLAB-based software tool to produce invariants for cyber-physical systems modeled using SLSF. Hynger uses MATLAB's application program interfaces (APIs) to interact with SLSF models during simulations [30], and inserts instrumentation points for selected state variables. Instrumentation points may be regarded as the observation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

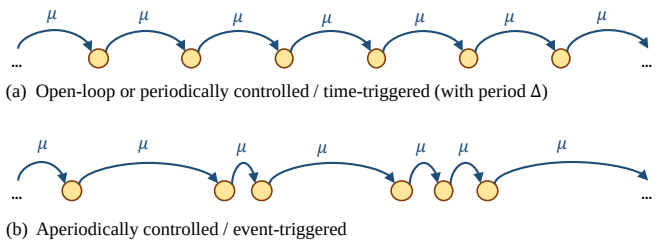

Fig. 4. The instrumentation points are added by Hynger into open-loop, periodically or aperiodically controlled SLSF models.

points to record state variable values at each simulation time-step. Hynger can instrument both open-loop or periodic, and aperiodic control actions, μ, as shown in Fig. 4. The instrumentation points are inserted into the SLSF diagram using function calls through following two types of callbacks:

1) Precondition Callback: It is called before a Simulink block output method executes, i.e., the valuation of the state variable is recorded before the Simulink block execution. Hence, the state valuation is recorded at time t.

2) Postcondition Callback: It is called after a Simulink block output method executes, i.e., the valuation of a state variable is recorded after the Simulink block execution. Hence, the valuation of the state is recorded at time $t + \delta$, where δ is the simulation time-step.

As the SLSF diagram is simulated using Hynger, these instrumentation callback functions automatically insert the instrumentation points to generate a trace file format compatible with Daikon, a dynamic analysis tool used to generate likely invariants for software programs [31]. The analysis performed on a software program by actually executing it on a host processor is called the *dynamic analysis*. As the computational overhead for Hynger grows linearly with the number of monitored state variables [30], the user can select fewer state variables for monitoring (e.g., the output voltages and currents in DC microgrid) to reduce the computational overhead. Moreover, instead of selecting the entirety of the Simulink model for instrumentation, the user can select fewer Simulink blocks to further reduce the performance overhead.

C. FDIA Detection Framework

This framework involves inferring and checking sets of invariants to determine if an FDIA is underway. While this builds on the Hynger tool, extensions will be required to execute the tool and analyze results at runtime. The FDIA detection framework is shown in Fig. 5. A CPS model is provided as an SLSF diagram $\mathcal{A}$. The SLSF diagram is instrumented (denoted as $\widehat{\mathcal{A}}$) using the Hynger tool, and is executed to generate a set of sampled, finite-precision traces $\mathscr{T}$ for given initial condition $\theta \in \Theta$. This adds instrumentation points for every input and output signal in the SLSF diagram. These generated traces are in Daikon compatible format that are passed on to Daikon, and analyzed to generate a set of candidate invariants $\hat{\Phi}$. However, Hynger/Daikon combination provides only the candidate invariants when used as standalone invariant generation tool. Each element $\hat{\varphi} \in \hat{\Phi}$ is then checked as actual invariant using the reachability analysis. The SpaceEx reachability analysis tool [33] is used to obtain the actual invariants. Changes in $\hat{\Phi}$ over time indicates an FDIA.

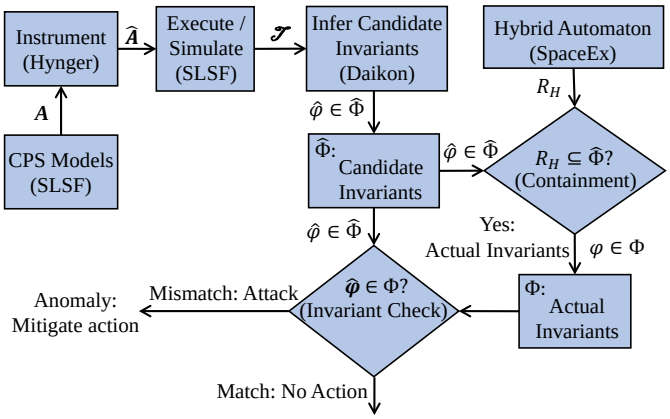

Fig. 5. FDIA detection framework using Hynger/Daikon to infer the candidate invariants and using SpaceEx reachability analysis to generate the reach sets.

