Virtual Prototyping for Distributed Control of a Fault-Tolerant Modular Multilevel Inverter for Photovoltaics

Luan Viet Nguyen, *Student Member, IEEE*, Hoang-Dung Tran, and Taylor T. Johnson, *Member, IEEE*

***Abstract*—In this paper, we present virtual prototyping of the distributed control for a modular multilevel inverter used as a grid-tie interface for photovoltaics. Due to the distributed control and inherent redundancy in the system composed of many panels and inverter modules, the system topology exhibits fault-tolerance capabilities that we study through virtual prototyping. The fault-tolerance is enable by several distributed algorithms, such as services to identify which if any agents controlling inverter modules have failed. A distributed identifier algorithm allows the system to keep track of the number of operating panels to appropriately regulate the DC voltage output of the panels using buck-boost converters, and determine appropriate switching times for H-bridges in the grid-tie. We evaluate the distributed inverter, its control strategy, and fault-tolerance through thousand of simulation scenarios in Mathworks Simulink/Stateflow. Our virtual prototyping framework allows for generating multilevel inverters composed of many inverter modules, and we evaluate inverters composed of five to dozens of inverter modules. Our analysis suggests the achievable total harmonic distortion (THD) of the modular multilevel inverter may allow for operating solar arrays in spite of failures of the power electronics, control software, and other subcomponents.**

***Index Terms*—distributed control, multilevel inverter, distributed inverter, hybrid systems.**

I. Introduction

Multilevel inverters have become popular in recent years for interfacing DC sources to the grid for a plethora of reasons, such as their ease of implementation, efficiency, fault-tolerance capabilities, etc. [1]–[8]. In this paper, we describe the model-based design and virtual prototyping analysis of a modular multilevel inverter implemented using fault-tolerant distributed control. The inverter is composed of N inverter modules that communicate and coordinate in a distributed manner to perform the DC to AC energy conversion. While the architecture allows for general DC sources, in this paper, we focus on inverter modules consisting of a solar panel composed of photovoltaic (PV) modules and corresponding electronics. In this case, each panel's control software and electronics implement maximum power-point tracking (MPPT) and regulate the panel output voltage using a buck-booster converter. Then, a $(2\mathsf{N}+1)$-level modular multilevel inverter is implemented using H-bridges to create a grid-tie connecting the regulated DC output of each buck-boost converter. An *inverter module* is the complete plant and computer controller consisting of a DC source (such as a panel), its microcontroller and network interface, buck-boost converter, and H-bridge. See Figure 1 for an overview of the modular multilevel inverter architecture.

The modules communicate with one another to ensure they switch at appropriate times to create the AC waveform for the grid that is in frequency, in-phase, and of the appropriate voltage. A distributed identifier algorithm is used by the N microcontrollers to determine (a) the number of non-faulty inverter modules, and (b) the switching time for each non-faulty inverter module to minimize total harmonic distortion (THD) for the AC grid-tie [9]. This setup makes the multilevel inverter modular, where it is not necessary to know the number of functioning modules $\mathsf{N_O} \leq \mathsf{N}$, a priori, as the distributed algorithm determines this. In addition, the distributed identifier algorithm lends the system to be fault-tolerant, whereby if some $\mathsf{N_F}$ of the N inverter modules and corresponding control software fails, the remaining panels and modules continue operating to ensure the grid-tie remains operational with reasonable THD and response time. Specifically, the switching times to connect each inverter module in series must be determined to minimize THD, which is equivalent to solving a certain distributed consensus problem in one dimension, and we base this distributed consensus algorithm on an existing fault-tolerant one-dimensional consensus algorithm [10].

We utilize an abstract failure model of both cyber and physical components, where software/hardware crash faults of any microcontroller are detected and tolerated, as are actuator stuck-at faults, which correspond to failed switches in the H-bridges. We characterize the THD of the system as a function of $\mathsf{N_F}$, the number of faulty modules, since as the number of faulty modules increases, the best response of the inverter will decrease, as the sinusoidal approximation has fewer discrete levels. In the optimal case, the best achievable THD of an inverter with N total modules and $\mathsf{N_F}$ faulty modules is that of an inverter with $\mathsf{N_O} = \mathsf{N} - \mathsf{N_F}$ functioning modules.

In this paper, we utilize the framework of *virtual prototyping* [11] to evaluate the general architecture, electrical component parameter variations, THD performance, and fault-tolerance capability. Instead of physically constructing one instance of the proposed architecture and evaluating it, we present simulation results for thousands of scenarios representing dozens of different system configurations, exploring a significantly larger design space than would be possible if

L.V. Nguyen and T.T. Johnson are with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019 USA, e-mail: (luanvnguyen@mavs.uta.edu, taylor.johnson@uta.edu. H.D. Tran is with the Department of Automatic Control, Ho Chi Minh City University of Transport, Ho Chi Minh City, Vietnam, e-mail: trhoang-dung@gmail.com.

Manuscript received March 17, 2014; revised September 1, 2014.

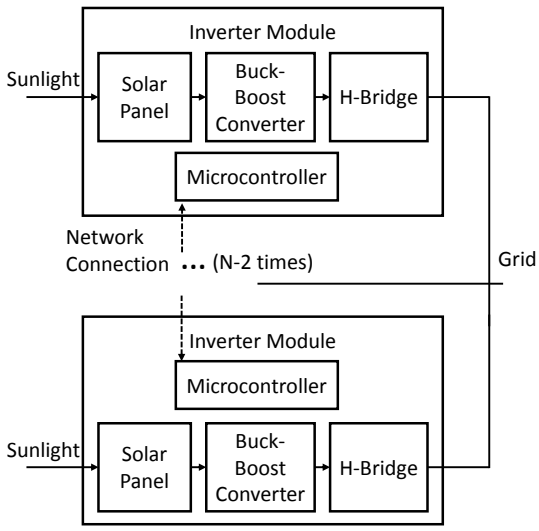


Fig. 1. Overview of the modular multilevel inverter for interfacing photovoltaics to the AC grid, consisting of N inverter modules, each of which is composed of a solar panel, a microcontroller with communication hardware and control software, a buck-boost converter, and an H-bridge for selecting polarity.

following an experimental process of building an individual prototype. The simulations are conducted in standard development environments (Mathworks Simulink/Stateflow, SLSF) and utilize high-fidelity models of the circuit components, the distributed control algorithms, and the interfaces between the cyber and physical states of the system. Overall, the virtual prototyping study presents a complete system-level picture of a sophisticated and fault-tolerant distributed cyber-physical system (DCPS) [12], and serves as a first step toward constructing instantiations of this system. Numerous modern systems, including other power electronics systems like the inverter analyzed in this paper, have been developed in this way recently, such as numerous components of a hybrid electric car [13], a power supply [14], and a tracking system for photovoltaic panels [15].

Related Work: There are abundant modulation techniques and control paradigms have been developed for multilevel converters such as sinusoidal pulse width modulation (SPWM), selective harmonic elimination (SHE-PWM), space vector modulation (SVM), and others. The two main advantages of multi-carrier PWM inverters in comparison to square-wave inverters are control over output voltage magnitude reduction in magnitudes of unwanted harmonic voltages. There are four main kinds of multi-carrier PWM: (a) in-phase disposition (IPD), (b) phase opposition disposition (POD), (c) alternative phase opposition disposition (APOD),and (d) phase disposition (PD). The PD method is used most frequently since it produces minimum harmonic distortion for the linetoline output voltage [16].

