

Invariants as a Unified Knowledge Model for Cyber-Physical Systems

Tamal Paul, Jonathan W. Kimball and Maciej Zawodniok
Department of Electrical and
Computer Engineering
Missouri University of Science and Technology

Thomas P. Roth and Bruce McMillin
Department of Computer Science
Missouri University of Science and Technology

Abstract—Cyber-Physical Systems (CPS) consist 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 components that facilitates system synthesis, verification, and monitoring is a significant challenge of a CPS research program. In the emerging smart grid, for example, system state provides input into distributed computer algorithms that manage power and energy via local computation with messaging passing over a computer network collectively resulting in control signals to advanced power electronics. Computational correctness, network timing, and frequency response are all system aspects that conspire to impede design, verification, and monitoring. This paper seeks to unify the knowledge present in these diverse aspects through developing common semantics that span each aspect of a CPS. Specifically, a “smart grid” type system is considered. Power commands to various loads and alternative energy sources are stepped in response to cyber controllers that are networked. This paper shows the development of a physical invariant, based on the theory of Lyapunov-like functions, and a cyber invariant, the governs the correctness of a power dispatch algorithm, and couples the two to develop an overall system stability invariant. The invariant approach is tested with two scenarios. In the first case, the system is subjected to two commanded pulses beyond the stable limit, with the second perturbing pulse being of a magnitude greater than the first, which makes the system unstable. In the second case, the system is subjected to two commanded pulses beyond its stable limit but with a comparatively smaller magnitude of the perturbing second pulse, which allowed the system to remain stable. The measure of stability is an energy function which, under certain conditions, serves as a Lyapunov-like invariant that is used to prove stability.

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 search and rescue) and large scale distributed coordination (such as automated traffic control and future smart electrical grid systems) that are highly efficient (renewable resource coordination). Common to these systems are three principal functional components, a (distributed) cyber component, and networking/communications component, and an underlying physical infrastructure. Integration, correctness, and stability are significant challenges in creating the tight

conjoining of these three components.

Integration is particularly vexing 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. The approach of this paper is fundamentally different, composing correctness instead of functionality. Thus, this paper constructs CPSs through compositional integration of the cyber, networking, and physical components rooted in correctness.

In the evolving smart grid, system state provides input to distributed cyber algorithms that manage power and energy by means of local computation while passing messages over a computer network to develop control signals for advanced power electronics. The proposed work introduces a cyber-physical approach towards the monitoring and control of switched physical entities interconnected by physical infrastructures. A unifying framework for designing cyber-physical systems is desired; however, the problem lies in the absence of any comprehensive tool to do so. 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.

An invariant, essentially, is a logical predicate on a system state that does 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 [5]. In this system, program actions are related to logical truth through axioms and inference rules. Invariants are widely used ranging from algorithm instruction [6] to never claims in model checking [7]. Invariants are well-understood for cyber processes, but extending them into the network and physical domains requires some insight.

From the physical perspective, as described in [8], Lyapunov functions can be applied to describe the dynamics of the

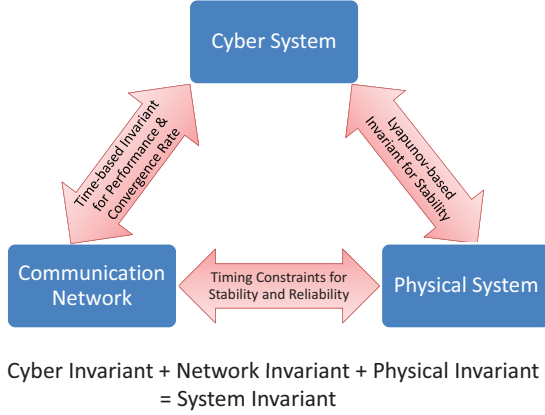


Fig. 1. Combining the physical and network Lyapounov-like functions and invariants with the cyber invariant concept yields a CPS system invariant, a semantically common method relating the cyber, physical, and network components. If the composition can be made noninterfering, the the resulting invariant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

physical system. Lyapunov functions can describe the complex behavior of a power grid [9]. However, there are no definite ways to find a Lyapunov function for nonlinear systems. In a switched system, a substitute (proxy) of a true Lyapunov function can be found by using either the norm of the state vector or the energy of the system (the latter being used in the present work). Under certain conditions, this may be termed a Lyapunov-like function [10] and can verify the stability characteristics of the system.

While not considered in this paper, Lyapunov functions have also been used to describe network stability [11]–[13]. 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.