For a given formal hybrid automaton $\mathcal{H}$ of a CPS, following definitions are introduced to extract the actual invariants from candidate invariants, as shown in Fig. 5:

Definition 3.1: For a hybrid automaton $\mathcal{H}$, all states encountered during executions are called the *reachable states* of $\mathcal{H}$. A state is already defined in Definition 2.1. Since the exact set of all reachable states is undecidable, reachability analysis tools compute the overapproximated sets of reachable states (called the *reach sets* for simplicity). In this work, SpaceEx [33] is used to compute the reach sets for a formal hybrid automaton $\mathcal{H}$, denoted by $\mathcal{R}_{\mathcal{H}}$.

Definition 3.2: The *property* ρ of a hybrid automaton $\mathcal{H}$ is defined as a Boolean-valued expression, that contains some or all state variables of $\mathcal{H}$, and evaluates to $True$ or $False$.

Definition 3.3: For a hybrid automaton $\mathcal{H}$, a state s is said to *satisfy* the property ρ (i.e., $s \vDash \rho$) if ρ evaluates to $True$ when all state variables are assigned values as defined by the state s.

Definition 3.4: For a hybrid automaton $\mathcal{H}$, a property ρ is an *invariant* of $\mathcal{H}$ if all its reach sets satisfy ρ, i.e., $\mathcal{R}_{\mathcal{H}} \vDash \rho$. A candidate invariant $\hat{\varphi} \in \hat{\Phi}$ is also a property of $\mathcal{H}$. Therefore, Definition 3.4 infers the actual invariants $\varphi \in \Phi$.

Definition 3.5: A candidate invariant $\hat{\varphi} \in \hat{\Phi}$ of a hybrid automaton $\mathcal{H}$ is the *actual invariant* $\phi \in \Phi$ iff $\mathcal{R}_{\mathcal{H}} \vDash \hat{\varphi} \in \hat{\Phi}$.

The candidate invariants for DC microgrids are obtained from Hynger in forms of bounds over the continuous state variables, and denoted as $[\mathcal{B}_l,\ \mathcal{B}_u]$. It is assumed that SLSF model depicts the hybrid automaton so that Hynger can find the set of candidate invariants $\hat{\Phi}$. Each $\hat{\varphi} \in \hat{\Phi}$ is then examined to ascertain whether it is an actual invariant as per Definition 3.5, i.e., checking whether $\mathcal{R}_{\mathcal{H}} \subseteq \hat{\varphi}$ holds.

The FDIA tends to disturb the consensus and hence the invariants as shown in case studies in Section IV. This change is employed to detect FDIA on DC microgrids.

Definition 3.6: A hybrid automaton $\mathcal{H}$ is said to be operating under FDIA scenario iff $\hat{\varphi} \notin \Phi$.

IV. Case Studies

A small-scale DC microgrid prototype is shown in Fig. 6, with the system parameters given in the Appendix. A comparison of the SLSF model simulation and the experimental data

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

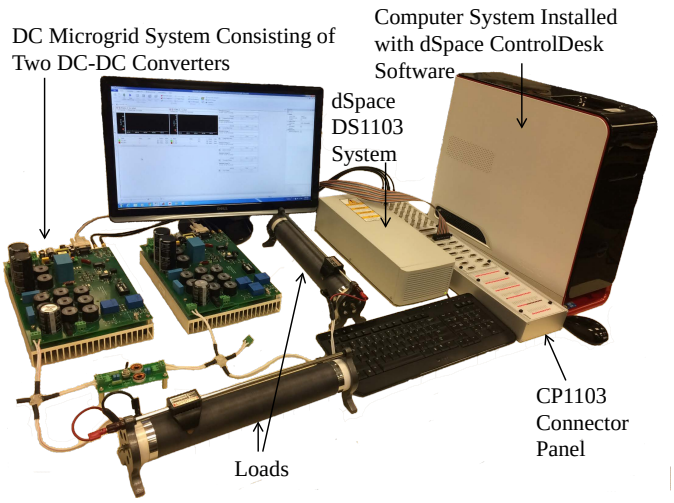

Fig. 6. Experimental setup for a DC microgrid consisting of two dc-dc converters and a dSpace DS1103 controller system.