Fault tolerance contributes critically to eliminate downtime in industrial processes and enhance safety of critical systems. There is extensive literature [8], [17]–[27] regarding fault-tolerance capabilities of single and multi-phase multilevel inverters, as due to their topology, they have inherent redundancy that may be useful for providing fault-tolerance due to switch and other failures. However, compared to all these existing works, our work exploits the modular and distributed design of the system to enable fault-tolerance in both software and hardware subcomponents, as distributed control algorithms are typically implemented in software and rely on networked communications, all of which have the capability to fail. For a recent overview of general reliability and fault-tolerance in power electronics, see [24], for a particular focus on multilevel inverters, see [23], and for a focus on the reliability of DC-to-DC converters in PV energy conversion systems, see [28]. In [17] the reliability of multilevel inverters was studied to present an argument against reliability necessarily decreasing due to increased component counts, each with their own failure rates, which is an important observation for the topology considered in our paper, as we do not minimize component numbers, but instead strive for modularity and system-level fault-tolerance. A single-phase fault-tolerant multilevel inverter is developed and experimentally validated with 5-level prototype in [18] and focuses on utilizing redundant circuitry and appropriate control for maintaining the output voltage. Fault-tolerance in multilevel inverters can be achieved by adding some power device to the basic topologies such as fourth-leg [29] or reconfiguring the flying capacitor multilevel inverter into a full binary combination scheme, and balance capacitor voltage by using three-phase joint switching states [30]. A comparison of several inverter topologies along with their cost and reliability tradeoffs is presented in [19]. In [20], a strategy is developed for reconfiguring carrier-based modulation signals to provide fault-tolerance in multilevel inverters due to switches either failing open circuit or short circuit, and is experimentally evaluated on a three-phase five-level prototype. The authors of [21] develop a fault diagnosis system for multilevel inverters using neural networks.

Overall, the vast majority of fault-tolerance capabilities in multilevel inverters focus on handling hardware faults using redundant hardware and topology (i.e., physical) solutions. In contrast, in this paper, we consider primarily software-based fault-tolerance methods that exploit the inherent redundancy in the modular inverter topology. These distributed software-based techniques have the capability to handle both hardware (e.g., switch failures) and software faults (e.g., microcontroller crashes) using software (i.e., cyber) solutions. The topology of the inverter we consider in this paper is similar to that of [7], [31], but we utilize a buck-boost converter for DC voltage regulation, and we focus on distributed control instead of communication-less control. The topology of the converters and distributed control in [8] are also similar to our study, although we focus on photovoltaics instead of wind conversion. Additionally, one instance is experimentally analyzed in [8], while our virtual prototyping approach focuses on the distributed control and fault-tolerance to examines thousands of different scenarios through thousands of virtual experiments. We do not focus on any particular maximum power point tracking (MPPT) scheme in this paper, but refer readers to numerous methods and their tradeoffs in [32].

Our virtual prototyping simulator is developed in Mathworks Simulink/Stateflow, and allows for simulating inverters composed of an arbitrary number of modules (N) and failures (N_F). Numerous power electronics simulation models have been developed previously using Simulink/Stateflow [33]–

[36], and our simulator utilizes high-fidelity models of both the cyber and physical components of each inverter module. A MATLAB simulation model for PV modules is presented in [33] and considers factors such as temperature, shading, etc. In [34], the authors develop a MATLAB/Simulink model of a grid-connected single-phase array with MPPT, but do not consider multilevel inverters as we do. In [35], the authors develop a MATLAB/Simulink model of PV modules accounting for numerous non-idealities, such as nonuniform irradiance. Requirements for DC-link voltages are detailed in [37]. A detailed MATLAB/Simulink for studying partial shading is studied in [36]. Overall, numerous practical issues of utilizing photovoltaics in micro-inverters, cascaded inverters, and multilevel inverters—such as nonuniform irradiance, common mode voltage issues, floating panels, large leakage currents, efficiency comparisons, etc.—have been addressed through modeling and experimentation elsewhere [33]–[36], [38], [39], but the contributions of our work are in the distributed control and virtual prototyping.

Contributions: The main contributions of this virtual prototyping study are: (a) the development and implementation of the fault-tolerant distributed control strategy for a modular multilevel inverter for DC-to-AC conversion, (b) the holistic design and analysis of a distributed cyber-physical system (CPS) in a standard virtual prototyping environment (Mathworks Simulink/Stateflow), and (c) the application of distributed and hybrid systems modeling techniques in the virtual prototyping process. We highlight that in contrast to most existing work on fault-tolerance of multilevel inverters, the failure model considered in this paper is an abstraction of both cyber and physical failures, and works by coordination through distributed control.

Paper Organization: The remainder of this paper is organized as follows. Section II presents the modular multilevel inverter and its distributed control, including the communication and computation capabilities of its subcomponents, as well as the cyber-physical failure model of the subcomponents. Section III presents the simulation-based analysis of the virtual prototype, including comparisons of THD with and without failures, different failure modes, and arrays consisting of $\mathsf{N} = 5$ to $\mathsf{N} = 35$ panels and inverter modules. Section IV concludes the paper and presents directions for future work.

II. Modular Multilevel Inverter Architecture

Preliminaries: For a set S, let $|S|$ be the cardinality of S, which is the number of elements in S. For a set S, let $S_\perp$ be $S \cup \{\perp\}$ where $\perp \notin S$. We model several of the cyber-physical components of the system using the hybrid automaton formalism. We begin by briefly reviewing hybrid automata, and refer interested readers to [40]–[43] for detailed definitions of such modeling formalisms, and to [44]–[47] for descriptions specified to power electronics and systems. A *hybrid automaton* is a (possibly nondeterministic) state machine with state that can evolve both instantaneously (through *discrete transitions*) and over intervals of time (according to *trajectories*). Variables are associated with types and are used as names for state components, such as currents, voltages, and times. For a set of variables V, a valuation $\mathbf{v}$ is a function that maps each variable $v \in V$ to a point in its type, denoted $type(v)$. The set of all possible valuations is $val(V)$. For a valuation $\mathbf{x}$, we use $\mathbf{x}.x$ to denote the value of the variable $x \in V$. Since the distributed system is composed of N inverter modules, each of which has its own power electronics, software, etc., we model the i^{th} inverter module as an automaton $\mathcal{A}_i$.

Mathematically, a hybrid automaton $\mathcal{A}_i$ is a tuple $\langle Var_i, \mathsf{Loc}_i, Q_i, \Theta_i, Edg_i, Grd_i, Rst_i, Flow_i, Inv_i \rangle$, where: (a) Var_i: is a set of variables, where $X_i \subseteq Var_i$ are the continuous, real-typed variables. (b) Loc_i: is a set of discrete locations. (c) $Q_i \triangleq val(Var_i)$ is the set of states, and is the set of all valuations of each variable $v \in Var_i$. A *state* is denoted by bold $\mathbf{x}$ and assigns values to every variable in the set of variables Var_i. For a state $\mathbf{x} \in Q_i$, the valuation of $\mathbf{x}.loc$ is called the *location*, and along with the valuations of any discrete variables, it describes the discrete state. The valuation of the continuous variables in X_i, that is $\{\mathbf{x}.x : x \in X_i\}$, is called the *continuous state* and is referred to as $\mathbf{x}.X_i$. (d) $\Theta_i \subseteq Q_i$ is a set of *initial states*. (e) Edg_i is the set of *edges*. (f) $Grd_i : Edg_i \to Q_i$ is a function that associates a *guard* (a valuation of V that must be satisfied such that a transition may be taken) with each edge. (g) $Rst_i : Edg_i \to (Q_i \to 2^{Q_i})$ is a function, called the *reset map*, associated with each edge. A reset map associates a set of states with each edge. (h) $Flow_i : \mathsf{Loc}_i \to (Q_i \to 2^{Q_i})$ associates a *flow map* with each location. (i) $Inv_i : \mathsf{Loc}_i \to 2^{Q_i}$ associates an *invariant* with each location.

The semantics of $\mathcal{A}_i$ are defined in terms of sets of *transitions* and *trajectories*. The set of transitions $\mathcal{D}_i \subseteq Q_i \times Q_i$ is defined as follows. We have $(\mathbf{v}, \mathbf{v}') \in \mathcal{D}_i$ if and only if, for $e = (\mathbf{v}.loc, \mathbf{v}'.loc)$, (a) $e \in Edg_i$, (b) $\mathbf{v} \in Grd_i(e)$, and (c) $\mathbf{v}' \in Rst_i(e)(\mathbf{v}.X)$. A *trajectory* for $\mathcal{A}_i$ is a function $\tau : [0, t] \to Q_i$ that maps an interval of time to states such that: (a) For all $t' \in [0, t]$, $\tau(t').loc = \tau(0).loc$, that is, the discrete state remains constant, (b) $(\tau \downarrow X)$, that is, the restriction of τ to X_i is a solution of the differential equation specified by the flow function $\dot{X}_i = Flow_i(\tau(0).loc)(\tau(0))$, and (c) For all $t' \in [0, t]$, $\tau(t') \in Inv_i(\tau(0).loc)$. The set of all the trajectories of $\mathcal{A}_i$ is written $\mathcal{T}_i$. The domain for a trajectory $\tau \in \mathcal{T}_i$ is denoted by $\tau.\mathsf{dom}$. We define $\tau.\mathsf{ltime}$ as the right endpoint of $\tau.\mathsf{dom}$, $\tau.\mathsf{lstate} \triangleq \tau(\tau.\mathsf{ltime})$, and $\tau.\mathsf{fstate} \triangleq \tau(0)$. An *execution* of $\mathcal{A}_i$ is a sequence $\alpha = \tau_0\tau_1 \ldots$, such that: (a) each $\tau_k \in \mathcal{T}_i$, (b) for each k, $(\tau_k(t), \tau_{k+1}(0)) \in \mathcal{D}_i$, where t is the right endpoint of the domain of τ_k, and (c) $\tau_0 \in \Theta_i$. A state $\mathbf{v} \in Q_i$ is said to be *reachable* if there exists a finite execution α that ends with $\mathbf{v}$.