The overarching goal of this work is to simultaneously treat all three aspects of a CPS, as depicted in Figure 1, creating a CPS through integration of non-interfering cyber, physical, and network invariants. The overarching goals of this work are to

- create invariant techniques for each fundamental component of a CPS,
- synthesize CPSs through the composition of cyber, physical, and network invariants,
- architect run-time adaptation of CPSs, through their invariants, and
- demonstrate the feasibility of the approach on a real-world testbed.

II. BACKGROUND

This section explores the idea of invariants from their origins in the cyber world for program correctness and how

they can be used within switched systems to represent stability as correctness invariants.

A. Cyber Invariants and Interference

Axiomatic proof systems for computer programs have their origins in the 1960s [5]. Key to these systems is a set of axioms and inference rules that relate program statements to logical theorems. Of particular interest are invariants, such as expressed by the following Hoare Triple for an iterative statement:

$$\frac{\{I_C\}S\{I_C\}}{\{I_C\} \text{ do } G \rightarrow S\{I_C\}} \quad (1)$$

When extended to concurrently executing sets of statements, S_1, S_2, S_n these proofs can be composed

$$\frac{\{I_{C1}\}S_1\{I_{C1}\}\{I_{C2}\}S_2\{I_{C2}\}\dots\{I_{Cn}\}S_n\{I_{Cn}\}, NI}{\{I_{C1} \wedge I_{C2} \wedge \dots \wedge I_{Cn}\}S_1//S_2//\dots//S_n\{I_{C1} \wedge I_{C2} \wedge \dots \wedge I_{Cn}\}} \quad (2)$$

NI represents the property of noninterference. Essentially, to show noninterference requires showing that for all actions a in some statement S_i and all assertions P_{jk} , $\{pre(a) \wedge P_{jk}\}a\{P_{jk}\}$ remains a theorem. 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 [14], 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. This invariant, I_C , on the correctness of the system forms the top leg of Figure 1.

B. Switched System Theory

A switched system is a fundamentally continuous-time system with changes that occur at discrete times [15]. A classic example is a bouncing ball: its dynamics are governed by gravity and Newton’s laws, and its velocity changes (instantaneously, as approximated) direction 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.

The continuous state of the system is expressed as a vector of state variables ($\mathbf{x} \in \mathbb{R}^n$) whose values describe the system at a given time, and whose time history describe the system dynamics from the initial conditions to the current time. The inputs, expressed as a vector \mathbf{u} , may be external to the system or generated with feedback of a combination of the state variables and outputs. The outputs, expressed as the vector \mathbf{y} , are the only measurable characteristics of the system. 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

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (3)$$

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 has the following properties:

- 1) $V(\mathbf{x})$ is positive definite, that is, $V(\mathbf{x}) > 0 \forall \mathbf{x} \neq 0, V(0) = 0$.
- 2) $V(\mathbf{x})$ is radially unbounded.
- 3) $\frac{dV}{dt} \leq 0$ along all trajectories ($\frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) \leq 0$).

If $\frac{dV}{dt}$ is non-positive, the system is stable. If $\frac{dV}{dt}$ is strictly negative, the system is asymptotically stable.

Unfortunately, while finding Lyapunov functions for low-order linear systems is straightforward, high-order systems pose significant computational challenges and no general techniques exist for non-linear systems. For a switched system, another class of functions may be considered, namely *Lyapunov-like* functions [16]–[18]. 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 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. This concept is shown conceptually in Figure 2.

Lyapunov-like functions are powerful tools for analysis of a complex CPS. For each operating mode of the physical subsystem, 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 becomes an invariant I_P on the stability of the system, as in the right leg of Figure 1.

III. DESCRIPTION OF THE MODEL

This paper, in particular, addresses the control strategy of a three converter (all in three phase) grid-connected system to deliver/absorb power as commanded by a cyber power dispatch algorithm from the external finite-inertia generator. The stable range of operation of the system depends on the power levels commanded and also on the dwell time between the perturbation instants.

The physical systems model was simulated using PSCAD. 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 had been designed keeping in mind the desired control strategy which is to produce the power as commanded by the cyber algorithm.

