Unified Invariants for Cyber-Physical Switched System Stability

Tamal Paul, Jonathan W. Kimball, Maciej Zawodniok, Thomas P. Roth, Bruce McMillin, and Sriram Chellappan

***Abstract*—Cyber-physical systems (CPS) consist of subsystems of distributed computation interconnected by computer networks that monitor and control switched physical entities interconnected by physical infrastructures. Finding a common semantic among these diverse subsystems that facilitates system synthesis, verification, and monitoring is a significant challenge of a CPS research program. Logical and temporal correctness of computational components, network timing, and frequency response are all system aspects that conspire to impede design, verification, and monitoring. Most current approaches ensure that each subsystem meets its individual specifications according to relevant metrics—stability of a physical system, safety and liveness of a cyber system, etc.—and then composes the overall system by functionality. The individual specifications are given in different semantics for each type of subsystem, and are in general equivalent to the cyber notion of correctness. This paper develops common semantics that span each aspect of a CPS through a new approach, unified invariants; unified invariants also ensure individual subsystem correctness but compose the overall system through logical truth instead of functionality. These individual invariants express and enforce system correctness common to the cyber, physical, and networking CPS subsystems and unified invariant approach ensures that the subsystems do not interfere with each others' correctness. In particular, the synthesis of switched dynamic CPSs will be unified by cyber, networking, and physical invariants rooted in the principal of Lyapunov-like functions. The goal is to make the resulting CPSs will be safe and stable at the system level, rather than just the subsystem level.**

***Index Terms*—CPS, invariant, Lyapunov-like.**

I. Introduction

THE TIGHT conjoining of, and coordination between, computational resources and physical components represents the core of cyber-physical systems (CPS) research. A wide variety of CPS challenge problems have been identified through numerous workshops over the last decade. These include autonomous systems [such as unmanned air vehicles (UAVs)] and large scale distributed coordination (such as automated traffic control and future generation smart electrical grid systems) that are highly efficient (such as renewable resource coordination) and often are comprised of switched systems. Common to these systems are four principal functional components: a (distributed) cyber component, a networking/communications component, an underlying physical infrastructure, and significant timing control. Integration, correctness, stability are significant challenges in conjoining these four components.

Integration is particularly problematic in CPS development. Incorrectness, stability, timing, and fault issues in one component can significantly impact the same features throughout the entire CPS. In current practice, taking all component functionalities together in CPS design is unwieldy due to complexity and composition issues of certain properties. As such, a unifying framework for CPS design is highly desirable, but as yet no comprehensive framework exists. What is missing is a semantically common method of relating cyber, network, and physical actions and dynamics. Some work is breaking through this barrier. Acumen [1] bridges the gap between analytic models and simulation codes. Invariants and predicate transformers on the state of CPS was explored for dynamical systems in [2] and more recently in [3] which gives a formalism for invariant interaction and incremental invariant composition. The interaction of invariants for purely cyber processes, has its origins in [4] which affords composition of sequential proofs governed by the property of noninterference.

Our foundational approach is fundamentally different, composing correctness instead of functionality. Logical *Invariants* define correctness of the CPS system. Thus, we constructs CPSs through compositional integration of the cyber, networking, scheduling, and physical components rooted in correctness (Fig. 1).

A. Project Background

Most work in CPS control systems, such as the networked control systems work surveyed in [5], focuses on the plant model of Fig. 2 in which a physical system (the plant) is acted on by a control system, potentially with either sensor readings or control settings transmitted over a computer network.

In this control systems work, network delays affect system stability and considerable work focussed on determining system stability bounds as a function of injected delay [5], sometimes using results from switched-systems theory as in [6]. In most cases, however, the controller is modeled using continuous-time or discrete-time state-space models which abstract away much of the underlying complexity of cyber control systems. As CPSs mature, the control consists of complex distributed calculations that are not easily represented in the same semantics as simple plant/network delay model. Hybrid automata [7] and timed I/O

Manuscript received August 17, 2012; revised January 21, 2013; accepted August 21, 2013. Date of publication December 12, 2013; date of current version December 24, 2013. This work was supported by the Future Renewable Electric Energy Delivery and Management Center, a National Science Foundation supported Engineering Research Center, under grant NSF EEC-081212; the Missouri S&T Intelligent Systems Center; and the Missouri S&T Energy Research and Development Center. Paper no. TSG-00504-2012.

T. Paul, J. W. Kimball, and M. Zawodniok are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA.

T. P. Roth, B. McMillian, and S. Chellappan are with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: ff@mst.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2013.2283171

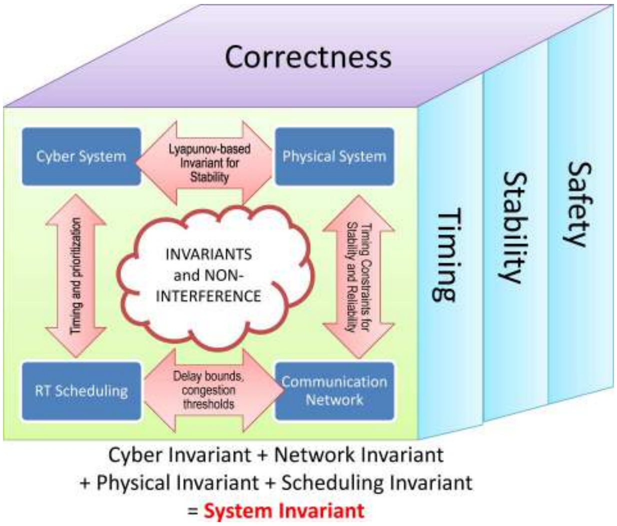
Correctness
Cyber System
Lyapunov-based Invariant for Stability
Physical System
Timing and prioritization
INVARIANTS and NON-INTERFERENCE
Timing Constraints for Stability and Reliability
RT Scheduling
Delay bounds, congestion thresholds
Communication Network
Timing
Stability
Safety
Cyber Invariant + Network Invariant
+ Physical Invariant + Scheduling Invariant
= System Invariant

Fig. 1. Interaction of communication network, cyber, scheduling, and physical systems.

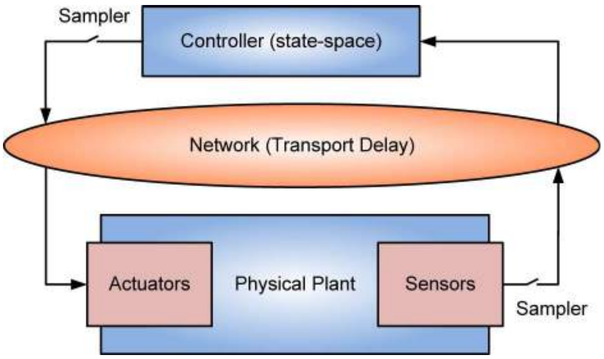

Fig. 2. Generic networked control system analyzed in e.g., [5].

