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

SLANDED 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 surfacing, and protocol hopping [14]. In this work, detection of FDIA in power electronics-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 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 GEneratorR) [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. 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 $v_{i_{i_{out}}}$ and output current $i_{i_{out}}$ of the $i^{th}$ 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 $\mathcal{H} = (Q, X, \Theta, U, F, T, E, G, inv)$:

- $Q = \{q_1, q_2, \ldots, q_N\}$ 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. $X = X_g \cup X_t$, such that $X_g$ is the set of global variables and $X_t$ is the set of local
variables. Further, $X_q = I_n \cup O$, where $I_n$ 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.
- $U = \{u_1, u_2, \ldots, u_N\}$ 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 $\mathbb{R}^n$.
- $T$ 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 $\forall \tau(q, x) \in T, \exists s \in \tau(q, x)$ that satisfies $inv(q)$ (i.e., $s \in \tau(q, x) \models inv(q)$). A continuous flow trajectory is given by

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

$E$ is a finite set of feasible discrete transitions allowed among the topologies. It is defined by a tuple $e = (q, q', g, x')$, 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 $\forall e \in G$. A guard must be satisfied by a state to take a discrete transition from a given topology to another. A state $s = (q, val(x))$ satisfies $g$ (i.e., $s \models g$) iff $q_k = q_i \in e = (q_1, q_2, q_3, x')$ 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 $s \models inv(q)$ 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 $H$ is defined by its execution, $e$. An execution is defined as a sequence of states, $e = s_0, s_1, s_2, \ldots$, obtained as a result of continuous flow trajectories and discrete transitions.

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

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

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

- Three topologies, shown in Fig. 2, are denoted by $Q = \{q_1, q_2, q_3\}$.
- $X = \{i_1^L, v_{\text{in}}^L, v_{\text{out}}^L, \text{control}_i\}$, where $X_i = \{i_1^L, v_{\text{in}}^L\}$ and $X_g = \{v_{\text{out}}^L, \text{control}_i\}$.
- $U = \{[0, 0, 0, 0]^T, [0, 0, 0, 0]^T, [0, 0, 0, 0]^T\}$ forms the input vector set.
- $E = \{(q_1, q_2, q_1, x'), (q_2, q_1, q_2, x'), (q_2, q_3, q_2, x'), (q_3, q_1, x')\}$ defines the feasible discrete transitions, e.g., $(q_2, q_3, q_2, x')$ means that a discrete transition from topology $q_2$ to $q_3$ is allowed, if the guard $q_3 = \{(i_1^L \leq 0)\}$ is satisfied and the continuous state is reset to $x'$.

- Guard set, for the corresponding elements of $E$, is defined by $G = \{(control_i = 0), (control_i = 1), (i_1^L \leq 0), (control_i = 1)\}$. 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 $u_1 = [v_{in}^1, 0, 0, 0]^T$ 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, $q_12 = \{control_i = 0\}$ 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 $u_2 = [0, 0, 0, 0]^T$.

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 consider 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 $G$ is defined as a pair (tuple) of a set of vertices and edges, i.e., $G = (\Lambda, \varepsilon)$. Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ define a set of $N$ vertices (nodes), and $\varepsilon \subseteq \Lambda \times \Lambda$ a set of edges. An edge from node $\lambda_i$ to $\lambda_j$ 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 $A = [a_{ij}]$ with weights $a_{ij} > 0$ if $(\lambda_i, \lambda_j) \in \varepsilon$, and $a_{ij} = 0$ otherwise. The local control protocol, $u_i$ for each agent $i$ is