A. Modular Multilevel Inverter Architecture and Modeling

The modular multilevel inverter consists of N inverter modules for implementing the modular multilevel inverter as a grid-tie for photovoltaics (see Figure 1). Each inverter module consists of a microcontroller (computer), communications system, a DC source (a solar panel in this paper), and power electronics. Each inverter module's power electronics

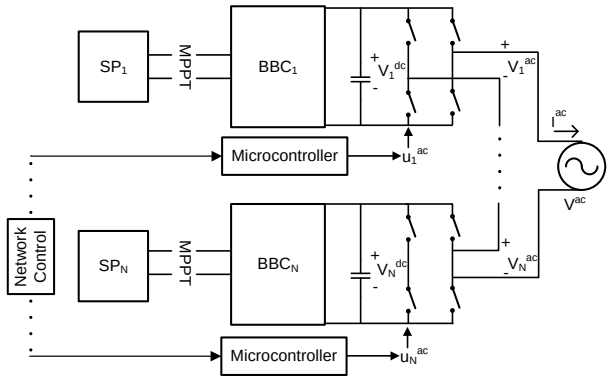


Fig. 2. High-level circuit diagram of the modular multilevel inverter illustrating the solar panels (SP_i) controlled with MPPT that feed the panel output voltage V_i^{sp} into the buck-boost converters (BBC_i) with output voltage V_i^{dc}. Next, the H-bridges switch at appropriate times to connect the N DC regulated voltage sources V_i^{dc} with potentially reversed polarity in series to create the grid connection voltage V^{ac}. The buck-boost converter control (V_i^{ref}) and H-bridge switching control (u_i^{ac}) for inverter module i depends upon network information from other inverter modules in the system.

consist of a DC-to-DC buck-boost converter for regulating the DC source's potentially varying output voltage, and an H-bridge for connecting and disconnecting the buck-boost converter's output voltage at appropriate times to generate the AC waveform (see Figure 2). We note that inclusion of a software-controlled bypass switch in parallel on the output of the H-bridge would allow the inverter to tolerate both open and circuit-circuit failures, although such a method has previously been investigated, so we instead focus on both cyber and physical failures [8]. We refer to each inverter module as an *agent* with a unique identifier $i \in ID$, where $ID \triangleq \{1, \ldots, \mathsf{N}\}$. We model the i^{th} inverter module's buck-boost converter as a hybrid automaton (see Figure 4) denoted $\mathcal{A}_i^{dc}$, and its H-bridge as a hybrid automaton (see Figure 6) denoted $\mathcal{A}_i^{ac}$. Each inverter module is specified as a hybrid automaton consisting of the composition of the individual components:

$$\mathcal{A}_i \triangleq \mathcal{A}_i^{dc} \parallel \mathcal{A}_i^{ac}. \tag{1}$$

For a given N, the complete system $\mathcal{A}$ composed of the N inverter modules consists of N panels, N buck-boost converters, N H-bridges, and computer control software and hardware is:

$$\mathcal{A} \triangleq \mathcal{A}_1 \parallel \ldots \parallel \mathcal{A}_{\mathsf{N}}, \tag{2}$$

where $\parallel$ is a parallel (concurrent) composition of automata (see, e.g., [43, Chapter 2]).

Each agent $i \in ID$ is associated with the following electrical (physical) real variables: (a) V_i^{sp}: the voltage output of agent i's solar panel and input to agent i's DC-to-DC converter, (b) I_i^{sp}: the output current of agent i's solar panel and input to agent i's DC-to-DC converter, (c) V_i^{ref}: the reference voltage for agent i's DC-to-DC converter to track, (d) V_i^{dc}: the voltage output of agent i's DC-to-DC converter and input to agent i's H-bridge, (e) I_i^{dc}: the current output of agent i's DC-to-DC converter and input to agent i's H-bridge, (f) V_i^{ac}: the voltage output of agent i's H-bridge and input to the grid, and (g) I_i^{ac}: the current output of agent i's H-bridge and input to the grid. Additionally, each agent $i \in ID$ is associated with the following communications and computational (cyber) quantities: (a) $\Delta_i^{ac} \triangleq \{\delta_i^{z+}, \delta_i^{p}, \delta_i^{z-}, \delta_i^{z-}\}$: a set of switching times for agent i's H-bridge to connect/disconnect V_i^{ac} with what polarity to the grid, (b) u_i^{ac}: the H-bridge control timer for agent i used to compare to the switching times in Δ_i^{ac}, (c) $Nbrs_i$: the communication neighbors of agent i, consisting of the agents to its left (denoted L_i) and right (denoted R_i). The left and right neighbors are defined to be the adjacent inverter modules, e.g., in Figure 1. Without failures, we have $L_i = i - 1$ and $R_i = i + 1$, for $i \geq 2$ and $i \leq \mathsf{N} - 1$, respectively, but will redefine these in the case of failures shortly. These variables define the set of variables Var_i of the automata $\mathcal{A}_i^{dc}$ and $\mathcal{A}_i^{ac}$. As we consider their compositions, we do not differentiate between variables of the two automata. Additionally, we note that all these variables are mappings from time to elements in the variables' types. For some $v \in Var_i$, we will denote this interchangeably by $\mathbf{x}.v$ for some reachable state $\mathbf{x}$, or by $v(t)$ for some time $t \in \mathbb{R}_{\geq 0}$ such that $t = \tau.\mathsf{ltime}$ and $\mathbf{x} = \tau.\mathsf{lstate}$, i.e., t is the endpoint of a trajectory τ ending in reachable state $\mathbf{x}$.

B. Failure Model and Distributed Failure Detection

We utilize the following failure model of each agent's physical and cyber components, inspired by similar models developed in [10], [48]. While H-bridge failure modes could potentially turn them into open circuits, thus disconnecting the inverter from the grid, we do not consider such scenarios and assume if the H-bridge fails, it fails as a short adding zero voltage to V^{ac}. We model general abstracted failures of the entire inverter module that do not cause open circuits, such as the microcontroller crashing, the buck-boost converter entering a failure mode, etc. We assume we have a method to detect failures within a finite time, e.g., through a heartbeat service for crash failures. This assumption is reasonable for both cyber and physical failures. For cyber failures—e.g., computer crashes and may recover, communication link is lost temporarily—such heartbeat services are typical in distributed computing systems, but here we desire the grid-tie to recover when the computer restarts or the communication link is restored. For physical failures, each inverter module may implement a detection method to detect if its subcomponents are faulty. Thus, this failure model is an abstraction of more detailed failures. Each agent $i \in ID$ is augmented with an additional Boolean-valued variable F_i indicating whether it has failed ($true$) or not ($false$). If agent $i \in ID$ is failed, then $F_i(t) = true$, and if not, $F_i(t) = false$. The set of failed agents is denoted by $ID_{\mathsf{F}}(t) \subseteq ID$ and is the set $\{i \in ID \mid F_i(t)\}$. We define the number of failed agents as $\mathsf{N}_{\mathsf{F}}(t) \triangleq |ID_{\mathsf{F}}(t)|$. The set of operating (non-failed) agents is denoted by $ID_{\mathsf{O}}(t) \subseteq ID$ and is the set $ID \setminus ID_{\mathsf{F}}(t)$. We also define the number of operating agents as $\mathsf{N}_{\mathsf{O}}(t) \triangleq |ID_{\mathsf{O}}(t)|$ and we note $\mathsf{N}_{\mathsf{O}}(t) = \mathsf{N} - \mathsf{N}_{\mathsf{F}}(t)$.