A. Switching Scheme of the Converter

The block diagram representation in Fig. 3 shows the control scheme for the converter switches in the present work. The active and reactive power transfer relationship for simulated system can be expressed in the $d-q$ reference frame [19] as

$$P = \frac{3}{2}(V_q I_q + V_d I_d) \quad (4)$$

$$Q = \frac{3}{2}(V_q I_d - V_d I_q) \quad (5)$$

In (4)-(5), V_q and I_q are the q -axis voltages and currents while V_d and I_d are the d -axis voltages and currents. The converter local currents (I_a, I_b, I_c) are converted into their respective d and q counterparts. The cyber algorithm delivers the commanded values of P and Q . Based on these desired values, P^* and Q^* , the ideal d and q axis currents for each converter can be calculated from the following set of equations.

$$I_d^* = 2(P^* V_d + Q^* V_q) \quad (6)$$

$$I_q^* = 2(P^* V_q - Q^* V_d) \quad (7)$$

In (6)-(7), P^* and Q^* are the commanded values of active and reactive powers from the cyber algorithm. For the present study the reactive power commanded is always zero.

The ideal d and q axis currents are converted back to their respective $a-b-c$ counterparts. Delta modulation is then imposed to control the converter currents to the desired $a-b-c$ current values. This generates the switching signals for the converters.

The ideal values of I_a, I_b, I_c of the three converters when then converted to their respective $d-q$ references will generate the commanded values of active and reactive power as per (4)-(5), if the system voltage and frequency are near nominal conditions. Not shown in Fig. 3 is a phase-locked loop that determines the system frequency to be used in the $abc-dq$ conversion.

B. Cyber-Physical Control Scheme

C. System Architecture

Consider the architecture of the FREEDM [20] system shown in Figure 4. FREEDM forms a microgrid of energy storage (DESD) energy resources (DRER), and LOADs to share power for the good of the entire system. FREEDM is envisioned as an architecture for future ‘‘Smart Distribution’’ systems. Intelligent flow controllers (Nodes) contain physical actuators (Solid State Transformers (SSTs)) that control power flow to and from a shared electrical bus, under direction of cooperating Distributed Grid Intelligence Processes (DGIs).

Each Node is potentially owned and located in a house or business. Within each Node, the DGI processes compute a power cost and use a drafting process [22]. Drafting is

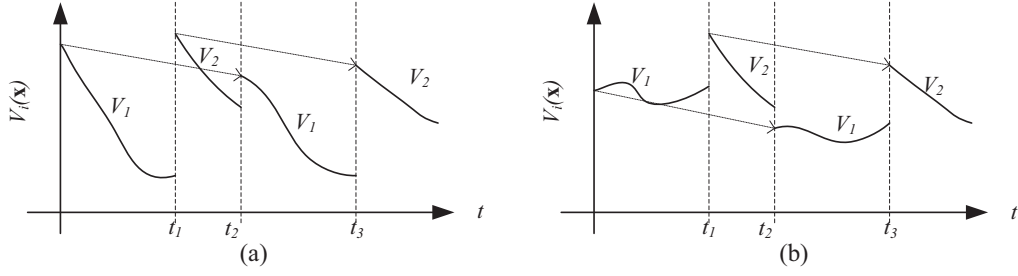


Fig. 2. Asymptotic stability using multiple Lyapunov functions (V_1 and V_2). (a) Two true Lyapunov functions. (b) One Lyapunov function (V_2), one Lyapunov-like function (V_1).

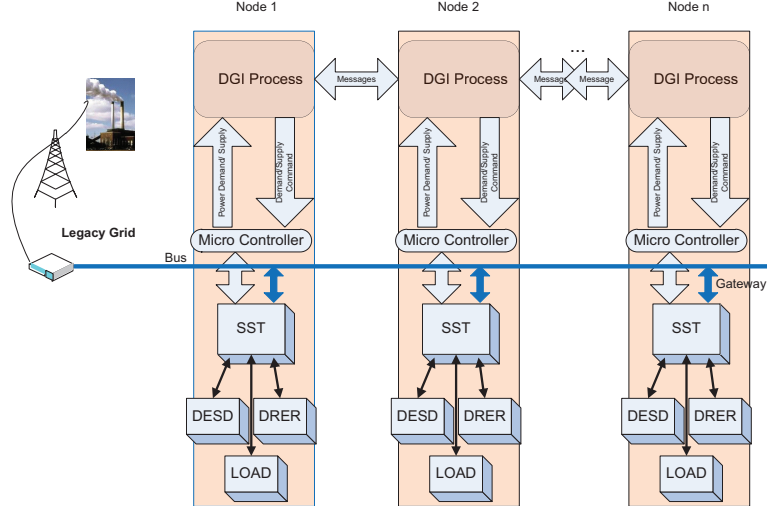


Fig. 4. FREEDM Power Management Architecture [21]

a receiver-initiated load-balancing procedure; if a Node (S) has available power generation capacity, it solicits bids from loaded Nodes (L).