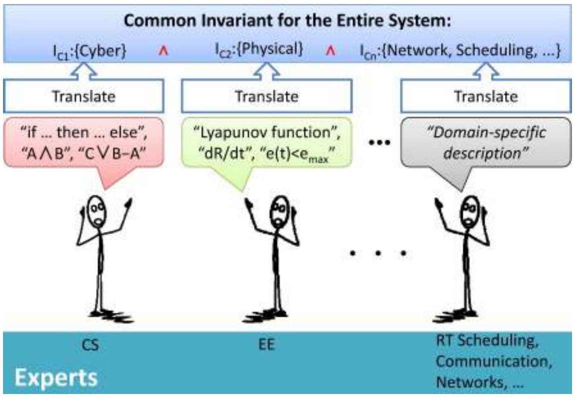
Common Invariant for the Entire System:
I_{C1}:{Cyber} ∧ I_{C2}:{Physical} ∧ I_{Cn}:{Network, Scheduling, ...}
Translate
Translate
Translate
"if ... then ... else", "A∧B", "C∨B–A"
"Lyapunov function", "dR/dt", "e(t)<e_{max}"
•••
"Domain-specific description"
CS
EE
RT Scheduling, Communication, Networks, ...
Experts

Fig. 3. Overview of the proposed, invariant-based approach.

automata [8] represent a simultaneous mix of continuous and discrete states the verification process [9], [10]. Markov jump systems (MJSs) show promise. Like a hybrid system, an MJS is a combination of a continuous-time system and a finite state machine whose state affects the continuous-time system. The difference is that the discrete portion is a Markov process rather than deterministic. Stability results have been shown for linear systems (MJLS) [11], [12] but there are fewer results for an MJS in which the underlying continuous-time system is nonlinear, as in this paper.

As an alterative to MJS, Lyapunov functions have been applied to describe physical system dynamics [13]. More recently, Lyapunov functions describe complex systems such as the power grid [14]. Lyapunov functions collapse the continuous system behavior into a scalar function. Unfortunately, for a nonlinear system there are no generic methods for finding a true Lyapunov function. If a Lyapunov function cannot be found, a common substitute is to use either the norm of the state vector or the energy of the system as a proxy for a true Lyapunov function. If this function is *Lyapunov-like* [15] conclusions can be drawn about stability. In other words, the stability bounded by a Lyapunov-like function is an invariant on the physical system state.

An *invariant* is a logical predicate on a system state that should not change its truth value if satisfied by the system execution. An axiomatic basis for the truth of invariants on cyber systems was first proposed by [16]. In this system, program actions are related to logical truth through axioms and inference rules. Invariants are widely used ranging from algorithm instruction [17] to never claims in model checking [18]. Invariants are well-understood for cyber processes, but extending them into the network and physical domains requires some insight.

Lyapunov functions have also been used to describe network stability [19]–[21]. Conceptually, Lyapunov-like functions can be constructed by modeling network traffic as a control feedback problem and then bounding the number and timing of outstanding messages and/or acknowledgements.

Combining the physical and network Lyapounov-like functions and invariants with the cyber invariant concept yields a CPS system invariant, a semantically common method relating the cyber, physical, and network components. If the composition can be made noninterfering, the resulting invariant governs the entire system operation as in Fig. 1.

II. Background

This section provides background on the idea of invariants from their origins in the cyber world for program correctness and how they can be used within switched systems and within the network to represent stability as correctness invariants. Key to understanding the proposed idea is the logical linkage among invariants. Fig. 3 depicts the development, contribution, and conjoining of individual invariants into a system invariant.

A. Cyber Invariants and Interference

From the cyber perspective, fundamental to system design is an understanding of the system's requirements specification. Axiomatic specifications are one way to represent these requirements and their associated proof systems for computer programs have their origins in the 1960s [16]. Key to these systems is a set of axioms and inference rules that relate program statements to logical truth. Of particular interest are invariants, or logical statements that must remain true throughout a system's execution. Individual programs may be composed into concurrently executing sets of statements, S_1, S_2, S_n. Invariants, however, cannot be arbitrarily composed. To handle potential conflicts, Noninterference NI is used. To show Noninterference requires showing that for all actions a in some statement S_i and all invariants P_{jk}, $pre(a) \wedge P_{jk}$ followed by action a, P_{jk} remains true. This is a powerful technique as it allows for composition of proofs in building a system instead of composition of statements. Prior work has shown that these proofs can span different cyber system aspects, such as timing and frequency [22], and that individual proofs can be composed together via noninterference. Conceptually, a cyber system can be composed with a physical system and a network system and shown to be stable through the composition of invariants. These cyber invariants, I_C on the correctness of the system form the upper left of Fig. 1.

B. Switched System Stability for Physical Systems

A switched system is a fundamentally continuous-time system with changes that occur at discrete times [23]. A classic example is a bouncing ball: its dynamics are governed by gravity and Newton's laws, and its velocity changes direction (instantaneously, as approximated) when it hits a surface. The switching instants may be related to the system dynamics, as in the ball example, or may be externally imposed. A switched system is distinguished from a hybrid system in that discrete state dynamics are not modeled.

Switched system analysis can identify switching sequences that are allowable and switching sequences that cause instability. The switching sequences may be restricted in the state space or in the time domain. This subsystem summarizes some key concepts used to analyze the stability of a switched system to provide context and notation for the cyber-physical analysis in the sequel.

A continuous-time system may be modeled using a vector of state variables ($\mathbf{x} \in \mathbb{R}^n$), a vector of inputs $\mathbf{u}$ that may be exogenous or internally generated via feedback, and a vector of measurable outputs, $\mathbf{y}$. The state-space formulation of a system, for some (possibly nonlinear) vector-valued functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ and some initial condition $\mathbf{x}_0$ on $\mathbf{x}$ at time t_0, is

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{1}$$

A well-known tool for stability analysis of an autonomous continuous system (that is, one with no external inputs $\mathbf{u}$) is a Lyapunov function, $V(\mathbf{x})$. A Lyapunov function is positive definite ($V(\mathbf{x}) > 0 \ \forall \mathbf{x} \neq 0, V(0) = 0$), radially unbounded, and non-increasing ($dV/dt = \partial V / \partial x \mathbf{f}(\mathbf{x}) \leq 0$). If dV/dt is non-positive, the system is stable. If dV/dt is strictly negative, the system is asymptotically stable.