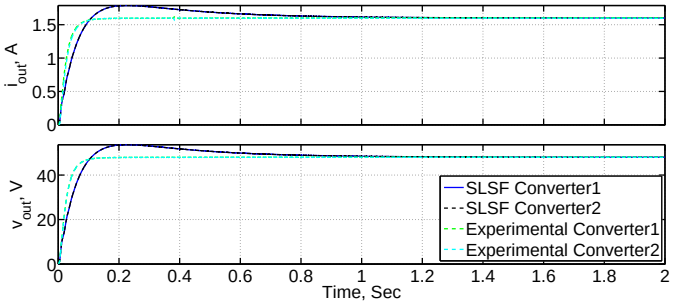

Fig. 7. SLSF simulation and experimental results for DC microgrid under no FDIA scenario showing stable output current and output voltage.

is shown in Fig. 7, for a stable output under no FDIA scenario. The effects of constrained FDIA on global variables, e.g., v_2^{est}, and w_2^{est} are shown in Fig. 8 and Fig. 9, respectively. The intruder may also disturb the consensus protocol when the current and voltage sensors are targeted as shown in Fig. 10 and Fig. 11, respectively. Unconstrained FDIA that involves targeting the entire set of global variables, is very effective in destabilizing the DC microgrid, as shown in Fig. 12.

A. FDIA Detection

For FDIA detection, the SLSF model formed using the methodology in Section II is instrumented using Hynger, and then simulated within the SLSF environment to generate traces under no FDIA scenario. This process generates the trace

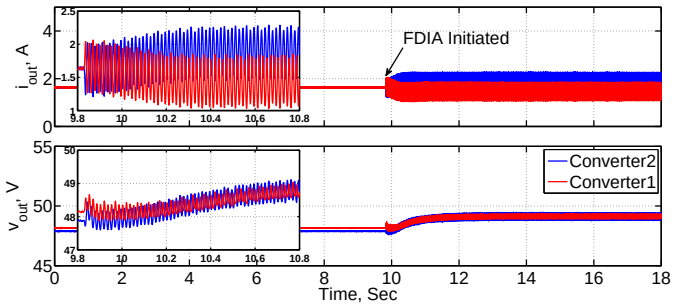

Fig. 8. Experimental data for the constrained FDIA, targeting v_2^{est}.

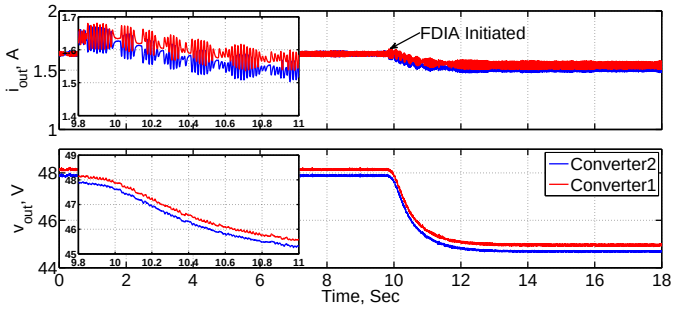

Fig. 9. Experimental data for the constrained FDIA, targeting w_2^{est}.

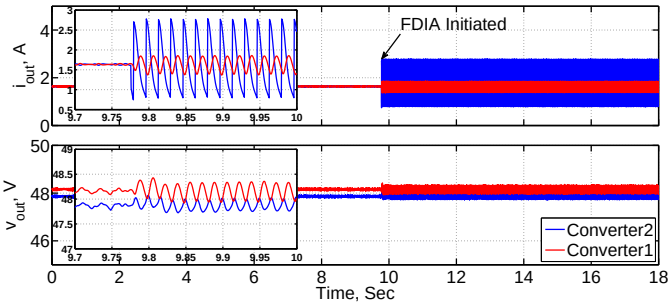

Fig. 10. Experimental data for the constrained FDIA targeting current sensor.

files, in Daikon compatible format, that are passed on to Daikon. Hence, the corresponding invariants are generated automatically, and shown in Table I. The SpaceEx reachability analysis tool computes the reach sets in the steady state, as seen in Fig. 13. It is shown that the experimental data and the simulation traces are contained within the reach sets. Moreover, reach sets satisfy the candidate invariants generated using Hynger under no FDIA scenario. Therefore, the invariants without FDIA of Table I are found to be the actual invariants as per Definition 3.5.