D. Lyapunov and Lyapunov-Like Analysis

The following analysis studies a simple system where a pair of FREEDM nodes are communicating over a network and are both connected to a power grid. The power flow on the grid is otherwise balanced internally. The communication between the two nodes (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. During mode I, 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. The delay time and queueing may result in power perturbations.

The dynamics of the average number of messages from S to L that have been sent but not received, K , can be modeled

as

$$\frac{dK}{dt} = \lambda - \mu \quad (8)$$

where $K \geq 0$ by definition. Therefore, the net error in power (i.e., the difference between the source power and the load power) is $P_{error} = \delta K$. This net error will tend to increase the grid frequency, which is governed by a simplified swing equation,

$$\frac{d\omega}{dt} = -\frac{V_1 V_2}{J\omega X} \sin(\theta - \theta_0) - \frac{D}{J}(\omega - \omega_0) + \frac{P_{err}}{J\omega} - \frac{kP^2}{J\omega} \quad (9)$$

where D is the natural damping due to frequency-sensitive loads and J is the effective rotational inertia. If a large enough error persists, the frequency will become too high. When this condition is detected, the system must switch to mode II, in which S no longer migrates additional quanta but instead follows a droop law of the form $P_S = P_{request} - m(\omega - \omega_0)$. During mode II, $\lambda = 0$ but $\mu \geq 0$ because L may continue to process messages in its queue of received messages. After switching modes at t_{II} , the net error in the power will be proportional to the number of messages that are still outstanding K , less the droop effect, so $P_{error}(t) = \delta K - m(\omega - \omega_0)$.

The decision whether to switch from mode I to mode II may be determined based on some metric of the system state. A potential Lyapunov-like function is a quadratic in the two

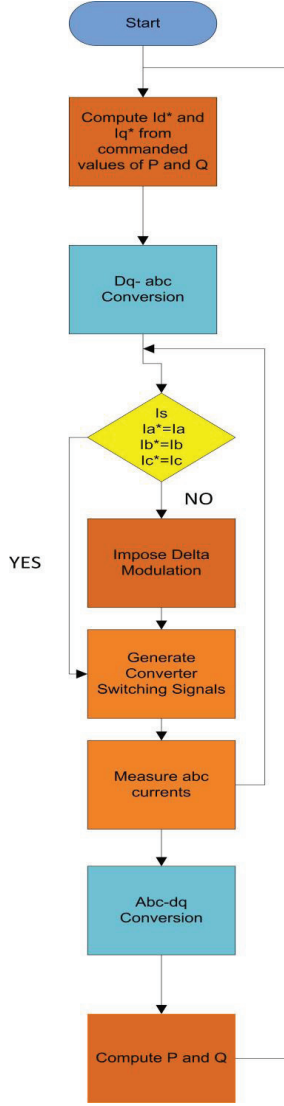


Fig. 3. Inverter switching scheme algorithm.

dynamic variables,

$$V(\omega, \theta) = \frac{J}{2}(\omega - \omega_0)^2 + \frac{V_1 V_2}{\omega X}(1 - \cos(\theta - \theta_0)). \quad (10)$$

$$\frac{dV}{dt} = J(\omega - \omega_0) \frac{d\omega}{dt} + \frac{V_1 V_2}{\omega X} (\sin(\theta - \theta_0)) \frac{d\theta}{dt} \quad (11)$$

Substituting (10) in (11)

$$\begin{aligned} \frac{dV}{dt} &= J(\omega - \omega_0) \left[-\frac{V_1 V_2}{J\omega X} (\sin(\theta - \theta_0)) - \frac{D}{J} (\omega - \omega_0) \right] \\ &= +\frac{\delta K}{J\omega} - \frac{m}{J\omega} (\omega - \omega_0) - \frac{kP^2}{J\omega} \\ &= +(\omega - \omega_0) \frac{V_1 V_2}{\omega X} \sin(\theta - \theta_0) \end{aligned}$$

Here V_1 is the generator back emf, V_2 is the bus voltage at the generator terminals, $\theta - \theta_0$ represents the torque angle, and k is power scaling factor that encapsulates the voltage and resistance and scales the gross power in terms of power

loss. This derivative must be negative for the function to be a Lyapunov-function. Therefore,

$$\{I_{P1} : (\omega - \omega_0)^2 (D\omega + m) + (\omega - \omega_0) (kP^2) > (\omega - \omega_0) (\delta K)\} \quad (12)$$

This parameterization of the droop constant gives us our invariant that guides the system within its stable restraints. For all values of m obeying the above invariant, the system must be stable. This equation can be solved further to obtain the droop constant in the droop control in terms of the system parameters:

$$m > \frac{\delta K - kP^2}{\omega - \omega_0} - D\omega \quad (13)$$

The absence of $\theta - \theta_0$ in the final invariant expression makes the proposed approach invariant independent of a generator.