Unfortunately, finding Lyapunov functions for high-order systems poses significant computational challenges and no general techniques exist for non-linear systems. For a switched system, another class of functions, *Lyapunov-like* functions [15], [24], [25], may be considered. A Lyapunov-like function must be positive definite and radially unbounded, just like a Lyapunov function. However, its derivative need not be negative. Instead, we are only concerned with its value at isolated points. Multiple Lyapunov-like functions may be used for different operating modes.

Consider a switched system that may operate in several different modes, enumerated by k. For each mode, define a Lyapunov-like function $V_k(\mathbf{x}_k)$. If the system switches between modes, the only values of $V_k(\mathbf{x}_k)$ that matter are the values at the instant when the k^{th} mode becomes active. If those values form a decreasing or non-increasing series, and the same holds true for all admissible values of k, then the switched system is stable. The use of multiple Lyapunov or Lyapunov-like functions to prove switched system stability is shown conceptually in Fig. 4. In Fig. 4(a), both modes may be described with true Lyapunov functions, and their decreasing values at switch-in times verifies overall stability. In Fig. 4(b), one mode may or may not be stable, as V_1 is not a true Lyapunov function, but the overall system is stable because of the decreasing sequences.

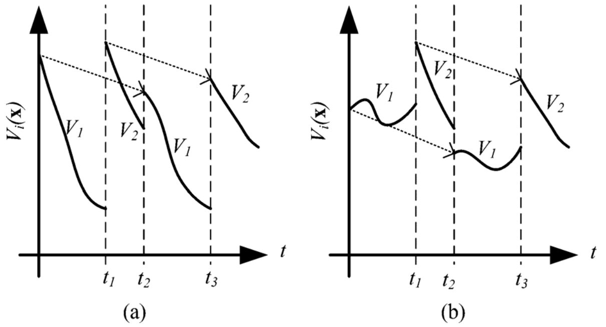

Fig. 4. Conceptual use of multiple Lyapunov functions (V_1 and V_2) to show stability of a switched system. (a) Two true Lyapunov functions. (b) One Lyapunov function (V_2), one Lyapunov-like function (V_1).

Lyapunov-like functions are proposed here for analysis of the physical portion of a complex CPS because of their suitability for integration with analysis of the cyber portion. For each operating mode of the physical sub-system, a Lyapunov-like function can be defined, such as the energy in the error in all the state variables. Then, as the cyber system state evolves, the Lyapunov-like functions can be checked. The derivative of the Lyapunov-like functions can be monitored and used to determine minimum and maximum times between switching instants. Instead of considering the dynamics of many state variables, analysis can focus on a single scalar function of those state variables. This scalar function, along with any other restrictions or assumptions involved in its derivation, becomes an invariant I_P on the stability of the system, as in the upper right corner of Fig. 1.

III. Invariant-Based Approach

When treated in isolation, it is not clear why certain constraints are present in a system; for example, why is the delay X or why must scheduling deadline D be met? These questions can only be partially answered by developing individual invariants; the response of the entire CPS is based on the composition of these invariants. Thus, the proposed research develops an approach to compositionally integrate the CPS system aspects shown in Fig. 1. In each aspect, cyber, physical, scheduling, and network, stability invariants, based on individual system theories are developed. These invariants are refined through the property of noninterference to develop a CPS. The next few sections discuss the idea and approach of developing a CPS through interaction of invariants through a motivating example.

A. Primitive Cyber Algorithm

Consider the architecture of a future generation smart grid (SmartGrid) [26] system shown in Fig. 5. A SmartGrid forms

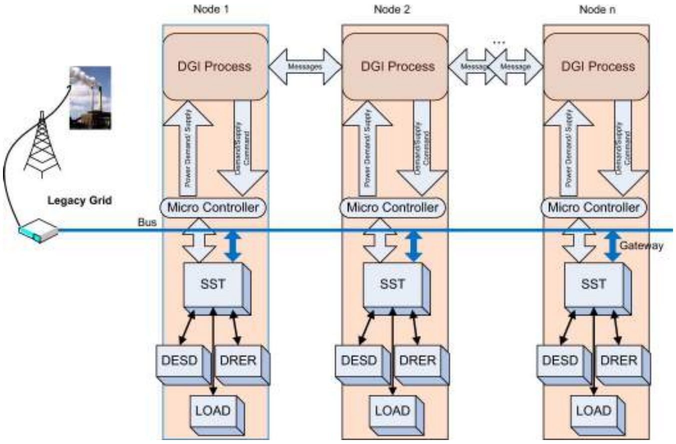

Fig. 5. SmartGrid power management architecture.

a microgrid of distributed energy storage devices (DESD) distributed renewable energy resources (DRER), and LOADs (programmable and non-programmable loads) to share power for the good of the entire system. Intelligent flow controllers (Nodes) contain physical actuators such as solid state transformers (SSTs) that control power flow to and from a shared electrical bus, under direction of cooperating Distributed Grid Intelligence Processes $(DGIs)$. As such, the SmartGrid is a switched system.

Each Node is potentially owned and located in a residence or business. Within the Nodes, the DGI processes compute a power cost and use a drafting process [27]. Drafting is a receiver-initiated load-balancing procedure; if a Node has available power generation capacity, it solicits bids from Nodes that demand power. Algorithm 1 represents a simplified version of this power management algorithm written in a communicating sequential processes (CSP)-like language [28]. Each of n Nodes contains a computational process that executes sequential algorithm code (lines 3–31 below) that communicates with its $n-1$ peer processes by send or receive message passing. The algorithm is triggered by a transition in the state of the underlying power system, either $high$ or low as measured in line 8.