A distributed gossip protocol [49] spreads the identifiers of any failed agents throughout the inverter, so any agent knows within a short period of time if any other agent is failed or not. Using this information, the left and right neighbors are redefined, respectively, as $L_i(t) = \max\{j \in ID | F_j(t) \wedge j < i\}$ and $R_i(t) = \min\{j \in ID | F_j(t) \wedge j > i\}$.

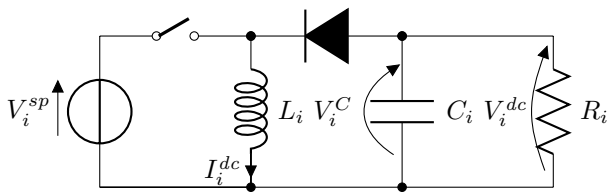


Fig. 3. Buck-boost converter circuit—a DC input V_i^{sp} is increased or decreased to a higher or lower DC output V_i^{dc} based on the number of non-faulty inverter modules.

Distributed Identification and Notification: Each agent $i \in ID$ is augmented with a variable id_i with index type $(type(id_i) = ID_\perp)$, which indicates its identifier in the set of operational agents, ID_{O}. First, each agent keeps track of the number of failures to its left (lower identifiers) as $L_i^F(t) = |\{j \in ID \mid F_j(t) = true \wedge j < i\}|$, and symmetrically $R_i^F(t)$ for agents to its right (higher identifiers). We observe that $\mathsf{N_F}(t) = L_i^F(t) + R_i^F(t)$, so agents may compute the number of failed agents. Each operational agent $i \in ID_{\mathsf{O}}$ determines id_i using the following local method:

$$id_i(t) = i - L_i^F(t). \tag{3}$$

Using this method, we have that $\max_{i \in ID} id_i(t) = \mathsf{N_O}(t)$. Together, these distributed identifier services allow each operational agent $i \in ID_{\mathsf{O}}$ to compute the number of operational and failed agents for use in determining the DC voltage reference V_i^{ref} and switching times Δ_i^{ac} as described next.

C. Buck-Boost Converter Model and Control

The buck-boost converter circuit appears in Figure 3. For the buck-boost converter model, we utilize an extension of the hybrid automaton model developed and analyzed in [46]. Each inverter module's buck-boost converter has two real-valued state variables modeling physical quantities: the inductor current I_i^{dc} and the capacitor voltage V_i^{dc}, depicted in Figure 3. These two state variables at time t are written in vector form as:

$$x_i(t) = \begin{bmatrix} I_i^{dc}(t) \\ V_i^{dc}(t) \end{bmatrix}.$$

The model includes the discontinuous conduction mode (DCM), see e.g., [50], [51].

The reference voltage for each DC-to-DC converter is:

$$V_i^{ref}(t) \triangleq \frac{V^p}{\mathsf{N_O}(t)}, \tag{4}$$

where V^p is the AC peak voltage (e.g., $V^p = \sqrt{2}V^{rms}$ for the root mean square (RMS) AC voltage V^{rms}). If $V_i^{ref}(t) < V_i^{sp}(t)$, then the buck-boost converter is in a buck mode and decreases its output voltage $V_i^{dc}(t)$. Otherwise, if $V_i^{ref}(t) > V_i^{sp}(t)$, then the buck-boost converter is in a boost mode and increases its output voltage $V_i^{dc}(t)$. Note that since $V_i^{ref}(t)$ is defined in terms of the number of operating agents $\mathsf{N_O}(t)$, it may vary over time.

D. H-Bridge Modeling and Distributed Control

We model the H-bridge plant as ideal switches with delays, where the output voltage of the inverter module is either:

Switch S_i State	A_i^m	B_i^m	Duty Cycle $\delta_i^{dc}(t)$
Open	$\begin{bmatrix} 0 & -\frac{1}{L_i} \\ \frac{1}{C_i} & -\frac{1}{R_i C_i} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\frac{V_i^{ref}(t)}{V_i^{ref}(t)+V_i^{sp}(t)}$
Close	$\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_i C_i} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{L_i} \\ 0 \end{bmatrix}$	
DCM	$\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{R_i C_i} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	

TABLE I
DYNAMICS OF AGENT i'S BUCK-BOOST CONVERTER $\mathcal{A}_i^{dc}$.

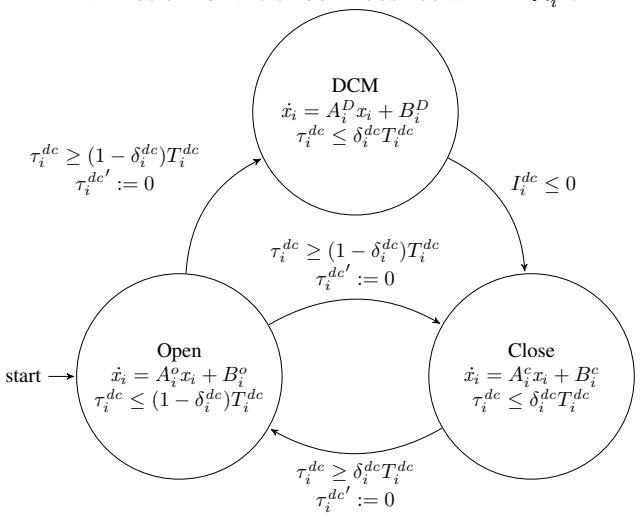


Fig. 4. Hybrid automaton model $\mathcal{A}_i^{dc}$ for agent i's buck-boost converter. The matrices and vectors A_i^o, A_i^c, A_i^D, B_i^o, B_i^c, and B_i^D are constant but may vary between inverter modules, and T_i^{dc} is constant. The state vector x_i, duty cycle δ_i^{dc}, and τ_i^{dc} are variables and vary with time.

(a) $V_i^{ac} = 0$: disconnected (locations $Zero^+$ and $Zero^-$), (b) $V_i^{ac} = V_i^{dc}$: connected in series with positive polarity (location $Positive$), or (c) $V_i^{ac} = -V_i^{dc}$: connected in series with reverse polarity (location $Negative$). The H-bridge controller that connects the output voltage is shown in Figure 6. The grid-tie AC voltage V^{ac} is then defined as the series connection of all $\mathsf{N_O}$ operating inverter modules output voltages:

$$V^{ac}(t) \triangleq \sum_{i \in ID_{\mathsf{O}}(t)} V_i^{ac}(t). \tag{5}$$

The set of switching times for the H-bridge to connect V_i^{dc} with different polarities to create V_i^{ac} is denoted:

$$\Delta_i^{ac}(t) \triangleq \{\delta_i^{z+}(t), \delta_i^{p}(t), \delta_i^{z-}(t), \delta_i^{n}(t)\}, \tag{6}$$

where the elements are respectively the time to spend with $V_i^{ac} = 0$, then the time to spend with $V_i^{ac} = V_i^{dc}$, then the time to spend with $V_i^{ac} = 0$ again, and finally the time to spend with $V_i^{ac} = -V_i^{dc}$ before repeating. See Figure 11 for an example of the switching signals illustrating these various transitions. For finding the switching times of the H-bridge, we utilize the following protocol and we derive the idealized switching times for each agent $i \in ID$:

$$\frac{i}{\mathsf{N_O}+1} = \sin\left(\frac{2\pi t}{T^{ac}}\right), \text{ and solving for } t,$$

$$t = \frac{T^{ac}}{2\pi} \sin^{-1}\left(\frac{i}{N+1}\right).$$

Of course, t is not unique, but defines the amount of time δ_i^{z+}

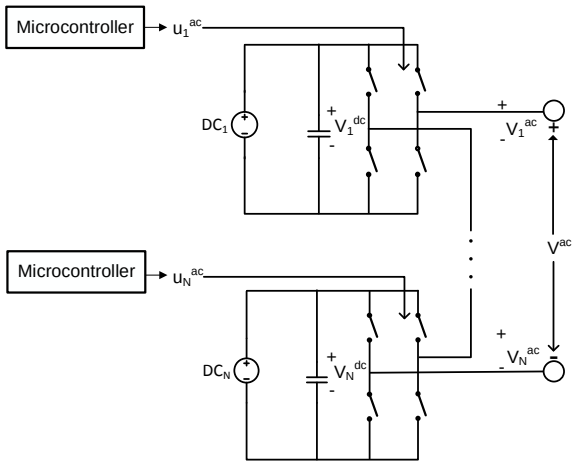


Fig. 5. For the purpose of the H-bridge control and finding the switching signals u_1^{ac}, ..., u_{N}^{ac}, the panel and buck-boost converter are abstracted and treated as ideal voltage sources (DC_1, ..., DC_{N}) due to the buck-boost converter regulation.