E. Cyber Algorithm

The overall stability of the system is only partially governed by I_{P1} . Since the cyber algorithm PowerBalance's (Algorithm 1) commands operate on the power system electronics, its actions are asynchronous with respect to those of the power electronics. Thus, the potential exists for PowerBalance's operation to interfere with the truth of the invariant I_{P1} .

In PowerBalance each of n processes executes an algorithm triggered by the state of the underlying power system, either *high* or *low*. At all times the system should maintain the invariant $\{I_{C1} : n\nu + \sum_i^n high_i + \sum_i^n low_i = g\}$ (where ν is the nominal load per Node, $high_i$ and low_i are the amount of variance above and below the nominal load). This algorithm approximates a distributed solution to the fractional knapsack problem [21]. The fractional knapsack problem requires the invariant $\{I_{C2} : \exists_{l,m} max high_l^r - max high_m^{r+1} < 0, r = 0, \dots, k-1\}$ (where $high^r$ indicates the r^{th} decision made by the algorithm. g represents the excess draw or supply to/from a grid connection. The function $migrate(\delta, j)$ is a command to the underlying power system to provide/accept a quantum of power to/from a Node j).

The invariant I_{C2} holds at termination of the algorithm, by the greedy choice principle. Due to lack of strict synchronization between cyber processes, assignment $low_i = low_i + \delta$ potentially interferes with the truth of the invariant I_{C1} . Potentially, k migrations are outstanding at any point in time. Thus, the invariant is relaxed to I'_{C1} until the migration has been received at process P_j :

$$\{Q'_{C1}/I'_{C1} : \{n\nu + \sum_i^n high_i + k\delta + \sum_i^n low_i = g\}$$

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 [23]).

For the proposed cyber-physical system, the final invariant is the conjunction of the cyber invariant and a logical statement related to the physical system stability. The physical portion may be analyzed with a Lyapunov or Lyapunov-like function. If we are concerned only with asymptotic stability, in this case, we must remain in mode II at all times and use I_{P1} directly. (Otherwise, \dot{V} may be positive.) If *boundedness* is sufficient,

then the following invariant is appropriate:

$$\{I_P : I_{P1} \vee (V(\omega, K) < V_{bound}) \vee (V(t) < V(t_{II}))\} \quad (14)$$

where V_{bound} is the maximum allowable value of V , $V(t)$ is the value of $V(\omega, P_{error})$ at the present time and $V(t_{II})$ is its value at the most recent previous switch over to mode II.

Thus, PowerBalance interferes with power system Lyapunov function that governs the "mismatch" between the sending and receiving processes. Lyapunov invariant (12) provides a relationship between the amount of frequency error and the number of pending/dropped migrations. To ensure that the actions of the $A_1 : P_j!select$ and $A_2 : low_i = low_i + \delta$ do not interfere with I'_{C1} , $\{I'_{C1} \wedge I_P\}A_1\{I_P \wedge I_N\}$ holds as a theorem when $k = K$, thus, bounding the cyber invariant by the Lyapunov invariant.

To guarantee the CPS maintains the invariant, the system invariant is added as a guard, $I'_{C1} \wedge I_P \wedge I_N$, as a weakest precondition on the communication, resulting in the following algorithm written in a CSP-like language [24].

IV. IMPLEMENTATION

PSCAD communicates to the cyber algorithms through a C++ socket interface. Send and receive functions were written to handle the transfer of arrays with arbitrary but known lengths. These functions were called through FORTRAN, the language embedded in PSCAD. The send function allowed PSCAD to transfer merged data to an external controller coded in C++, while the receive function probed for controller responses that could be imported back into the PSCAD environment.

Two PSCAD components were designed to encapsulate the C socket code: `pscad_send` and `pscad_recv`. The `pscad_send` component took an n -dimension line as an input which was created through combining multiple data signals. Its parameters specified the IP address of the external cyber control and a time offset R_d of the delay between each transmission. During runtime, `pscad_send` transmitted the current state of its associated signals to the given address once every R_d simulation seconds.

The `pscad_recv` component provided an n -dimension line as an output which was data tapped to access individual signals. Its output contained the most recent set of signals obtained from an external program tasked with controlling the simulation. During runtime, `pscad_recv` updates its output line with the most recent set of received data. Fig. 5 shows the actual implementation of these components in the PSCAD environment, where the input of `pscad_send` corresponds to $x(t)$ and the output of `pscad_recv` corresponds to $x(t+1)$.