Algorithm 1: PowerBalance, Symbols from the CSP language used are in **Appendix A**

```
1  **PowerBalance** $P_i$ ::
2    **var** $k = 0$
3    **do**
4      Until $t_{cycle}$ expires
5      $\{I_{C1} : n\nu + \sum_l^n high_l + \sum_l^n low_l = g\}$
       // Power Flow Invariant
6
7      $\{I_{C2} : \exists_{l,m} max\ high_l^r - max\ high_m^{r+1} < 0, r = 0, \ldots, k-1\}$   // Knapsack
          //Invariant
8      $status = input()$
       /* A process in $low$ status
       executes this code segment   */
9
10     $status = low \rightarrow broadcast\_request$
11     **do**
12       $\forall l = 1, \ldots, n$   // Receive responses
           from any processes
13
14         $P_l?response[l]$
15         $\Box P_{l+1}?response[l+1]; \Box \ldots$
16       $sort(response), highestcorrespondingtoP[j]$
17       $P_j!select$    // Send the Winning
         Migration
18
19       $low_i = low_i + \delta$
20       $migrate(\delta, j)$   // Command the local
         device to transfer Power to $j$
21
22       $\{Q_{C1} : n\nu + \sum_l^n high_l + \sum_l^n low_l \geq g\}$
         /* A process in $high$ status
         executes this code segment /*
23
24       $\Box status = high \wedge P_i?request \rightarrow P_i!response$
25       $\Box status = high \wedge P_i?select \rightarrow$   // Receive
         the Winning Migration
26
27       **do**
28         $migrate(\delta, j)$   // Command the local
           device to receive Power from $j$
29
30         $high_i = high_i - \delta$
31       $k = k + 1$
32       $\{I_{C1}, I_{C2}\}$
33     $\{I_{C1}, I_{C2}\}$
```

If the Node is in the low state, it has low demand and can supply power and executes the code in lines 10–20. In line 10, P_i broadcasts an offer to supply to the other $n-1$ processes. In lines 11–15, P_i awaits for responses from these processes. If it receives a response, in lines 16–17 of P_i it awards power to the Node that has the most demand by selecting it (sends a $P_j!select$ message).

Any Node, P_j in the $high$ state has a high demand and executes the code in lines 24–30. In line 24 it receives an offer $(P_j?request)$ and responds to the broadcaster, P_i by sending a response $(P_i!response)$. If P_j is still in the $high$ state, it accepts the message $[P_i?select$ in line 25)] to agree to a migration.

When both P_i and P_j have agreed to migrate power, each executes the function $migrate(\delta, i)/migrate(\delta, j)$ which is a command to the underlying power system to provide/accept a quantum of power to/from a node i/j. The algorithm runs as many times as possible until a predetermined t_{cycle} (line 4) is reached and which time other activities are scheduled.

At all times the system must correspond to Kirchoff's laws, so the cyber algorithm must maintain the invariant at line 5, $\{I_{C1} : n\nu + \sum_l^n high_l + \sum_l^n low_l = g\}$ (where ν is the nominal load per Node) which indicates that the sum of low demand (supply) and high demand must equal g which represents the excess draw or supply to/from a grid connection. The sort at line 16 causes a greedy selection of the migration, and, as such, approximates a distributed solution to the fractional knapsack problem [29]. The fractional knapsack problem requires the invariant at line 7,

$\{I_{C2} : \exists_{l,m} max\ high_l^r - max\ high_m^{r+1} < 0, r = 0, \ldots, k-1\}$, in other words, as the algorithm executes, the imbalance between the total amount of supply vs. the total amount of demand decreases.

The invariant I_{C2} holds at termination of the algorithm, by the greedy choice principle. However, due to lack of strict synchronization, assignment $low_i = low_i + \delta$ interferes with the truth of the invariant I_{C1} and assertion Q_{C1} until the migration has been received at process P_j. Potentially, kQ_{C1} (and I_{C1}) must be relaxed to:

$$\left\{ \frac{Q'_{C1}}{I'_{C1}} : \{n\nu + \sum_{l}^{n} high_l + k\delta + \sum_{l}^{n} low_l = g \right\}.$$

Since, in this model, when a rendezvous occurs at the select message, this modified invariant becomes true in the receiving process (by the rule of satisfaction [30]). Further relaxation of PowerBalance, moving to asynchronous communication, introduces an additional interference — the PowerBalance invariant can be invalidated indefinitely unless satisfaction can be obtained by guaranteeing receipt of the select message.

B. Lyapunov Function Development for Integrated Physical and Network System

Continuing with the motivating example of Section III-A, a simplified version of the SmartGrid system described in [26] (Fig. 5) was implemented with the physical systems model simulated using PSCAD as shown in Fig. 6. The system is comprised of three converters (all in three phase) interconnected through lines of reasonable impedance and eventually connected to a generator by a transformer. Among the three converters, two converters inject power (source) into the grid and act as inverters. One converter delivers power out of the grid (load), thereby acting as a rectifier. The gate signals to the converters were designed keeping in mind the desired control strategy which is to produce the power as commanded by the cyber algorithm. The parameters of the microgrid are:

- Generator: rated 39.8 kV line-to-neutral, 837 A, inertia constant 31.17 s
- Transformer: 100 MVA, Y-Y, leakage reactance 0.1 pu, 69 kV:100 kV
- Lines: $3.97\ \Omega$, $6.89\ \mathrm{m}\mathbf{H}$
- SST: 120 kV nominal, with LCL filter including $4.2\ \mu\mathbf{F}$ and $10\ \mathrm{K}\Omega$

This is a model of a finite-inertia system, in which power imbalance may result in frequency changes. Real-world examples of such a system include military forward operating bases, electric ships, and grid-connected systems with sufficient local generation to support off-grid operation for short times.

As a proof of concept, the system under consideration has only one generator and a small number of solid-state transformers (SSTs). Using the methods of [31], this concept could be extended to more complex systems, such as systems with many more SSTs and multiple generators. In such systems, the dynamic model is of a higher order and the energy function is more complex, but the underlying physical principles are the same.

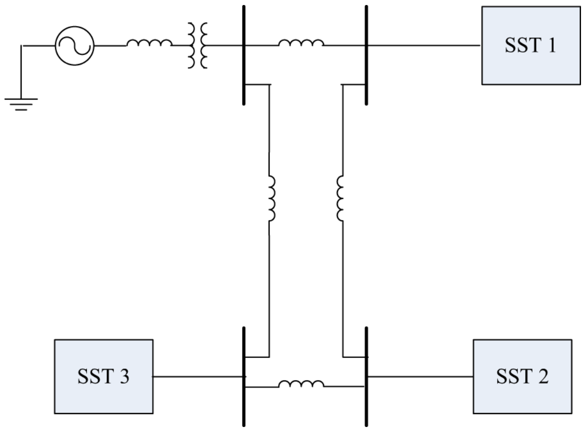

Fig. 6. Topology of the three-phase three converter grid system.

In this system each pair of nodes communicate over a network and are both connected to a microgrid, defined here as a system with limited power generation capability.