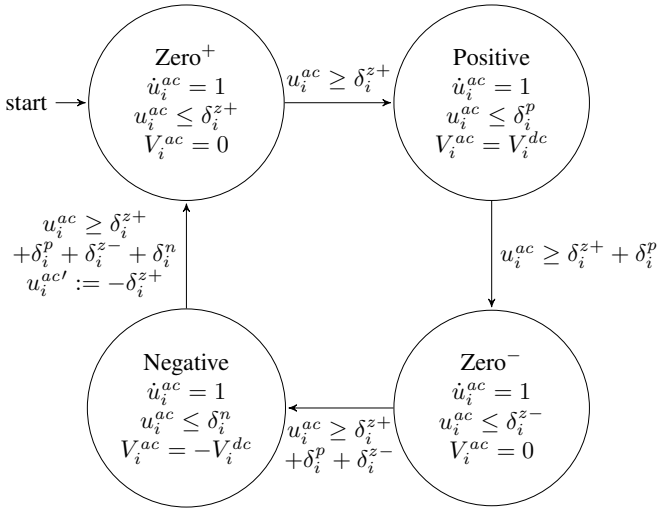


Fig. 6. Hybrid automaton model $\mathcal{A}_i^{ac}$ for agent i's H-bridge switching logic.

spent in the zero state before switching to the positive output state. The other waiting times simply subdivide the period, and accounting for failures using i's identifier id_i out of the $\mathsf{N_O}$ operating agents, we have:

$$\delta_i^{z+}(t) \;=\; \frac{T^{ac}}{2\pi} \sin^{-1}\left(\frac{id_i(t)}{\mathsf{N_O}(t)+1}\right), \qquad (7)$$

and likewise for the shifted switching times δ_i^{p}, δ_i^{n}, and δ_i^{z-}. We assume that the sinusoid used to generate the switching times in Equation 7 is synchronized with the grid phase, using, e.g., a phase-locked loop (PLL), which can be implemented in a distributed fashion by informing all operational agents of the grid phase. Refer to Figure 11 for examples of the switching times generated using this method with failures.

III. Virtual Prototype Simulation Analysis

Next we describe the virtual prototyping simulation setup to analyze the distributed control and fault-tolerance capabilities of the modular multilevel inverter. We wrote a MATLAB program to programmatically generate Simulink/Stateflow (SLSF) models of the inverter for varying the number of inverter modules (N). Specifically, for a given N, the program generates

Component / Parameter Name	Symbol	Value
Buck-Boost Input Voltage	$V_i^{sp}(t)$	18.6 V $\pm\ \epsilon$
Desired Buck-Boost Output Voltage	$V_i^{ref}(t)$	$\frac{V^{rms}}{\mathsf{N_O}(t)}$ V
Actual Buck-Boost Output Voltage	$V_i^{dc}(t)$	varies
Load Resistance	R_i	4 Ω $\pm$ 5%
Capacitor	C_i	60 uF $\pm$ 5%
Inductor	L_i	40 uH $\pm$ 5%
Switching Period	T_i^{dc}	4 μs
Switch-closed duty cycle	$\delta_i^{dc}(t)$	varies
Switch-open duty cycle	$1-\delta_i^{dc}(t)$	varies
Grid Period	T^{ac}	0.0167 s
Grid Frequency	f^{ac}	60 Hz
Desired Grid Voltage	V^{grid}	120 V^{rms}, 60 Hz
Actual Array Voltage	$V^{ac}(t)$	varies

TABLE II
Summary of variables and parameters used in simulations.

an array $\mathcal{A}$ consisting of the N solar panels, N inverter modules, and their control software composed together, i.e.,, as specified in Equation 2. That is, the simulator generates SLSF simulation models corresponding to Figures 1 and 2. The various parameters used for the circuit components are summarized in Table II. The grid-tie was configured for a standard residential-style connection at 120 V and 60 Hz. The control logic for both automata $\mathcal{A}_i^{dc}$ and $\mathcal{A}_i^{ac}$ are implemented as continuous-time state-machines using Stateflow. Using these programmatically-generated array models, we have performed thousands of simulations for analyzing the system in scenarios with and without failures, as detailed next.

A. Total Harmonic Distortion (THD) with Static Failures

Static failures are those that occur before the grid-tie is connected and do not affect the dynamic performance. Figure 7 shows an example execution for $\mathsf{N} = 35$ inverter modules with both no failures and $\mathsf{N_F} = 5$ static failures, along with an execution for $\mathsf{N} = 10$ inverter modules with no failures. Figure 8 shows the THD of the array as a function of the number of operating agents, $\mathsf{N_O}$. Additionally, Figure 8 shows the THD for *static* failures, which are those where some agents are failed at start-up and remain failed. The results illustrate that increasing the number of static failures returns the array to the achievable THD in an array with $\mathsf{N_F}$ fewer inverter modules. The different curves in Figure 8 correspond to the numbers of non-failed agents $\mathsf{N_O}$ for a given array of N inverter modules. The simulations varied $\mathsf{N_F}$ from 0 (no failures) to 6 (six failed agents), and N from 10 agents through 35 agents, corresponding to 21 and 71 levels, respectively. For example, in the $\mathsf{N} = 10$ configuration with no failures ($\mathsf{N_F} = 0$), the THD of the array is around 5%. In the $\mathsf{N} = 15$ configuration with $\mathsf{N_F} = 5$ failures, the THD is also around 5%. These configurations may result in too high a THD for the grid-tie, but the THD is around 2.5% for $\mathsf{N_O} \geq 16$, so as long as there are at least a large fraction of functioning panels and inverter modules in large arrays, the grid-tie could be connected.

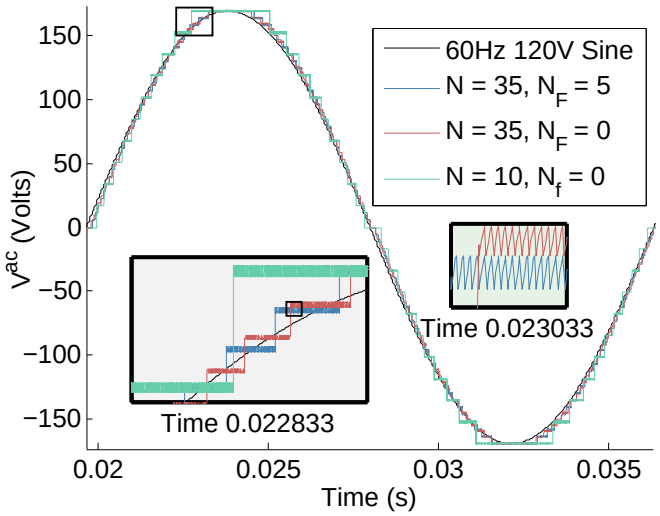


Fig. 7. Executions of three configurations of the array, with $\mathsf{N} = 35$ agents and $\mathsf{N_F} = 5$ failures, with $\mathsf{N} = 35$ agents and no failures, and $\mathsf{N} = 10$ agents and no failures. The figures illustrate the different H-bridge switching times and buck-boost regulated voltage levels in different configurations.