Next, FDIA detection approach is tested when the adversary breaks into the communication link from agent 2 to agent 1. A false data signal is spoofed into x_2^{pu}, at time $t = 0.6\ s$, through the compromised communication link. x_2^{pu} is the per-unit current information of agent 2 that is communicated to agent 1. The DC microgrid under FDIA scenario is again instrumented using Hynger, and simulated in the SLSF environment to generate traces and the corresponding invariants. The output of the instrumented model under FDIA is plotted in Fig. 14 for both agents 1 and 2. It can be observed that the consensus protocol is disturbed under FDIA.

The corresponding invariants for the DC microgrid under

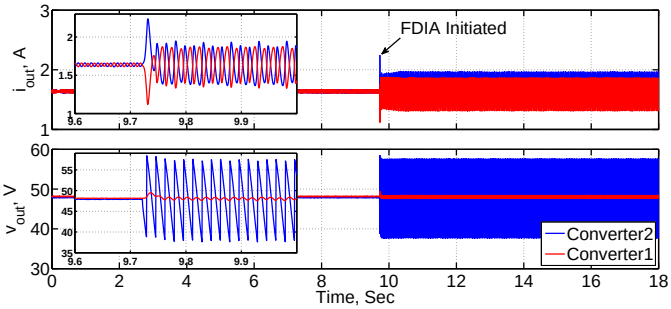

Fig. 11. Experimental data for the constrained FDIA targeting voltage sensor.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

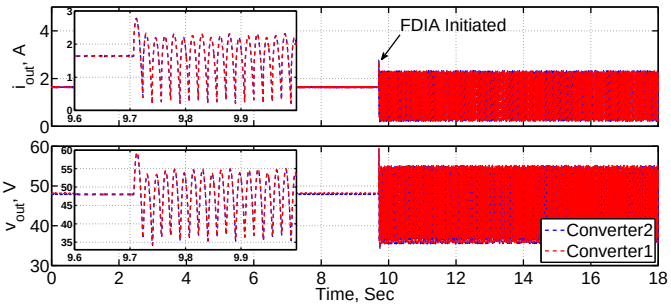

Fig. 12. Experimental data plots for the DC microgrid, under unconstrained FDIA, targeting the entire set of global variables.

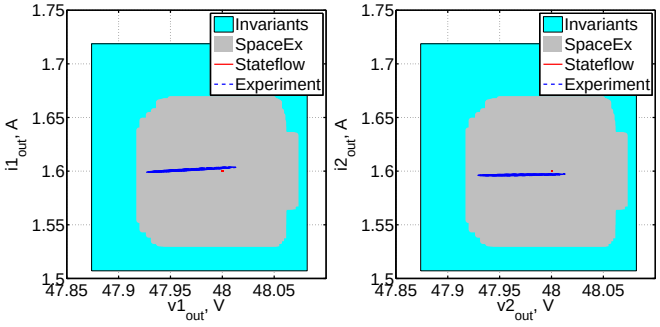

Fig. 13. Phase-portrait comparison of Hynger generated invariants, SpaceEx, SLSF, and experimental results, in the steady state, for DC microgrid under normal conditions (i.e., without FDIA). The experimental and simulation results are contained within the reach sets computed using SpaceEx. Moreover, it is also shown that the SpaceEx reach sets satisfy the invariants.

FDIA scenario are generated automatically using Hynger. This invariant set is then compared with the actual invariants, i.e., invariants under no FDIA scenario to detect intrusion. A comparison of the invariants with and without FDIA scenario is tabulated in Table I. It is evident by comparison that FDIA detection condition mentioned in Definition 3.6, i.e., $\hat{\varphi} \notin \Phi$, is satisfied for the two scenarios, detecting the FDIA.

B. FDIA Mitigation Strategies

Once an FDIA is detected, various mitigation strategies can suppress the effects of the attack. As an example, three possible mitigation strategies are experimentally demonstrated.

1) Physical mitigation strategy: The affected converter may be taken offline after an FDIA is detected. Once the affected converter 2 is disconnected, proper microgrid operation is restored, as shown in Fig. 15.