The basic experimental setup consisted of manipulated traces of the PowerBalance C++ code resulting from sequences of inputs $x(t)$ producing an output $x(t+1)$ as a vector of power migrations. PSCAD provided the input as the most recent snapshot of power settings for the three converters, specified as an array in the form $x(t) = \{t, P1(t), Q1(t), P2(t), Q2(t), P3(t), Q3(t)\}$. These are parameterized by t (time) representing the latency involved in

Algorithm 1: PowerBalance Cyber Algorithm

PowerBalance P_i ::

```

var  $k = 0$ 
do
   $\{I'_{C1} : n\nu + \sum_l^n high_l + k\delta \sum_l^n low_l = g\}$ 
  // Power Flow Invariant

   $\{I_{C2} : \exists_{l,m} max high_l^r - max high_m^{r+1} < 0, r =$ 
   $0, \dots, k-1\}$  // Knapsack
  // Invariant

   $status = input()$ 
   $status = low \rightarrow broadcast\_request$ 
  do
     $\forall l = 1, \dots, n$  // Receive responses
    from any processes

     $P_l?response[l]$ 
     $\square P_{l+1}?response[l+1]; \square \dots$ 

     $sort(response), highestcorrespondingtoP[j]$ 
    /* Guard the migration by the
    developed Invariant */
   $I'_{C1} \wedge I_P \wedge I_N \rightarrow$  do
     $P_j!select;$  // Send the Winning
    Migration

     $low_i = low_i + \delta$ 
     $migrate(\delta, j)$ 

    // Command the local device to
    transfer Power to  $j$ 

     $\{Q'_{C1} : \{n\nu + \sum_l^n high_l + k\delta + \sum_l^n low_l = g\}$ 
     $\square status = high \wedge P_j?request \rightarrow P_j!response$ 
     $\square status = high \wedge P_j?select \rightarrow$  // Receive
    the Winning Migration

  do
     $migrate(\delta, j)$  // Command the local
    device to receive Power from  $j$ 

     $high_i = high_i - \delta$ 

   $k = k + 1$ 
   $\{I_{C1}, I_{C2}\}$ 

```

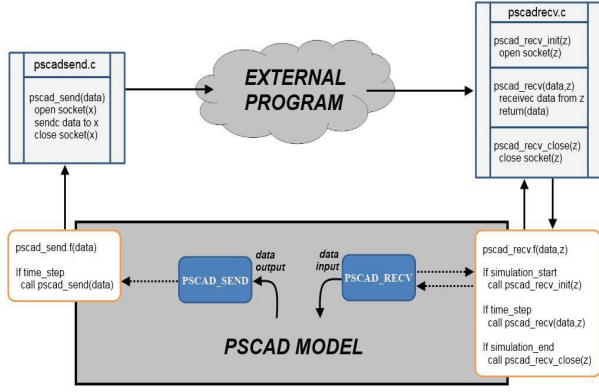


Fig. 5. Architecture of the PSCAD model communicating with the external environment.

Time	SST 1	SST 2	SST 3
0.0	0	0	0
0.8	10	0	0
0.9	10	-20	0
1.0	20	0	-20

Tab. I

SWITCHED CYBER SYSTEM STATES RESULTING IN UNBALANCED POWER MIGRATIONS.

the powerBalance algorithm computation. The migrate power commands for the converters were dispatched to PSCAD in the form $x(t+1) = \{P1C, Q1C, P2C, Q2C, P3C, Q3C\}$.

For the experiments reported in this paper, the external controller mapped simulation time to a set of power changes in the form $m(t) = \{1, dP1, dQ1, dP2, dQ2, dP3, dQ3\}$.

V. SIMULATIONS AND RESULTS

All simulations have been carried out in PSCAD. The power pulses were commanded by the cyber control at each converter end to emulate the power migration discussed in Section III-D.

The first case shows when the system is forced into an unusual state (one that does not satisfy the cyber invariant I'_{C1} through a series of switched system states that result in unbalanced power migrations. In Table I, the system is forced to remain in Mode I (only cyber commands control the system) while contents of each row at 0.8 s and 0.9 s do not sum to zero causing a variation in the frequency domain (Fig. 6. Even at time 1.0 s, when I'_{C1} becomes satisfied (by guarding the migration to reduce excess power transfers), the combined system invariant $I'_{C1} \wedge I_P \wedge I_N$ is still invalidated. The reason is that the proposed Lyapunov-like function is not strong enough to *recover* from frequency instability without switching to a different system mode. The error in energy for the system is very important since it becomes the Lyapunov-like function and hence our tool to ascertain stability.