The power flow on the grid is balanced internally, except for any transfers between nodes. The communication between the two nodes (labeled "source" and "load," or S and L) involves a request from L for power from S. Each request is for one quantum of power δ. If the message from L to S is dropped, nothing happens. If S receives the message, it increases its power output by δ and sends an acknowledgment to L. Whenever L receives a valid message, it increases its load by δ. Messages are then sent at a rate of λ and received at a rate of μ (after accounting for transit time and queueing), for an average delay time R_d as described below. S is actively migrating power and $\lambda \geq \mu$. For a brief time at the beginning of the communication process, $\mu = 0$ due to the transit time.

As discussed in more detail in [32], the dynamics of the average number of messages from S to L that have been sent but not received, K, can be modeled as $dK/dt = \lambda - \mu$ where $K \geq 0$ by definition. Therefore, the net imbalance in power (i.e., the difference between the total generated power and the total load power) is approximately $P_{imb} = \delta K$. This net imbalance will tend to increase the grid frequency, ω, above the nominal frequency, ω_0. The dynamics are governed by a simplified swing equation,

$$\begin{aligned} \frac{d\omega}{dt} &= -\frac{V_1 V_2}{J\omega X} sin(\theta - \theta_0) - \frac{D}{J}(\omega - \omega_0) + \frac{P_{imb}}{J\omega} - \frac{kP^2}{J\omega}, \\ \frac{d\theta}{dt} &= \omega - \omega_0, \end{aligned} \tag{2}$$

where $\theta - \theta_0$ is the relative electrical angle between the generator's internal voltage and terminal voltage, D is the natural damping due to frequency-sensitive loads, J is the effective rotational inertia, and k models the line losses due to the gross power flow, P. Other terms relate to the lumped equivalent generator: V_1 is the internal generated voltage, V_2 is the terminal voltage, and X is the synchronous reactance. To ensure stability, a droop law of the form

$$P_S = P_{request} - m(\omega - \omega_0) \tag{3}$$

is used, where $P_{request}$ is the desired power generation or load as determined by the distributed cyber algorithm, P_S is the actual power supplied or absorbed by the node, and m is a droop constant that may be adjusted. Due to the droop, the actual power imbalance, P_{imb}, will differ from δK by $m(\omega - \omega_0)$. A candidate Lyapunov-like function is

$$V(\omega, \theta) = \frac{J}{2}(\omega - \omega_0)^2 + \frac{V_1 V_2}{\omega X}(1 - cos(\theta - \theta_0)). \qquad (4)$$

In a real power system, there are many causes of instability, such as faults, impedance ratios, inadequate VAR support, and so forth [33]–[34]. For the present work, the assumption is that conventional stability analysis has been performed to ensure that the physical system is itself stable. However, the addition of a cyber layer adds the possibility of power imbalance, as derived above, that still destabilizes the system. The actual operation of the cyber controller or network could be more complex than is described here, but its impact on the physical subsystem enters through K regardless. Therefore, the focus of the physical subsystem analysis will be on the effect of K.

C. *Physical Invariant*

Depending on the average delay R_d (which causes a corresponding steady-state positive value of K, λR_d) and the communication protocol, V may begin to increase, which may indicate instability.

For the system to be stable, first, ω must be positive, a trivial condition given the physics of the system. Second,

$$\left\{I_{P1} : (\omega - \omega_0)^2 (D\omega + m) \right.$$
$$\left. +(\omega - \omega_0)(kP^2) > \delta K(\omega - \omega_0)\right\}. \qquad (5)$$

All terms have been arranged to be positive during over-frequency conditions. Whether this inequality holds depends on the system operating point, the correction rate (if nonzero), and the system parameters.

The objective is to formulate an invariant that may be used in the cyber analysis. If we are concerned only with asymptotic stability, we must use I_{P1} directly. If *boundedness* is sufficient, then the following invariant is appropriate:

$$\{I_P : I_{P1} \vee (V(\omega, \theta) < V_{bound}) \vee (V(t) \leq V(t_x))\} \qquad (6)$$

where V_{bound} is the maximum allowable value of V, $V(t)$ is the value of $V(\omega, \theta)$ at the present time and $V(t_x)$ is its value at the most recent previous violation of I_{P1} due to a large value of K.

The resulting cyber-physical system is synthesized by guarding the statement $P_j!select$ with the invariant $I_P \wedge I'_{C1}$. In other words, a migration is only performed if cyber-physical system is stable. Using the simulation, the following operational modes were explored.

1) Steady-state operation that satisfies the Lyapunov criterion that was derived results in stable operation.
2) Steady-state operation that clearly violates the Lyapunov criterion results in marginal stability or instability.
3) Switching between two operating points requires investigation of Lyapunov-like behavior. Depending on the timing, even switching between two stable modes can result in instability.

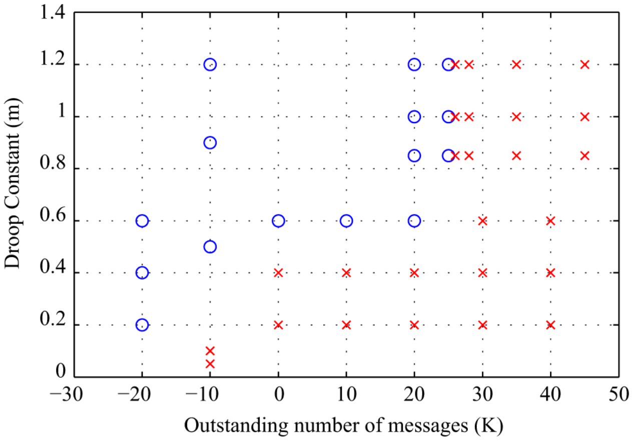

Fig. 7. Simulated system behavior. Circles indicate stable operation and crosses indicate unstable operation. For every simulated case, the truth value of $I_{P1} \wedge I'_{C1}$ corresponds to the PSCAD simulated results.

The simulation model is shown in Fig. 6. Parameters for (2–6) include: $J = 4.2\ kg \cdot m^2$, $V_1 = 39.8\ kV$, $V_2 = 39.326\ kV$, $X = 31.184\ \Omega$, $k = 0.0075$, $D = 0.025$, and $\delta = 100\ kW$. Power imbalance is δK, so for example, $K = 26$ corresponds to 2600 kW of imbalance.

First, the steady-state cases were examined. Each simulation used a different droop constant, m, and a different steady-state value of outstanding messages, K. Fig. 7 indicates which combinations are stable and which are unstable. In every case, the truth value of $I_{P1} \wedge I'_{C1}$ and the observed behavior were the same—if $I_{P1} \wedge I'_{C1}$ was false, the system was indeed unstable.