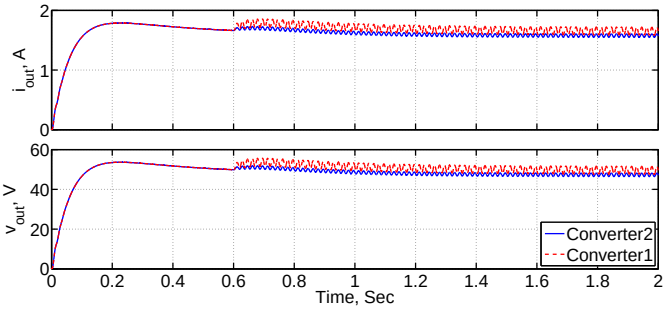

Fig. 14. The SLSF model of DC microgrid is instrumented using Hynger, and the simulation output results for the instrumented model under FDIA scenario are shown demonstrating that the consensus protocol is disturbed. These instrumented traces are passed on to Daikon to generate invariants.

TABLE I
INVARIANTS WITHOUT AND WITH FDIA

Variable	Without FDIA	With FDIA
v_1^{out}	$[47.874, 48.0818]$	$[47.9917, 48.0486]$
v_2^{out}	$[47.8739, 48.0818]$	$[47.9917, 48.5258]$
i_1^{out}	$[1.5071, 1.7187]$	$[1.418, 1.6016]$
i_2^{out}	$[1.5071, 1.7187]$	$[1.5997, 1.6175]$

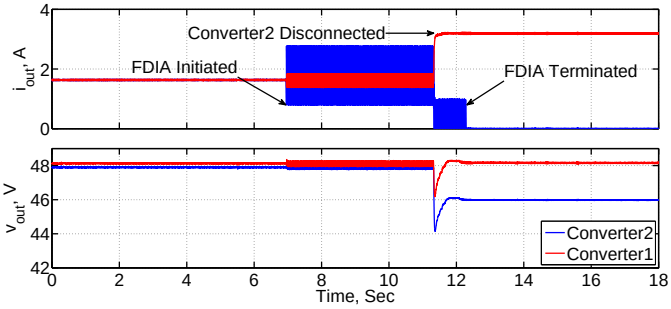

Fig. 15. Experimental data for the constrained FDIA targeting the current sensor of converter 2. The affected converter is disconnected to stabilize the DC microgrid.

2) Communication-based mitigation strategy: The communication link of the effected agent (converter) can be disconnected so that other agents may not be effected. Once the communication link between the affected converter 2 and non-affected converter 1 is disconnected, the output of converter 1 is stabilized, as shown in Fig. 16. The output of converter 2 still remains unstable.

3) Control-based mitigation strategy: One can use a modified control scheme to reduce the effects of FDIA, by augmenting the controller with a false data suppressing mechanism (e.g., filters [39]). As shown in Fig. 17, FDIA is initiated at about $8.5\ s$, and the modified control scheme is put into action at about $11.97\ s$ to suppress FDIA effects, and the output of the entire DC microgrid is stabilized.

C. Stealthy Attacks with Minimal Weights

The intruder could potentially fabricate an attack vector to bypass the proposed FDIA detection framework, if the changes in candidate invariants, and microgrid operation, are negligible. This is demonstrated through the following two

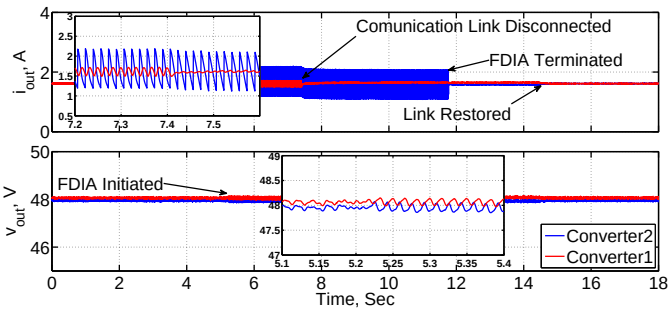

Fig. 16. Experimental data for the constrained FDIA targeting the current sensor of converter 2. The communication link between the affected converter and non-affected converter 1 is disconnected to stabilize converter 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2656905, IEEE Transactions on Industrial Informatics

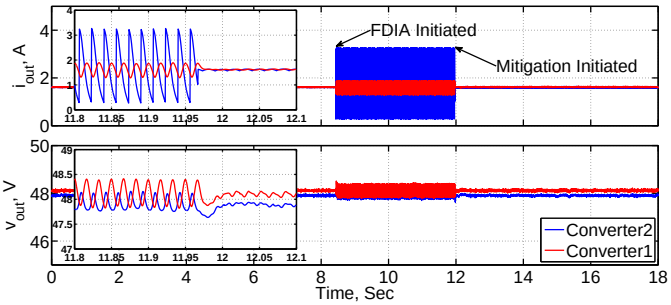

Fig. 17. Experimental data for the constrained FDIA targeting the current sensor of converter 2. As FDIA is detected, the control strategy is augmented with a false data suppression mechanism. It is shown that this controller-based mitigation action has stabilized the entire DC microgrid output.