The Lyapunov-like function that describes this behavior is derived from the error in the system energy (15):

$$V = \Delta E = \frac{1}{2} J(\omega - \omega_0)^2 \quad (15)$$

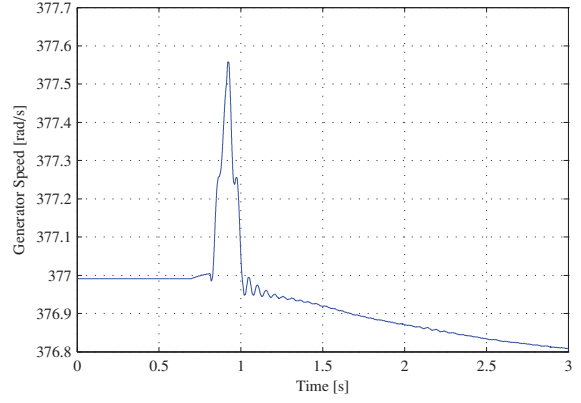


Fig. 6. Frequency variations in switched system response with only cyber correction

where J is the inertia constant of the system, ω is the angular frequency at which the system operates and ω_0 is the nominal frequency of 60 Hz at steady state level.

Fig. 7 shows another scenario where there are perturbations at time instants 0.2 s and 0.3 s. The first perturbation commanded a power pulse of 30 MW which was beyond the stable operating level. The second perturbation was of a much smaller magnitude, 12 MW, which also was above the stable operating range. The lower graph is a plot of the energy function V , which is proposed as a Lyapunov-like function. Whenever the power is commanded beyond the normal operating limits, the system frequency will also shift and will no longer remain at the nominal operating point of 60 Hz. Hence as per (15), deviation of the frequency from the nominal level will cause an ensuing increase in the error of the energy function, which indicates potential instability. Thus as can be seen from Fig. 7, the energy function rises at those two specific instants when the system is disturbed beyond its normal range of operation but eventually goes back to zero, indicating that the system had regained stability. Since the value of V at successive switching instants decreases, V is a Lyapunov-like function and the system is stable.

The next scenario is illustrated by Fig. 8. The perturbation instants are the same as in the previous case and so is the magnitude of the first perturbation. However in this case the second perturbation is much higher (60 MW). Such a high disturbance does not allow the system to regain its stability which had been corroborated by the energy function plot of Fig. 8. Since the value of V at switching instants is not decreasing, the system may be unstable, and in this case indeed diverges from the desired operating point.

The heavy red lines in Figs. 7-8 indicate the increase or decrease in the energy function at subsequent switching instants.

VI. CONCLUSION AND FUTURE WORK

This paper describes a method to unify the knowledge present in the diverse areas of power, computing, and networking. The key to bridging these areas is through a unified treatment of invariants and the noninterference of actions.

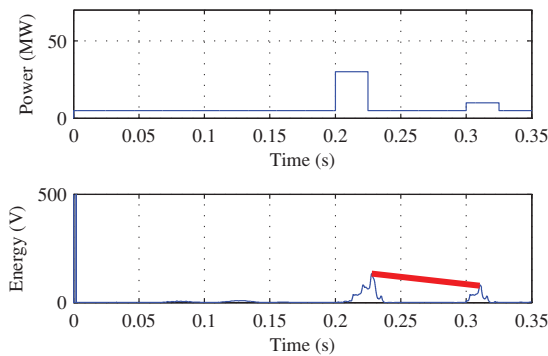


Fig. 7. Simulated microgrid performance in response to commanded power pulse where system remains stable.

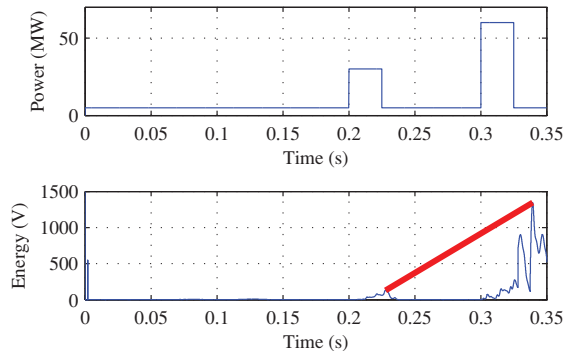


Fig. 8. Simulated microgrid performance in response to commanded power pulse where system is unstable.

Invariants are a natural fit to explain cyber systems. Lyapunov-like functions provide invariants for physical and network systems and noninterference is the glue that ties all three together.