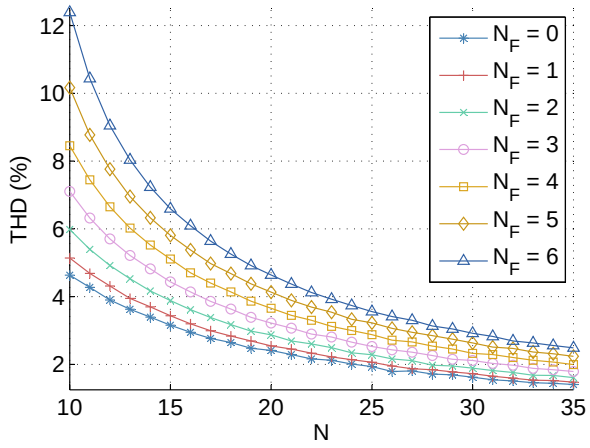


Fig. 8. THD for different array configurations consisting of N panels and inverter modules (agents), along with different numbers of statically failed agents $\mathsf{N_F}$ at system start-up. The y-axis scale is logarithmic.

B. THD with Dynamic Failures

Dynamic failures are those that occur once the grid-tie is operational and connected. We consider dynamic failures ($\mathsf{N_F} = 1$) of one agent at a time. Figures 9 and 10 each, respectively, show the grid-tie voltage V^{ac} versus time for three executions with one random dynamic failure that occurs at a uniformly distributed random time in the period. These scenarios are considered as failures at different times result in varying performance degradation of the THD. For instance, one scenario is where a failure of an agent that is not connected to V^{ac} at a time instant. One hypothesis is that such a failure may not negatively impact the THD, as it is not connected to the output. However, each of the remaining operational agents $i \in ID_{\mathsf{O}}$ must (a) increase their output voltages V_i^{dc} since there is one fewer level, and (b) change their H-bridge switching times Δ_i^{ac} using the algorithm of Equation 7. Figure 11 shows the H-bridge output voltage V_i^{ac} for each agent $i \in ID$ for a configuration with $\mathsf{N} = 6$ agents and one dynamic failure.

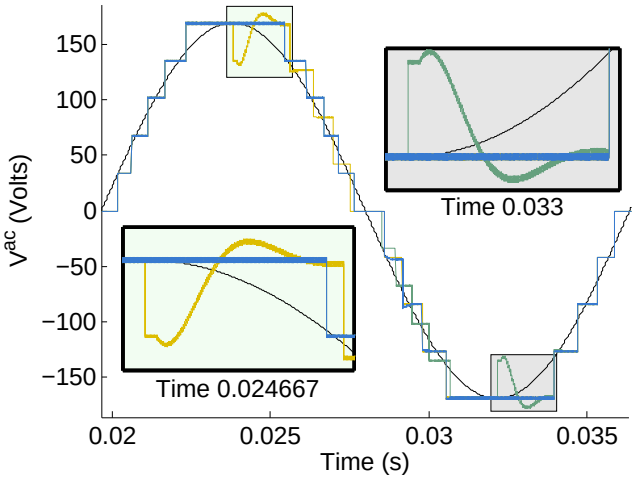


Fig. 9. The black line is an ideal 60 Hz 120 V sine, and the green, yellow, and blue lines are each an execution of $\mathcal{A}$ with $\mathsf{N} = 5$ agents and 1 dynamic failure at a different random time. The failure causes the total number of voltage levels to transition from $(2\mathsf{N}+1) = 11$ to $(2\mathsf{N_O}+1) = 9$ levels. The zoom plots illustrate the fast recovery as the buck-boost converter reference voltage control and the H-bridges' switching times are changed. Note that in each of the three executions, in the first quarter-period ($t \leq 0.022$) there are $\mathsf{N} = 5$ positive voltage levels as there are 5 functioning agents, and the recovery is fast enough that by fourth quarter-period ($t \geq 0.0325$) there are $\mathsf{N} = 4$ negative voltage levels due to the one dynamic failure ($\mathsf{N_F} = 1$).

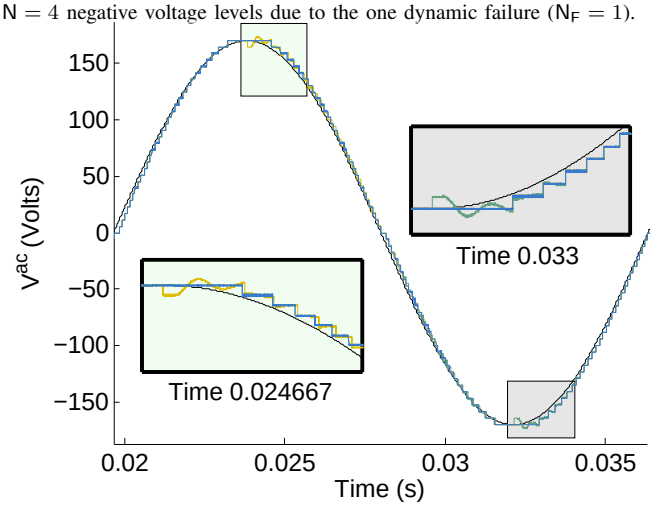


Fig. 10. The black line is an ideal 60 Hz 120 V sine, and the green, yellow, and blue lines are each an execution of $\mathcal{A}$ with $\mathsf{N} = 30$ agents and 1 dynamic failure at a different random time. The failure causes the total number of voltage levels to transition from $(2\mathsf{N}+1) = 61$ to $(2\mathsf{N_O}+1) = 59$ levels. The zoom plots illustrate the fast recovery as the buck-boost converter reference voltage control and the H-bridges' switching times are changed. Note that in each of the three executions, in the first quarter-period ($t \leq 0.022$) there are $\mathsf{N} = 30$ positive voltage levels as there are 30 functioning agents, and the recovery is fast enough that by fourth quarter-period ($t \geq 0.0325$) there are $\mathsf{N} = 29$ negative voltage levels due to the one dynamic failure ($\mathsf{N_F} = 1$).

Figure 12 shows averaged THD versus time over two periods ($2T^{ac}$) for arrays composed of $\mathsf{N} = 5$ to 35 agents in increments of 5 agents where a single dynamic failure ($\mathsf{N_F} = 1$) occurs in the first of the two periods. These results correspond to the scenarios depicted in Figures 9 and 10, with the averaged THD in Figure 13. Figure 12 indicates that in the case of a single failure, the THD of the N agent system returns to that of the $\mathsf{N} - 1$ agent system quickly (within one period T^{ac}). It is unlikely more than a single dynamic failure would occur simultaneously before recovery, which as shown in Figures 9, 10, 11, and 12, happens in under half

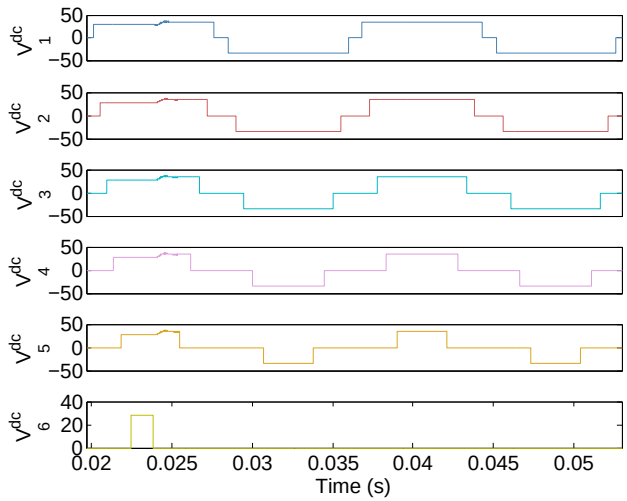


Fig. 11. Execution of $\mathsf{N} = 6$ agents with one dynamic failure ($\mathsf{N_F} = 1$) illustrating the H-bridge switching signals and output voltages V_i^{dc} for each agent $i \in ID$. The failure causes agent $i = 6$ to have $F_i = true$ and $V_i^{dc} = 0$, so the operating agents $i \in ID_{\mathsf{O}}$ increase their reference voltages V_i^{ref} and update their switching times Δ_i^{ac}.