Next, switching cases were explored, with the droop constant fixed at $m = 1$. All cases followed a pattern of alternating power imbalance values, which might result from messages accumulating and being resolved. The two primary levels, $K = 4$ and $K = 6$, are both stable modes, that is, they satisfy I_{P1}. However, as is frequently observed in switched systems, the switching action can itself lead to instability. This touches on the concept of *minimum dwell time* [23]. “Dwell time” is the time spent in a given operating mode, here referring to a certain value of K. If the dwell time is sufficiently long, the states settle near their equilibrium and stability is assured as long as each mode is independently stable. The switching times used are listed in Table I. These values $(K \in \{4, 6\})$ also reflect a relatively well-designed network with only a small number of outstanding messages, and are therefore taken to be reasonable examples of CPS behavior.

In the first case, the dwell time in the $K = 4$ mode is sufficient that the dynamics decay to essentially zero. Therefore, as can be seen from 10, the switched system is stable. This verifies that “long enough” times between switching correspond nearly to steady-state performance as given in Fig. 7.

Next, the dwell time was decreased, but was adequate to achieve Lyapunov-like behavior. The lower waveform in Fig. 11 shows the energy function corresponding to the power imbalance pattern shown in the upper plot. After an initial increase, there is a net decrease in energy at the beginning of each mode. Therefore, the system is stable, and when the

TABLE I
SWITCHING TIMES FOR VALUES OF K FOR THE DIFFERENT SIMULATED CASES

K	**Fig. 10**	**Fig. 11**	**Fig. 12**
0	0.0	0.0	0.0
2	0.8	0.8	0.8
6	1.8	1.8	1.8
4	2.6	2.8	3.0
6	6.0	3.8	3.5
4	6.8	4.8	4.7
6	10.2	5.8	5.2
4	11.0	6.8	6.4
6	14.4	7.8	6.9
4	15.2	8.8	8.1

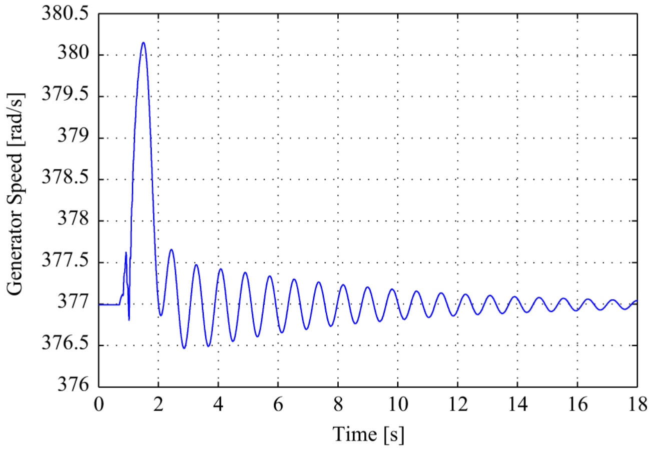

Fig. 8. Sample system response with $K = 26$ and $m = 1$, demonstrating stable behavior of generator speed (ω).

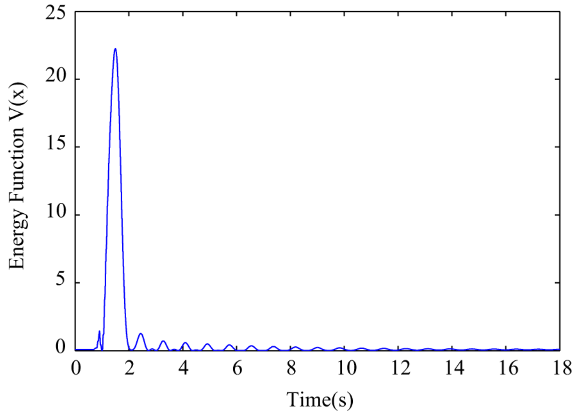

Fig. 9. Energy function $V(\omega, \theta)$ corresponding to Fig. 8.

switching ends, the steady-state performance is observed. This behavior equates to the last term in I_P, $V(t) \leq V(t_x)$. The relevant times (t_x) are indicated with circles on the trendlines, and correspond to changes in K. Two trendlines are indicated, corresponding to the two different modes between which the system switches.

As a test, the dwell time was decreased so that the invariant $I_P \wedge I'_{C1}$ is deliberately violated; thus, the energy function is no longer Lyapunov-like, as shown in Fig. 12. For some duration, the net growth still remains within the region of attraction. Eventually, though, $V > V_{bound}$ and the system is no longer stable (as may be seen after 8 s in Fig. 12).

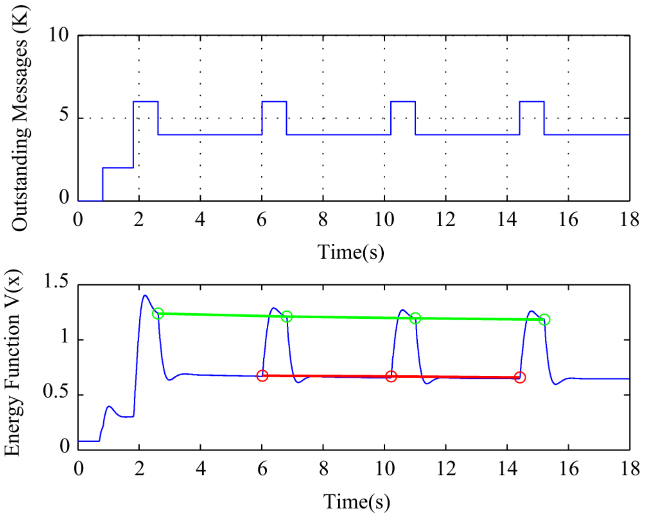

Fig. 10. Sufficient dwell time to achieve steady-state behavior. Trend lines indicate decrease at the switching instants.

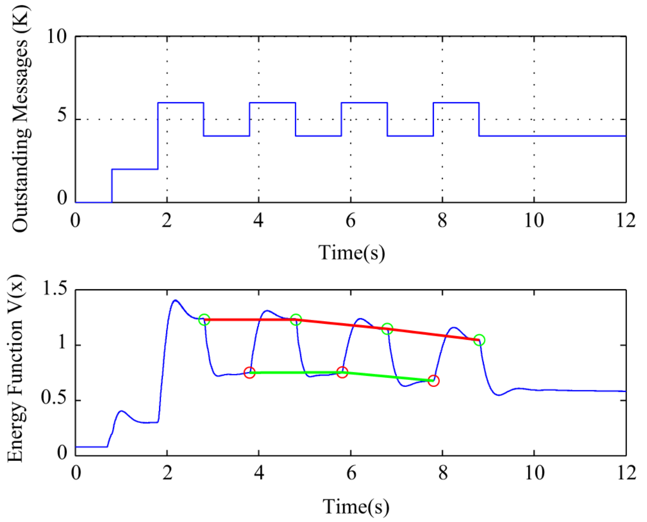

Fig. 11. Switched sequence showing Lyapunov-like stability, as indicated by decreasing trendlines at switching instants.