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

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.
The global voltage reference for \( N \) agents is defined as \( V_{\text{ref}} = [v_{\text{ref},1}, v_{\text{ref},2}, \ldots, v_{\text{ref},N}]^T \), input DC voltage as \( V_{\text{in}} = [v_{\text{in},1}, v_{\text{in},2}, \ldots, v_{\text{in},N}]^T \), output DC voltage vector as \( V_{\text{out}} = [v_{\text{out},1}, v_{\text{out},2}, \ldots, v_{\text{out},N}]^T \), output current vector as \( I_{\text{out}} = [i_{\text{out},1}, i_{\text{out},2}, \ldots, i_{\text{out},N}]^T \), and the voltage estimation vector for the voltage observer module as \( V_{\text{est}} = [v_{\text{est},1}, v_{\text{est},2}, \ldots, v_{\text{est},N}]^T \), the per-unit current vector as \( X_{\text{pu}} = [x_{\text{pu},1}, x_{\text{pu},2}, \ldots, x_{\text{pu},N}]^T \), and the estimate of the voltage deviation vector for the noise cancellation module as \( W_{\text{est}} = [w_{\text{est},1}, w_{\text{est},2}, \ldots, w_{\text{est},N}]^T \). Here, \( x_{\text{pu}} \) refers to the loading percentage of the \( i \)th agent. As shown in Fig. 3, \( X_i \) depicts the information vector communicated from the \( i \)th agent to the \((i-1)\)th and \((i+1)\)th agents, such that \( X_i = [x_{\text{pu},1}, x_{\text{pu},2}, \ldots, x_{\text{pu},N}]^T \). Moreover, \( X_{i-1} \) and \( X_{i+1} \) of Fig. 3 are defined similarly. Communication links are modeled as low-pass filters \((T_1 \text{ and } T_2 \text{ in Appendix})\) to emulate delays inherent in the data exchange process, as in [3], [36], [37]. Here, \( i_{\text{out}} \) and \( i_{\text{est}} \) are passed through \( T_1 \) to get the per-unit current \( x_{\text{pu}} \) and the voltage \( v_{\text{est}} \), respectively.

The control sub-layers are discussed next.

1) Proportional load sharing sub-layer: The \( i \)th 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 \( C \) 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

\[
x_{\text{pu},i-1 \rightarrow i} = (x_{\text{pu},i-1} - x_{\text{pu},i}) \cdot C(i, i-1)
\]

(4)

and

\[
x_{\text{pu},i+1 \rightarrow i} = (x_{\text{pu},i+1} - x_{\text{pu},i}) \cdot C(i, i+1),
\]

(5)

respectively. Let the state variable of the PI controller be \( x_i \), then the corresponding ODE is given by

\[
x_i = (x_{\text{pu},i-1 \rightarrow i} + x_{\text{pu},i+1 \rightarrow i}) \cdot I(i, i).
\]

(6)

The output, \( v_i \), of this layer is given by

\[
v_i = (x_{\text{pu},i-1 \rightarrow i} + x_{\text{pu},i+1 \rightarrow i}) \cdot P(i, i) + x_i,
\]

(7)

which is passed to the primary control sub-layer.

2) Global voltage regulation sub-layer: If \( A \) 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

\[
v_{\text{est},i-1 \rightarrow i} = (v_{\text{est},i-1} - v_{\text{est},i}) \cdot A(i, i-1)
\]

(8)

and

\[
v_{\text{est},i+1 \rightarrow i} = (v_{\text{est},i+1} - v_{\text{est},i}) \cdot A(i, i+1)
\]

(9)

respectively. This voltage estimate is then passed through an integrator, with the state variable \( v_{\text{est},i} \), such that

\[
v_{\text{est},i} = (v_{\text{est},i-1 \rightarrow i} + v_{\text{est},i+1 \rightarrow i}).
\]

(10)

In the noise-cancellation module, the \( i \)th agent shares the estimate information of the voltage deviation \( w_{\text{est},i} \) with its immediate neighbors. The actual voltage deviation for the \( i \)th agent is

\[
w_i = (v_{\text{est},i} - v_{\text{est},i}).
\]

(11)

If \( B \) 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

\[
w_{\text{est},i-1 \rightarrow i} = (w_{\text{est},i-1} - w_{\text{est},i}) \cdot B(i, i-1)
\]

(12)

and

\[
w_{\text{est},i+1 \rightarrow i} = (w_{\text{est},i+1} - w_{\text{est},i}) \cdot B(i, i+1),
\]

(13)

respectively. This estimate is then passed through an integrator, with the state variable \( w_{\text{est},i} \), such that

\[
w_{\text{est},i} = (w_{\text{est},i-1 \rightarrow i} + w_{\text{est},i+1 \rightarrow i}).
\]

(14)

The estimate for the voltage deviation, \( w_{\text{est},i} \), is

\[
w_{\text{est},i} = w_i + w_{\text{est},i}.
\]

(15)

This estimate is then passed to a second integrator with a gain \( K \) of dimension \( N \times N \), and with the state variable \( w_{\text{est},i} \),

\[
w_{\text{est},i} = w_{\text{est},i}.
\]

(16)