While invariants are a naturally occurring artifact of formal cyber system specification, with Lyapunov functions, there are no generic tools that would enable system designers to find a Lyapunov-like function (V) for a given non linear switched system. The proposed approach used the energy of the error as the Lyapunov-like function for the given system. To extend the concept, the dynamics of V need to be estimated over short time scales to determine how much V changes between mode changes. Singular perturbation methods [25] enable the separation of the dynamics into “slow” and “fast” time scales to simplify the analysis.

Correction of the system if the invariant becomes invalidated is another issue. Droop is one corrective action, message backlog reduction, or stopping the power migration from one or more SSTs. can improve the system stability. Further investigation is required to explore more integrated cyber-physical-network corrective actions. The key aspect is that any proposed correction must not interfere with the correctness of the system invariant.

ACKNOWLEDGMENT

The authors would like to thank the Intelligent Systems Center and the Energy Research and Development Center at

Missouri S&T for their support of this project.

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 *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '10. New York, NY, USA: ACM, 2010, pp. 1–11. [Online]. Available: <http://doi.acm.org/10.1145/1795194.1795196>
- [2] M. Sintzoff and F. Geurts, “Analysis of dynamical systems using predicate transformers - attraction and composition,” in *Analysis of Dynamical and Cognitive Systems*, 1993, pp. 227–260.
- [3] “Incremental invariant generation for compositional design,” Verimag Research Report, Tech. Rep. TR-2010-6, 2010.
- [4] S. Owicki and D. Gries, “An axiomatic proof technique for parallel programs,” *Acta Informatica*, vol. 6, pp. 319–340, 1976.
- [5] C. A. R. Hoare, “An axiomatic basis for computer programming,” *Communications of the ACM*, vol. 12, no. 10, pp. 576–585, October 1969.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.
- [7] *SPIN homepage* <http://spinroot.com/spin/whatispin.html>.
- [8] C. Chen, *Linear System Theory and Design*. New York: Holt, Rinehart and Winston, 1984.
- [9] M. Roozbehani, M. Dahleh, and S. Mitter, “Robust and distributed decisions for future cyber-physical energy networks,” June 2009.
- [10] M. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.
- [11] L. Massouli, “Structural properties of proportional fairness: Stability and insensitivity,” *Ann. Appl. Probab.*, vol. 17, no. 3, pp. 809–839, 2007.
- [12] M. Zawodniok and S. Jagannathan, “Predictive congestion control protocol for wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 6, no. 11, pp. 3955–3963, 2007.
- [13] S. Jagannathan, *Wireless ad hoc and sensor networks: protocols, performance, and control*, 2007.
- [14] Y. Sun, B. McMillin, X. F. Liu, and D. Cape, “Verifying Noninterference in a Cyber-Physical System: The Advanced Electric Power Grid,” in *Proceedings of the Seventh International Conference on Quality Software (QSIC)*, Portland, OR, October 2007.
- [15] D. Liberzon, *Switching in Systems and Control*. Boston: Birkhauser, 2003.
- [16] H. Ye, A. N. Michel, and L. Hou, “Stability analysis of systems with impulse effects,” *IEEE Transactions on Automatic Control*, vol. 43, no. 12, pp. 1719–1723, 1998.
- [17] —, “Stability theory for hybrid dynamical systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 461–474, 1998.
- [18] M. S. Branicky, “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.
- [19] H. Akagi, Y. Kanazawa, and A. Nabae, “Instantaneous reactive power compensators comprising switching devices without energy storage components,” *IEEE Transactions on Industry Applications*, vol. IA-20, pp. 625–630, May/June 1984.
- [20] 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,” *Proceedings of the IEEE*, vol. 99, no. 1, pp. 133–148, Jan. 2011.
- [21] R. Akella, F. Meng, D. Ditch, B. McMillin, and M. Crow, “Distributed power balancing for the FREEDM system,” in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, October 2010, pp. 7–12.
- [22] L. M. Ni, C.-W. Xu, and T. B. Gendreau, “A Distributed Drafting Algorithm for Load Balancing,” *IEEE Transactions on Software Engineering*, vol. 11, pp. 1153–1161, 1985.
- [23] G. Levin and D. Gries, “A proof technique for communicating sequential processes,” *Acta Inf.*, vol. 15, pp. 281–302, 1981.
- [24] C. Hoare, *Communicating Sequential Processes*. Prentice Hall, 1985.
- [25] J. Kimball and P. Krein, “Singular perturbation theory for dc-dc converters and application to pfc converters,” *IEEE Transactions on Power Electronics*, vol. 23, pp. 2970–2981, November 2008.