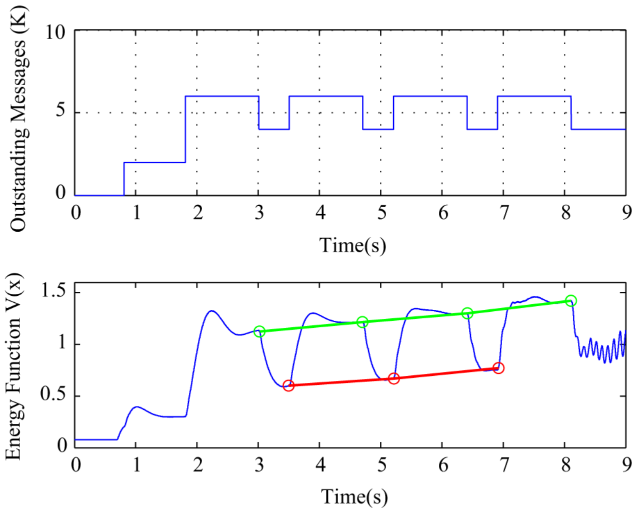

Fig. 12. Switched sequence violating Lyapunov-like criterion, with increasing trend at switching instants, and becoming unstable.

IV. Conclusion and Future Work

This paper showed how the unified invariants approach categorizes both cyber and physical stability so that a stable and safe cyber-physical system is compositionally constructed. Lyapunov-like functions constrain physical power system behavior and cyber invariants constrain algorithm behavior. For a test system, these invariants were used to determine stable and unstable switched system sequences. Switching is assumed to be deterministic, governed by a complex, distributed cyber process.

The analysis of this paper is limited to determining the effect on the system when a process is unresponsive (the omission fault model in which a process does not respond to a migration) or slow (network congestion, and delays in the cyber-physical interface), and, as such, increases the K value). The general fault case would include malicious (as in the general model of Byzantine [35] errors) or models of cyber-physical attack.

The next step is to determine analytical requirements that guarantee stability, such as the minimum dwell time or upper bound on the energy function for a given system with disturbances of a particular structure. After analyzing the effects of deterministic switching, conceptualizing a CPS as an MJS [36]–[38] with random switching will be explored.

Future work will compose cyber/physical, physical/network, and network/physical invariants as well as benign and malicious code failures (such as errors introduced by code that is not type-safe). These composed invariants will guard an adaptive real-time environment to ensure system correctness in a variety of decentralized, information poor, and degraded operating environments. The ultimate goal is to synthesize stable and safe cyber-physical systems using unified invariants.

Appendix
CSP Syntax

Communicating Sequential Processes (CSP) [28] was developed to describe the two fundamental concepts of distributed programs, namely sequential code segments that execute concurrently and communicated by rendezvous message passing. A portion of the CSP syntax relevant to this paper is shown below.

CSP Command	Meaning
$P_j!data\_item$	Send $data\_item$ to process P_j.
$P_i?data\_item$	Receive $data\_item$ from process P_i.
$C \rightarrow S$	Execute statement S if C evaluates to true. A communication in a guard evaluates to true if the communication will be successful.
$\Box C \rightarrow S$	Execute statement S if C evaluates to true. If many possible statements can be selected, pick one nondeterministically.

References

[1] Y. Zhu, E. Westbrook, J. Inoue, A. Chapoutot, C. Salama, M. Peralta, T. Martin, W. Taha, M. O'Malley, R. Cartwright, A. Ames, and R. Bhattacharya, "Mathematical equations as executable models of mechanical systems," in *Proc. 1st ACM/IEEE Int. Conf. Cyber-Physical Syst. (ICCPS '10)*, New York, 2010, pp. 1–11.

[2] M. Sintzoff and F. Geurts, "Analysis of dynamical systems using predicate transformers- attraction and composition," in *Analysis of Dynamical and Cognitive Systems Advanced Course Stockholm, Sweden, August 9–14, 1993 Proceedings*, 1993, vol. 888, Lecture Notes in Computer Science, pp. 227–260.

[3] S. Bensalem, A. Legay, T.-H Nguyen, J. Sifakis, and R. Yan, "Incremental invariant generation for compositional design," in *Proc. 4th IEEE Int. Symp. Theoretical Aspects Software Eng. (TASE)*, Aug. 2010, pp. 157–167.

[4] S. Owicki and D. Gries, "An axiomatic proof technique for parallel programs," *Acta Informatica*, vol. 6, pp. 319–340, 1976.

[5] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. IEEE*, vol. 95, pp. 138–162, 2007.

[6] M. C. F. Donkers, W. P. M. H. Heemels, N. V. D. Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Trans. Autom. Control*, vol. 56, pp. 2101–2115, 2011.

[7] T. A. Henzinger, "The theory of hybrid automata," in *Proc. IEEE Symp. Logic Comput. Sci.*, 1996, pp. 278–292.

[8] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Comp. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[9] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid system verification," *IEEE Trans. Autom. Control*, vol. 48, pp. 64–75, Jan. 2003.

[10] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proc. IEEE*, vol. 91, pp. 986–1001, Jul. 2003.

[11] V. P. Jilkov, X. R. Li, and D. S. Angelova, "Estimation of markovian jump systems with unknown transition probabilities through Bayesian sampling," *Lecture Notes in Computer Science*, vol. 2542, pp. 307–315, 2003.

[12] L. Zhang and E.-K. Boukas, "Stability and stabilization of Markovian jump linear systems with partly unknown transition probabilities," *Automatica*, vol. 45, no. 2, pp. 463–468, 2009 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109808004512

[13] C.-T. Chen, *Linear System Theory and Design*. New York: Holt, Rinehart and Winston, 1984.

[14] M. Roozbehani, M. Dahleh, and S. Mitter, "Robust and distributed decisions for future cyber-physical energy networks," in *New Res. Directions Future Cyber-Physical Energy Syst.* , Jun. 2009 [Online]. Available: http://www.ece.cmu.edu/ nsf-cps/file.php?id=87

[15] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 475–482, 1998.

[16] C. A. R. Hoare, "An axiomatic basis for computer programming," *Commun. ACM*, vol. 12, no. 10, pp. 576–585, Oct. 1969.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed ed. Cambridge, MA, USA: MIT Press, 2001.