The average voltage of the microgrid as estimated by the \( i \)th agent, based on the neighbor information, is

\[
v_i^{avg} = v_{\text{est},i} = v_{\text{est},i} + w_{\text{est},i} - w_{\text{est},i} \cdot K(i, i).
\]

(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 \( i \)th agent is passed through this PI controller. Let the state variable for PI controller be denoted by \( v_i^{avg} \), then the ODE is given by

\[
v_i^{avg} = (v_{\text{est},i} - v_i^{avg}) \cdot I(i, i).
\]

(18)

The voltage regulation term at the controller output is

\[
v_i^{reg} = v_i^{avg} + (v_{\text{ref},i} - v_i^{avg}) \cdot P(i, i).
\]

(19)

3) Primary control sub-layer: There is a PI controller with parameters \( P_{\text{mc}} \) and \( I_{\text{mc}} \), and a transfer function \( T_2 \). The output of \( T_2 \) is denoted by \( y_{\text{mc}}^{\text{out}} \). The input \( v_i^{\text{ref}} \) is

\[
y_{\text{mc}} = v_i^{\text{ref}} + v_i^{\text{reg}}.
\]

(20)
The expression for \( v_i^e \) and \( v_i^{reg} \) are given by (7) and (19), respectively. The ODE for the state variable \( x_i^{pl} \) associated with the PI controller is given by

\[
\dot{x}_i^{pl} = (y_i^{mc} - y_i^{vo}) \cdot I_{mc}.
\]  
(21)

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

\[
y_i = P_{mc}(y_i^{mc} - y_i^{vo}) + x_i^{pl},
\]  
(22)

that drives the MOSFET of the ith converter. The cyber layer has two topologies, i.e., \( \text{controli}_1 \) and \( \text{controli}_0 \), as shown in Fig. 2, to generate control signal \( \text{controli} \).

It may evaluate to 'ON' and 'OFF' pulses for the MOSFET, respectively. The hybrid automaton generates \( \text{controli} == 1 \) in \( \text{controli}_1 \), and \( \text{controli} == 0 \) in \( \text{controli}_0 \). 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, \( \text{logic}_i \), 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 \( \text{controli}_1 \) to the topology \( \text{controli}_0 \), hence generating control signal \( \text{controli} == 0 \). Whereas, transition from \( \text{controli}_0 \) to \( \text{controli}_1 \) 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 \( \text{controli} == 1 \).

4) Instantiation of the cyber layer: The instantiation of the hybrid automaton 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 \( X = X_l \cup X_r \), where, \( X_l = \{x_i^{pl}, v_i^{est}, w_i^{est}, w_i^{reg}, v_i^{avg}, x_i^p \} \) and \( X_r = \{x_i^{est}, v_i^{est}, w_i^{est}, v_i^{out}, v_i^{pu}, x_i^{est}, w_i^{est}, w_i^{pu}, x_i^{out}, \text{controli}_i \} \).
- \( E = \{(q_4, q_5, q_4, x'), (q_5, q_4, q_5, x') \} \) defines the feasible discrete transitions, e.g., \( (q_5, q_4, q_5, x') \) means a discrete transition from the topology \( q_4 \) to \( q_5 \) is allowed, if the guard \( g_{q_5} = \{(t_i \geq T_i)\} \) is satisfied and the continuous state is reset to a new value \( x' \).
- Guard set, for the corresponding elements of \( E \), is defined by \( G = \{\text{logic}_i \leq 0\}, (t_i \geq T_i)\} \).
- 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 \( \text{controli} \) and \( X_i^{out} \) as shown in Fig. 2, where \( X_i^{out} = [v_i^{out}, q_i^{out}]^{T} \) and \( \text{controli}_i \) drives the switching in the physical layer. A 50 \( \mu \)s fixed time-step for the numerical solver in the Simulink, and 4 \( \mu \)s 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 \( H_{ip} \) and \( H_{ic} \) denote the hybrid automata of the converter and the controller for the \( i \)th agent, respectively. They are compatible if they meet following three conditions

1) \( H_{ip} \subseteq O_{ic} \cup O_{i+1} \cup O_{i-1} \),
2) \( H_{ic} \subseteq O_{ip} \cup O_{i+1} \cup O_{i-1} \),
3) \( O_{ip} \cap O_{ic} \cap O_{i+1} \cap O_{i-1} = \emptyset \).