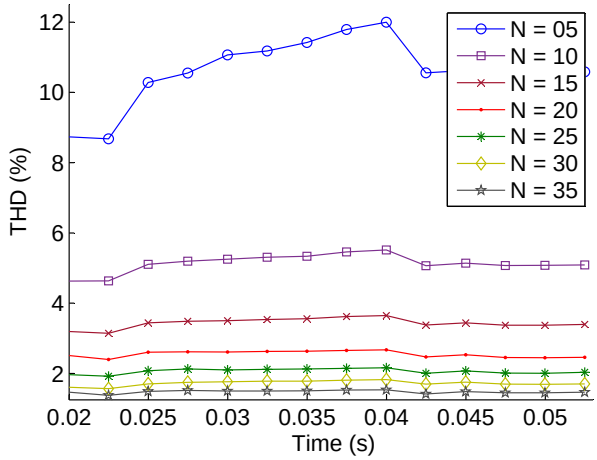


Fig. 12. Averaged THD versus time for the 20 executions with uniformly sampled random failure time for $\mathcal{A}$ with $\mathsf{N} = 5$ through $\mathsf{N} = 35$ agents and 1 dynamic failure at different random times. The time-varying lines are the THD at different instances due to the dynamic failure. The y-axis scale is logarithmic.

a grid period T^{ac}. Furthermore, if one failure occurs, from our previous analysis of THD under static failures (Figure 8), we see that the array behavior simply returns to the system's behavior with $\mathsf{N} - 1$ operating agents. Thus, if more than a single dynamic failure occurs ($\mathsf{N_F} > 1$, as long as each failure is spaced out enough in time (greater than a half grid period apart), the overall behavior will just return the array to the behavior with $\mathsf{N} - 1$, then $\mathsf{N} - 2$, ..., $\mathsf{N} - \mathsf{N_F}$ agents operating.

IV. Conclusion and Future Work

In this paper, we presented a model-based design and virtual prototyping analysis of a modular multilevel inverter used to interface N DC voltage sources—in this paper, solar panels—to the grid. In addition to the modular multilevel inverter considered in this paper, the virtual prototyping, failure modeling, and analysis may be useful in scenarios using multilevel inverters as grid-ties for other DC sources and in other topologies. Our results illustrate the feasibility of individual and multiple

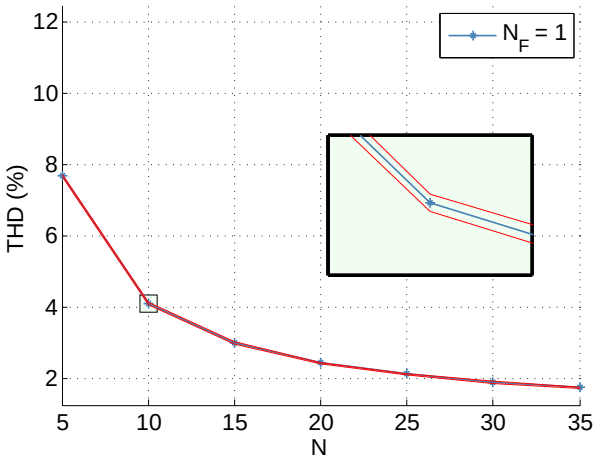


Fig. 13. Averaged THD of 20 simulations for different array configurations consisting of N panels and inverter modules (agents) in increments of 5 between 5 and 35, along with a one dynamically failed agent, $\mathsf{N_F} = 1$, that fails at a uniformly-sampled random time in the grid period T^{ac}. The y-axis scale is logarithmic, the center line is the mean, and the upper and lower lines are $\alpha = 95$ confidence intervals.

inverter modules failing in certain ways, while being able to keep the grid-tie operational with acceptable performance deterioration (in terms of THD). In future work, we plan to construct an actual prototype of the modular multilevel inverter and evaluate its fault-tolerance capabilities in real-world scenarios, although the detailed virtual prototyping analysis presented in this paper indicates such a system will be feasible, and has extensively explored the design space of possible system instantiations to be investigated in experimental work. For an actual prototype, we plan to employ a switching scheme to vary the switching times used by each agent's H-bridge to decrease wear by periodically changing identifiers of all the agents using a distributed identifier algorithm. Additionally, given the formal hybrid automaton models of the system, we plan to formally verify several specifications of the distributed control algorithms regardless of the number of inverter modules, N, using the *Passel* verification tool [43]. For example, one basic specification is that the switching logic of the modules never results in modules with opposite polarity voltages being connected together for the grid tie. Alternatively, this can be formulated as a verification problem for timed automata as done previously for an array with fixed size (N) in [44].

References

[1] J.-S. Lai and F. Z. Peng, "Multilevel converters-a new breed of power converters," *Industry Applications, IEEE Transactions on*, vol. 32, no. 3, pp. 509–517, May 1996.

[2] L. Tolbert, F. Z. Peng, and T. Habetler, "Multilevel converters for large electric drives," *Industry Applications, IEEE Transactions on*, vol. 35, no. 1, pp. 36–44, Jan. 1999.

[3] B. McGrath and D. Holmes, "Multicarrier PWM strategies for multilevel inverters," *Industrial Electronics, IEEE Transactions on*, vol. 49, no. 4, pp. 858–867, Aug. 2002.

[4] S. Kjaer, J. Pedersen, and F. Blaabjerg, "A review of single-phase grid-connected inverters for photovoltaic modules," *Industry Applications, IEEE Transactions on*, vol. 41, no. 5, pp. 1292–1306, 2005.

[5] L. Franquelo, J. Rodriguez, J. Leon, S. Kouro, R. Portillo, and M. Prats, "The age of multilevel converters arrives," *Industrial Electronics Magazine, IEEE*, vol. 2, no. 2, pp. 28–39, June 2008.

[6] C. Cecati, F. Ciancetta, and P. Siano, "A multilevel inverter for photovoltaic systems with fuzzy logic control," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 12, pp. 4115–4125, 2010.

[7] B. Johnson, "Control, analysis, and design of distributed inverter systems," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2013.

[8] M. Parker, L. Ran, and S. Finney, "Distributed control of a fault-tolerant modular multilevel inverter for direct-drive wind turbine grid interfacing," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 2, pp. 509–522, Feb. 2013.

[9] F. Filho, L. Tolbert, Y. Cao, and B. Ozpineci, "Real-time selective harmonic minimization for multilevel inverters connected to solar panels using artificial neural network angle generation," *Industry Applications, IEEE Transactions on*, vol. 47, no. 5, pp. 2117–2124, 2011.

[10] T. T. Johnson and S. Mitra, "Safe flocking in spite of actuator faults using directional failure detectors," *Journal of Nonlinear Systems and Applications*, vol. 2, no. 1-2, pp. 73–95, Apr. 2011.

[11] G. G. Wang, "Definition and review of virtual prototyping," *Journal of Computing and Information Science in Engineering*, vol. 2, no. 3, pp. 232–236, Jan. 2003.

[12] T. T. Johnson, "Fault-tolerant distributed cyber-physical systems: Two case studies," Master's thesis, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, May 2010.

[13] L. U. Gokdere, K. Benlyazid, R. A. Dougal, E. Santi, and C. W. Brice, "A virtual prototype for a hybrid electric vehicle," *Mechatronics*, vol. 12, no. 4, pp. 575–593, 2002.

[14] Q. Li, F. Lee, and T. Wilson, "Design verification and testing of power supply system by using virtual prototype," *Power Electronics, IEEE Transactions on*, vol. 18, no. 3, pp. 733–739, May 2003.

[15] C. Alexandru and C. Pozna, "Virtual prototype of a dual-axis tracking system used for photovoltaic panels," in *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*, June 2008, pp. 1598–1603.

[16] B. Urmila and D. Subbarayudu, "Multilevel inverters: A comparative study of pulse width modulation techniques," *Journal of scientific and engineering research*, vol. 1, no. 13, pp. 1–5, 2010.

[17] C. Turpin, P. Baudesson, F. Richardeau, F. Forest, and T. Meynard, "Fault management of multicell converters," *Industrial Electronics, IEEE Transactions on*, vol. 49, no. 5, pp. 988–997, Oct. 2002.

[18] A. Chen, L. Hu, L. Chen, Y. Deng, and X. He, "A multilevel converter topology with fault-tolerant ability," *Power Electronics, IEEE Transactions on*, vol. 20, no. 2, pp. 405–415, 2005.

[19] F. Chan and H. Calleja, "Reliability: A new approach in design of inverters for pv systems," in *International Power Electronics Congress, 10th IEEE*, Oct. 2006, pp. 1–6.

[20] M. Ma, L. Hu, A. Chen, and X. He, "Reconfiguration of carrier-based modulation strategy for fault tolerant multilevel inverters," *Power Electronics, IEEE Transactions on*, vol. 22, no. 5, pp. 2050–2060, Sep. 2007.