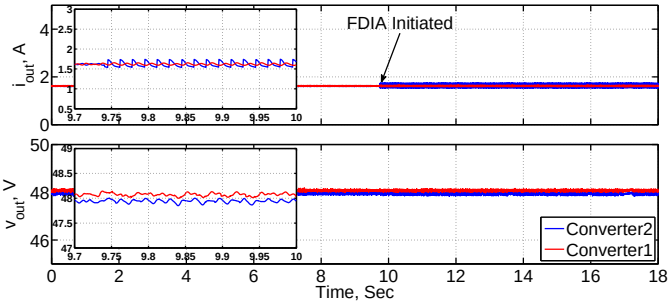

Fig. 18. FDIA, targeting the current sensor of converter 2, with the minimal-weight attack vector that can be detected using this framework.

experiments. First, an attack vector with small weights is designed that can be detected using the proposed framework. The invariants for the output current generated using Hynger are $i_1^{out} = [1.55, 1.77]$ and $i_2^{out} = [1.55, 1.77]$. These invariants are deviated from the corresponding actual invarinats tabulated in Table I, indicating the presence of an FDIA. The negative effects of this FDIA are shown in Fig. 18. It is demonstrated that an FDIA with such minimal destabilizing effects can still be detected using the proposed framework. Next, an attack vector with smaller weights is fabricated to bypass through this FDIA framework. The invariants for the output current generated using Hynger are $i_1^{out} = [1.5071, 1.7187]$ and $i_2^{out} = [1.5071, 1.7188]$ that are comparable with the actual invariants in Table I, hence missing the FDIA. However, the negative effects of this FDIA are negligible, as seen in Fig. 19, as they do not disturb the microgrid operation.

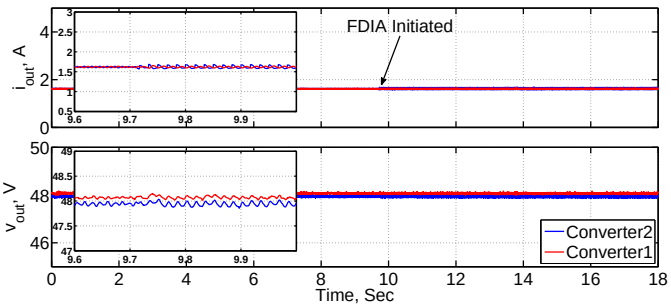

Fig. 19. FDIA, targeting the current sensor of converter 2, with extra minimal-weight attack vector that can bypass the proposed FDIA detection framework. As seen, the effects of FDIA are negligible and do not affect stability.

V. Conclusion

FDIA disturbs the consensus protocols used in the distributed control of cyber-physical DC microgrids. An FDIA detection framework is presented whereby the attack detection problem is formalized as identifying a change in the set of candidate invariants. The candidate invariants are generated using Hynger, that provides an interface between SLSF models and the Daikon tool, which is an invariant inference tool. The hybrid automaton of cyber-physical DC microgrid is presented to obtain the reach sets through reachability analysis. Moreover, the SLSF model of a DC microgrid is also developed to generate the candidate invariants. The actual invariants are obtained after verifying whether the reach sets are contained within the candidate invariants. The candidate invariants generated by Hynger are compared with the actual invariants to successfully detect FDIA.

Appendix

Buck converter parameters are $L = 2.64\ mH, C = 2.2\ mF$, and $F_s = 60\ kHz$. The local loads are $R_1 = R_2 = 30\ \Omega$. The transfer functions are given by:

$$\mathbb{T}_1 = \frac{1}{0.01s + 1},\ \mathbb{T}_2 = \frac{1}{0.05s + 1}. \tag{25}$$

The DC microgrid parameters are: $V_{ref} = [48\ \ 48]^T$, $I_{max} = [4\ \ 4]^T$, $V_{in} = [80\ \ 80]^T, P_{mc} = 0.01, I_{mc} = 1$, $\mathbb{A} = 25\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, $\mathbb{B} = \mathbb{A}, \mathbb{C} = 0.5\mathbb{A}, I = 3\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, P = 0.05\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, K = \begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}$.