Subscripts \( p \) and \( c \) denote the plant (i.e., converter) and the controller, respectively.

The corresponding input and output variables, for the \( i \)th agent, are

\[
\begin{align*}
H_{ip} & = \{\text{controli}_1\}, \\
H_{ic} & = \{v_i^{out}, v_i^{out}, x_{i-1}, x_{i+1}\}, \\
O_{ip} & = \{v_i^{out}, v_i^{out}\}, \\
O_{ic} & = \{\text{controli}_1, X_i\}.
\end{align*}
\]  
(23)

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

\[
\begin{align*}
O_{i+1} & = \{v_{i+1}^{out}, v_{i+1}^{out}\}, \\
O_{i+1} & = \{\text{controli}_1, X_{i+1}\}, \\
O_{i-1} & = \{v_{i-1}^{out}, v_{i-1}^{out}\}, \\
O_{i-1} & = \{\text{controli}_1, X_{i-1}\}.
\end{align*}
\]  
(24)

It is obvious that the \( i \)th agent (comprising converter and controller) meets the compatibility conditions of Definition 2.2, and a parallel composition can be formed. For the \( i \)th agent, the parallel composition is \( H_i = H_{ip} \parallel H_{ic} \). The DC microgrid is a parallel composition of \( N \) agents, i.e., \( H_1 \parallel H_2 \parallel \cdots \parallel H_i \parallel \cdots \parallel H_N \parallel H_1 \), where \( \forall i \ H_i = H_{ip} \parallel H_{ic} \).

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., \( X \)) that are vulnerable to the FDIA. An FDIA mixes the original data/measurement vector. Each element of the FDIA vector is denoted by \( W_{ip} \) and is targeted. For unconstrained FDIA, all \( W_{ip} \) are nonzero. Under constrained FDIA, the intruder has limited access to one or a few global variables, and formulates the FDIA vector to target these. Let \( X_g \in R^k \) be the vector containing the global variables. FDIA vector \( W \in R^k \) may be formulated to obtain the compromised vector \( Z = X_g + \alpha W \), where \( \alpha \) is a real valued multiplicitive 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 \( W \in R^k \) 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...
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, \( \mu_i \), 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 \( \delta \) 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 \( \mathcal{A}_i \)) using the Hynger tool, and is executed to generate a set of sampled, finite-precision traces \( \mathcal{T} \) 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 \( \rho \) 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 \( \rho \) (i.e., \( s \models \rho \)) if \( \rho \) 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 \( \rho \) is an invariant of \( \mathcal{H} \) if all its reach sets satisfy \( \rho \), i.e., \( \mathcal{R}_\mathcal{H} \models \rho \). 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 \( \varphi \in \Phi \) iff \( \mathcal{R}_\mathcal{H} \models \hat{\varphi} \). The candidate invariants for DC microgrids are obtained from Hynger in forms of bounds over the continuous state variables, and denoted as \( [B_l, B_u] \). 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., $\nu_{2,est}$, and $\nu_{2,est}$ 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 files, in Daikon compatible format, that are passed on to Daikon. Hence, the corresponding invariants are generated automatically, and shown in Table I. The SpaceX 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 $x_{2,pu}^{est}$, at time $t = 0.6 s$, through the compromised communication link. $x_{2,pu}^{est}$ 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
FDIA scenario is 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., $\varphi \notin \Phi$, 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.

2) Communication-based mitigation strategy: The communication link of the affected 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

**TABLE I**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Without FDIA</th>
<th>With FDIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1^{\text{out}}$</td>
<td>[47.874, 48.0818]</td>
<td>[47.9917, 48.0486]</td>
</tr>
<tr>
<td>$v_2^{\text{out}}$</td>
<td>[47.8739, 48.0818]</td>
<td>[47.9917, 48.5258]</td>
</tr>
<tr>
<td>$i_1^{\text{out}}$</td>
<td>[1.5071, 1.7187]</td>
<td>[1.418, 1.6016]</td>
</tr>
<tr>
<td>$i_2^{\text{out}}$</td>
<td>[1.5071, 1.7187]</td>
<td>[1.5997, 1.6175]</td>
</tr>
</tbody>
</table>

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.

**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.

**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.

**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.
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:

$$T_1 = \frac{1}{0.01s + 1}, \quad T_2 = \frac{1}{0.05s + 1}.$$  \hfill (25)

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

REFERENCES


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.