[18] SPIN homepage [Online]. Available: http://spinroot.com/spin/whatispin.html.

[19] L. Massouli, "Structural properties of proportional fairness: Stability and insensitivity," *Ann. Appl. Probab.*, vol. 17, no. 3, pp. 809–839, 2007.

[20] M. Zawodniok and S. Jagannathan, "Predictive congestion control protocol for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 3955–3963, 2007.

[21] S. Jagannathan, *Wireless Ad Hoc and Sensor Networks: Protocols, Performance, and Control*. Boca Raton, FL, USA: CRC, 2007.

[22] Y. Sun, B. McMillin, X. F. Liu, and D. Cape, "Verifying noninterference in a cyber-physical system: The advanced electric power grid," in *Proc. 7th Int. Conf. Quality Software (QSIC)*, Portland, OR, USA, Oct. 2007.

[23] D. Liberzon, *Switching in Systems and Control*. Boston, MA, USA: Birkhauser, 2003.

[24] H. Ye, A. N. Michel, and L. Hou, "Stability analysis of systems with impulse effects," *IEEE Trans. Autom. Control*, vol. 43, no. 12, pp. 1719–1723, 1998.

[25] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 461–474, 1998.

[26] A. Q. Huang, M. L. Crow, G. T. Heydt, J. P. Zheng, and S. J. Dale, "The future renewable electric energy delivery and management (FREEDM) system: The energy internet," *Proc. IEEE*, vol. 99, no. 1, pp. 133–148, Jan. 2011.

[27] L. M. Ni, C.-W. Xu, and T. B. Gendreau, "A distributed drafting algorithm for load balancing," *IEEE Trans. Software Eng.*, vol. 11, pp. 1153–1161, 1985.

[28] C. Hoare, *Communicating Sequential Processes*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1985.

[29] R. Akella, F. Meng, D. Ditch, B. McMillin, and M. Crow, "Distributed power balancing for the freedm system," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2010, pp. 7–12.

[30] G. Levin and D. Gries, "A proof technique for communicating sequential processes," *Acta Inf.*, vol. 15, pp. 281–302, 1981.

[31] S. Shojaeian, J. Soltani, and G. A. Markadeh, "Damping of low frequency oscillations of multi-machine multi-UPFC power systems, based on adaptive input-output feedback linearization control," *IEEE Trans. Power Syst.*, vol. 27, pp. 1831–1840, Nov. 2012.

[32] T. Paul, J. W. Kimball, M. Zawodniok, T. P. Roth, and B. McMillin, "Invariants as a unified knowledge model for cyber-physical systems," in *Proc. IEEE Int. Conf. Service Oriented Comput. Appl., Int. Workshop Knowledge Service Technol. Life, Environ., Sustain. (KASTLES)*, 2011.

[33] K. R. Padiyar, *Power System Dynamics: Stability and Control*. Kent, U.K.: Anshan Ltd, 2004.

[34] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*. Champaign, IL, USA: Stipes, 2007.

[35] F. Cristian, H. Aghili, and R. Strong, "Clock synchronization in the presence of omission and performance failures, and processor joins," in *Proc. IEEE 16th Int. Symp. Fault-Tolerant Comput. Syst.*, 1992, pp. 218–223.

[36] P. Bolzern, P. Colaneri, and G. D. Nicolao, "On almost sure stability of continuous-time Markov jump linear systems," *Automatica*, vol. 42, pp. 983–988, 2006.

[37] P. Bolzern, P. Colaneri, and G. D. Nicolao, "Almost sure stability of Markov jump linear systems with deterministic switching," *IEEE Trans. Autom. Control*, vol. 58, pp. 209–213, Jan. 2013.

[38] C. Li, M. Z. Q. Chen, J. Lam, and X. Mao, "On exponential almost sure stability of random jump systems," *IEEE Trans. Autom. Control*, vol. 57, pp. 3064–3077, Dec. 2012.

Tamal Paul (S'09) received his B.Tech degree in electrical engineering from National Institute of Technology Durgapur, India in 2010, and the M.S. degree in electrical engineering from the Missouri University of Science and Technology (Missouri S&T), Rolla, MO, USA, in 2012 where he is currently working toward the Ph.D degree. His current research interests include analysis, design, and control of switched system stability in cyber-physical systems.

Jonathan W. Kimball (M' 96–SM' 05) received the B.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 1994, and the M.S. degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, IL, USA, in 1996 and 2007 respectively.

From 1996 to 1998, he was with Motorola, Phoenix, AZ, USA. He is currently an Assistant Professor at Missouri S&T, Rolla, MO, USA.

Dr. Kimball is a member of Eta Kappa Nu, Tau Beta Pi, and Phi Kappa Phi. He is a licensed Professional Engineer in the State of Illinois. He has served on the IEEE Power Electronics Society Administrative Committee, as a Member-at-Large and as Education Chair, since 2010 and is the current chairperson of the St. Louis Chapter of the Industry Applications Society.

Maciej J. Zawodniok (S'03, M'06) received a Ph.D. degree in Computer Engineering from the University of Missouri, Rolla, MO, USA, in 2006.

He is currently an Assistant Professor in Computer Engineering and Assistant Director of NSF I/UCRC on Intelligent Maintenance Systems at Missouri S&T. Dr. Zawodniok's research focuses on adaptive and energy-efficient protocols for wireless networks, network-centric systems, network security, cyber-physical and embedded systems with applications to manufacturing and maintenance. He received the NSF CAREER award in 2010

Thomas Roth received the B.S. degree in computer science from Missouri S&T, Rolla, MO, USA, in 2011, where he is currently working towards the Ph.D. degree. His research interests are in the detection of dishonest peers in distributed cyber-physical systems.

Bruce McMillin (SM '07) is currently a Professor of Computer Science and Director of the Center for Information Assurance at Missouri S&T, Rolla, MO, USA. He leads and participates in interdisciplinary teams in formal methods for fault tolerance and security in distributed embedded systems with an eye towards critical infrastructure protection. He is leading the distributed grid intelligence project of the Future Renewables Engineering Research Center.

Sriram Chellappan received the Ph.D. degree in Computer Science and Engineering from Ohio State University, Columbus, OH, USA, in 2007. He is an Assistant Professor of Computer Science at Missouri S&T, Rolla, MO, USA, where he directs the SCoRe (Social Computing Research) Group. His research interests are in cyber security, cyber-physical systems, mobile networking, and human centered computing. He received the NSF CAREER Award in 2013.