References

[1] T. Dragičević, X. Lu, J. C. Vasquez, and J. M. Guerrero, "Dc microgrids - part i: A review of control strategies and stabilization techniques," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 4876–4891, July 2016.

[2] ——, "Dc microgrids - part ii: A review of power architectures, applications, and standardization issues," *IEEE Transactions on Power Electronics*, vol. 31, no. 5, pp. 3528–3549, May 2016.

[3] V. Nasirian, S. Moayedi, A. Davoudi, and F. Lewis, "Distributed cooperative control of dc microgrids," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2288–2303, Apr 2015.

[4] L. Meng, T. Dragičević, J. Roldn-Prez, J. C. Vasquez, and J. M. Guerrero, "Modeling and sensitivity study of consensus algorithm-based distributed hierarchical control for dc microgrids," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1504–1515, May 2016.

[5] X. Lu, X. Yu, J. Lai, J. Guerrero, and H. Zhou, "Distributed secondary voltage and frequency control for islanded microgrids with uncertain communication links," *IEEE Transactions on Industrial Informatics, doi:10.1109/TII.2016.2603844*, 2016.

[6] F. Luo, Y. Chen, Z. xu, G. Liang, Y. Zheng, and J. Qiu, "Multi-agent based cooperative control framework for microgrids' energy imbalance," *IEEE Transactions on Industrial Informatics, doi:10.1109/TII.2016.2591918*, 2016.

[7] Z. Jin, G. Sulligoi, R. Cuzner, L. Meng, J. C. Vasquez, and J. M. Guerrero, "Next-generation shipboard dc power system: Introduction smart grid and dc microgrid technologies into maritime electrical netowrks," *IEEE Electrification Magazine*, vol. 4, no. 2, pp. 45–57, June 2016.

[8] Q. Li, C. Peng, M. Chen, F. Chen, W. Kang, J. Guerrero, and D. Abbott, "Networked and distributed control method with optimal power dispatch for islanded microgrids," *IEEE Transactions on Industrial Electronics, doi:10.1109/TIE.2016.2598799*, 2016.

[9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information Systems and Security*, vol. 14, no. 1, pp. 13:1–13:33, Jun. 2011.

[10] P. Srikantha and D. Kundur, "Denial of service attacks and mitigation for stability in cyber-enabled power grid," in *Proceedings of IEEE Innovative Smart Grid Technologies Conference*, Washington, DC, 2015, pp. 1–5.

[11] X. Zhong, L. Yu, R. Brooks, and G. Venayagamoorthy, "Cyber security in smart dc microgrid operations," in *Proceedings of IEEE 1st International Conference on DC Microgrids*, Atlanta, GA, 2015, pp. 86–91.

[12] Z. Lu, W. Wang, and C. Wang, "Camouflage traffic: Minimizing message delay for smart grid applications under jamming," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 31–44, Jan 2015.

[13] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, Dec 2014.

[14] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, Sept 2016.

[15] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *Preprints of the First Workshop on Secure Control Systems, CPSWEEK*, Stockholm, Sweden, 2010.

[16] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, June 2011.

[17] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid, doi: 10.1109/TSG.2015.2495133*, 2016.

[18] M. Talebi, C. Li, and Z. Qu, "Enhanced protection against false data injection by dynamically changing information structure of microgrids," in *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop*, Hoboken, NJ, 2012, pp. 393–396.

[19] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, Apr 2013.

[20] V. Kounev, D. Tipper, A. A. Yavuz, B. M. Grainger, and G. F. Reed, "A secure communication architecture for distributed microgrid control," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2484–2492, Sept 2015.

[21] A. Mazloomzadeh, O. A. Mohammed, and S. Zonouzsaman, "Empirical development of a trusted sensing base for power system infrastructures," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2454–2463, Sept 2015.

[22] L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih, and Z. Han, "Detecting false data injection attacks on power grid by sparse optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, Mar 2014.

[23] S. Li, Y. Yilmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2725–2735, Nov 2015.

[24] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, Sep 2015.