[21] S. Khomfoi and L. Tolbert, "Fault diagnostic system for a multilevel inverter using a neural network," *Power Electronics, IEEE Transactions on*, vol. 22, no. 3, pp. 1062–1069, May 2007.

[22] A. Ristow, M. Begovic, A. Pregelj, and A. Rohatgi, "Development of a methodology for improving photovoltaic inverter reliability," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 7, pp. 2581–2592, Jul. 2008.

[23] P. Lezana, J. Pou, T. Meynard, J. Rodriguez, S. Ceballos, and F. Richardeau, "Survey on fault operation on multilevel inverters," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 7, pp. 2207–2218, Jul. 2010.

[24] Y. Song and B. Wang, "Survey on reliability of power electronic systems," *Power Electronics, IEEE Transactions on*, vol. 28, no. 1, pp. 591–604, Jan. 2013.

[25] S. Harb and R. Balog, "Reliability of candidate photovoltaic module-integrated-inverter (pv-mii) topologies–a usage model approach," *Power Electronics, IEEE Transactions on*, vol. 28, no. 6, pp. 3019–3027, 2013.

[26] Y. Zhao, J. de Palma, J. Mosesian, R. Lyons, and B. Lehman, "Line-line fault analysis and protection challenges in solar photovoltaic arrays," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 9, pp. 3784–3795, 2013.

[27] J. Fischer, S. Gonzalez, M. Herran, M. Judewicz, and D. Carrica, "Calculation-delay tolerant predictive current controller for three-phase inverters," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 233–242, 2014.

[28] S. Dhople, A. Davoudi, A. Dominguez-Garcia, and P. Chapman, "A unified approach to reliability assessment of multiphase dc-dc converters in photovoltaic energy conversion systems," *Power Electronics, IEEE Transactions on*, vol. 27, no. 2, pp. 739–751, Feb 2012.

[29] S. Bolognani, M. Zordan, and M. Zigliotto, "Experimental fault-tolerant control of a pmsm drive," *Industrial Electronics, IEEE Transactions on*, vol. 47, no. 5, pp. 1134–1141, 2000.

[30] X. Kou, K. A. Corzine, and Y. L. Familiant, "Full binary combination schema for floating voltage source multi-level inverters," in *Industry Applications Conference, 2002. 37th IAS Annual Meeting. Conference Record of the*, vol. 4. IEEE, 2002, pp. 2398–2404.

[31] B. Johnson, P. Krein, and P. Chapman, "Photovoltaic ac module composed of a very large number of interleaved inverters," in *Applied Power Electronics Conference and Exposition (APEC), 2011 Twenty-Sixth Annual IEEE*, Mar. 2011, pp. 976–981.

[32] T. Esram and P. Chapman, "Comparison of photovoltaic array maximum power point tracking techniques," *Energy Conversion, IEEE Transactions on*, vol. 22, no. 2, pp. 439–449, June 2007.

[33] H. Patel and V. Agarwal, "MATLAB-based modeling to study the effects of partial shading on PV array characteristics," *Energy Conversion, IEEE Transactions on*, vol. 23, no. 1, pp. 302–310, Mar. 2008.

[34] M. Ropp and S. Gonzalez, "Development of a MATLAB/Simulink model of a single-phase grid-connected photovoltaic system," *Energy Conversion, IEEE Transactions on*, vol. 24, no. 1, pp. 195–202, Mar. 2009.

[35] K. Ding, X. Bian, H. Liu, and T. Peng, "A MATLAB-Simulink-based PV module model and its application under conditions of nonuniform irradiance," *Energy Conversion, IEEE Transactions on*, vol. 27, no. 4, pp. 864–872, Dec. 2012.

[36] A. Maki and S. Valkealahti, "Effect of photovoltaic generator components on the number of MPPs under partial shading conditions," *Energy Conversion, IEEE Transactions on*, vol. 28, no. 4, pp. 1008–1017, Dec. 2013.

[37] A. Yazdani and R. Iravani, *Voltage-Sourced Converters in Power Systems : Modeling, Control, and Applications*. Wiley, 2010.

[38] R. Gonzalez, J. Lopez, P. Sanchis, and L. Marroyo, "Transformerless inverter for single-phase photovoltaic systems," *IEEE Transactions on Power Electronics*, vol. 22, no. 2, pp. 693–697, Mar. 2007.

[39] P. Sun, "Cascade dual-buck inverters for renewable energy and distributed generation," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2012.

[40] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995.

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

[42] D. K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager, *The Theory of Timed I/O Automata*, ser. Synthesis Lectures in Computer Science. Morgan & Claypool, 2006.

[43] T. T. Johnson, "Uniform verification of safety for parameterized networks of hybrid automata," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL 61801, 2013.

[44] T. T. Johnson, Z. Hong, and A. Kapoor, "Design verification methods for switching power converters," in *Power and Energy Conference at Illinois (PECI), 2012 IEEE*, Feb. 2012, pp. 1–6.

[45] S. Hossain, S. Dhople, and T. T. Johnson, "Reachability analysis of closed-loop switching power converters," in *Power and Energy Conference at Illinois (PECI)*, 2013, pp. 130–134.

[46] L. V. Nguyen and T. T. Johnson, "Benchmark: Dc-to-dc switched-mode power converters (buck converters, boost converters, and buck-boost converters)," in *Applied Verification for Continuous and Hybrid Systems Workshop (ARCH 2014)*, Berlin, Germany, Apr. 2014.

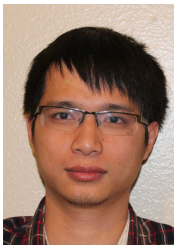
[47] M. Althoff, "Formal and compositional analysis of power systems using reachable sets," *Power Systems, IEEE Transactions on*, no. 99, pp. 1–11, 2014.

[48] T. T. Johnson, S. Mitra, and K. Manamcheri, "Safe and stabilizing distributed cellular flows," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS)*. Genoa, Italy: IEEE, June 2010.

[49] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[50] R. P. Severns and G. Bloom, *Modern DC-to-DC Switchmode Power Converter Circuits*. New York, New York: Van Nostrand Reinhold Company, 1985.

[51] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, 2nd ed. Springer, 2004.



Luan Viet Nguyen Luan Viet Nguyen is a PhD Student in Computer Science and Engineering at the University of Texas at Arlington. He graduated with honors in May 2013 from the MSc program at the Catholic University of America, where he had previously earned his BSEE in December 2012. He is currently working in the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory (VeriVITAL), under the supervision of Prof. Taylor T. Johnson. His main research interests are cyber-physical system (CPS), distributed system, formal method, and control theory, with a focus on conducting reachability analysis and building software tools to enhance the reliability of CPS.

Hoang-Dung Tran Hoang-Dung Tran is an Assistant Lecturer in Automatic Control at Ho Chi Minh University of Transport and is a research assistant in the Automation Group at Ton Duc Thang University, Vietnam. He earned his BSEE from the Gifted Undergraduate Program at Ho Chi Minh University of Technology, Vietnam in December 2007. He graduated from the MSc program in July 2013 at the Huazhong University of Science and Technology, China. His research interests are cyber-physical systems, networked control systems, multi-agents systems, and control theory.

Taylor T. Johnson Taylor T. Johnson is an Assistant Professor of Computer Science and Engineering at the University of Texas at Arlington (since September 2013), where he directs the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory (VeriVITAL). Dr. Johnson completed his PhD in Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign in 2013, where he worked in the Coordinated Science Laboratory with Prof. Sayan Mitra. Dr. Johnson completed his MSc in ECE at Illinois in 2010, earned a BSEE from Rice University in 2008, was a visiting graduate research assistant at the Air Force Research Laboratory (AFRL)'s Space Vehicles Directorate at Kirtland Air Force Base in 2011, and was a visiting faculty researcher at the AFRL's Information Directorate in 2014. Dr. Johnson worked in industry for Schlumberger at various times between 2005 and 2010 helping develop new downhole embedded control systems. Dr. Johnson's research focus is developing algorithmic techniques and software tools to improve the reliability of cyber-physical systems. Dr. Johnson has published over twenty papers on these methods and their applications in areas like power and energy systems, aerospace, and robotics, two of which were recognized with best paper awards, from the IEEE and IFIP, respectively.