[25] D. Rawat and C. Bajracharya, "Detection of false data injection attacks in smart grid communication systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1652–1656, Oct 2015.

[26] J. Zhao, G. Zhang, M. La Scala, Z. Dong, C. Chen, and J. Wang, "Short-term state forecasting-aided method for detection of smart grid general false data injection attacks," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–11, Oct 2015.

[27] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–9, Aug 2014.

[28] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580–591, Mar 2014.

[29] G. Hug and J. A. Giampapa, "Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, Sep 2012.

[30] T. T. Johnson, S. Bak, and S. Drager, "Cyber-physical specification mismatch identification with dynamic analysis," in *Proceedings of 6th International Conference on Cyber-Physical Systems*, Seattle, WA, 2015, pp. 208–217.

[31] M. D. Ernst, J. Cockrell, W. G. Griswold, and D. Notkin, "Dynamically discovering likely program invariants to support program evolution," *IEEE Transactions on Software Engineering*, vol. 27, no. 2, pp. 99–123, Feb 2001.

[32] M. D. Ernst, J. H. Perkins, P. J. Guo, S. McCamant, C. Pacheco, M. S. Tschantz, and C. Xiao, "The Daikon system for dynamic detection of likely invariants," *Science of Computer Programming*, vol. 69, no. 1, pp. 35–45, Dec. 2007.

[33] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *Proceedings of 23rd International Conference on Computer Aided Verification*, Snowbird, UT, 2011, pp. 379–395.

[34] T. Henzinger, "The theory of hybrid automata," in *Proceedings of IEEE Symposium on Logic in Computer Science*, New Brunswick, NJ, 1996, pp. 278–292.

[35] D. Olivares, A. Mehrizi-Sani, A. Etemadi, C. Canizares, R. Iravani, M. Kazerani, A. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. Jimenez-Estevez, and N. Hatziargyriou, "Trends in microgrid control," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905–1919, Jul 2014.

[36] Q. Shafiee, V. Nasirian, J. C. Vasquez, J. M. Guerrero, and A. Davoudi, "A multi-functional fully distributed control framework for ac microgrids," *IEEE Transactions on Smart Grid, doi:10.1109/TSG.2016.2628785*, 2016.

[37] S. Moayedi, V. Nasirian, F. L. Lewis, and A. Davoudi, "Team-oriented load sharing in parallel dc-dc converters," *IEEE Transactions on Industry Applications*, vol. 51, no. 1, pp. 479–490, Jan 2015.

[38] N. Lynch, R. Segala, and F. Vaandrager, "Hybrid I/O automata," *Information and Computation*, vol. 185, no. 1, pp. 105–157, Aug 2003.

[39] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, part i, ii, and iii," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120–135, Jan 1970.

Omar Ali Beg (S'14) received the B.E. and M.S. degrees in electrical engineering from National University of Sciences and Technology, Pakistan. He is presently working toward his Ph.D. degree in electrical engineering at University of Texas at Arlington, TX, USA. He is recipient of the US Air Force Research Laboratory summer research fellowship 2015. His research interests include the formal verification of software-controlled power electronics devices.

Taylor T Johnson (S'05-M'13) received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2010 and 2013, respectively. He is an Assistant Professor of Electrical Engineering and Computer Science at the Vanderbilt University, Nashville, TN, USA. He received the Air Force Office of Scientific Research Young Investigator Program award in 2016 and the National Science Foundation Computer and Information Science and Engineering Research Initiation Initiative award in 2015. His research interests include developing algorithmic techniques and software tools to improve the reliability of cyber-physical systems.

Ali Davoudi (S'04-M'11-SM'15) received his Ph.D. in electrical and computer engineering from the University of Illinois, Urbana-Champaign, IL, USA, in 2010. He is currently an Associate Professor in the Electrical Engineering Department, University of Texas, Arlington, TX, USA. His research interests include various aspects of modeling and control of power electronics and finite-inertia power systems. Dr. Davoudi is an Associate Editor for IEEE Transactions on Transportation Electrification and IEEE Transactions on Energy Conversion. He has received 2014 Ralph H. Lee Prize paper award from IEEE Transactions on Industry Applications, best paper award from 2015 IEEE International Symposium on Resilient Control Systems, 2014-2015 best paper award from IEEE Transactions on Energy Conversion, and 2016 Prize Paper Award from the IEEE Power and Energy Society.