Mitigating Communication Delays in Remotely Connected Hardware-in-the-Loop Experiments

James L. Cale ☉, Senior Member, IEEE, Brian B. Johnson, Member, IEEE, Emiliano Dall’Anese ☉, Member, IEEE, Peter M. Young ☉, Senior Member, IEEE, Gerald Duggan, Poorva A. Bedge ☉, Student Member, IEEE, Daniel Zimmerle, Member, IEEE, and Leah Holton

Abstract—This paper introduces a potential approach for mitigating the effects of communication delays between multiple, closed-loop hardware-in-the-loop experiments which are virtually connected, yet physically separated. The approach consists of an analytical procedure for the compensation of communication delays, along with the supporting computational and communication infrastructure. The control design leverages tools for the design of observers for the compensation of measurement errors in systems with time-varying delays. The proposed methodology is validated through computer simulation and hardware experimentation connecting hardware-in-the-loop experiments conducted between laboratories separated by a distance of over 100 km.

Index Terms—Delay compensation, drift observability, hardware-in-the-loop (HIL).

I. INTRODUCTION

S
cystem emulation using hardware-in-the-loop (HIL) is a frequently used research and validation tool because it enables dynamic cosimulation of models with physical hardware in real time, see [1]–[7]. Several national laboratories, universities, and industrial companies are now pursuing approaches in forming virtually connected simulation testbeds, whereby individual HIL experiments (representing subsystems) at distant locations share state information through communication links to emulate larger, virtually connected systems. Motivation for this research is driven largely by the possibility of sharing experimental resources across large geographical distances without the need for physical relocation of (potentially scarce, sensitive, or massive) research equipment.

Previous research has been performed in connecting geographically separated testbeds in real time, with broad applications. For instance, in [8], a power system for a hybrid electric vehicle using a driver-in-the-loop motion simulator located in Michigan was connected in real time to a hybrid power train in California. To support naval power systems research, the work in [9] used (software-only) simulation of a shipboard power system where components were geographically dispersed and the simulation communicated between laboratories in Mississippi and South Carolina.

To evaluate distributed energy penetration on the U.S. electric grid, a joint experiment analyzing high penetration photovoltaics (PV) on an electrical distribution feeder was performed between two U.S. national laboratories using remote HIL, wherein a large-scale grid simulation (software-only) at a laboratory in Washington was virtually connected with a set of physical, residential PV inverters operating at another laboratory in Colorado [10].

More recently, real-time (software only) simulation of two large ac power systems connected through an HVdc link was performed between laboratories in Norway and Germany [11] and between Germany and South Carolina [12]. Researchers at a national laboratory in Australia performed a closed-loop, remote HIL with a national laboratory in Colorado, where a power network simulator and a physical PV+battery inverter system in the U.S. were virtually connected to a physical PV inverter in Australia to demonstrate coordinated solar PV firming on electrical distribution feeders [13].

However, previous research has not adequately addressed strategies for mitigating communication latencies specific to remotely connected HIL experimentation. As the complexity, number, and physical distances between remotely connected HIL systems increase, it is expected that communication delays will begin to adversely impact the time resolutions possible in multisystem(location) HIL experiments. In particular, communication sampling times between remotely located processors place a fundamental limitation on the effective bandwidth of the combined experiment due to the Nyquist frequency cutoff. Moreover, communication latencies from network traffic and security firewalls at potential host locations can be a significant
source of delay. Advanced methods for mitigating communication latencies are needed to enable larger and more complex virtually connected testbeds.

This paper describes a potential control methodology, computational and communications architecture for helping to mitigate the effects of communication delays between remote HIL experiments, and validates the method on an example system through simulation and hardware demonstration.

Key contributions of this paper are: 1) introducing concepts from the control of drift-observable, time-delayed systems as a potential framework and approach for mitigating communication delays in remotely connected HIL systems; and 2) demonstrating the approach in simulation and hardware.

The remainder of this paper is as follows. Section II provides a brief background on HIL and previous research in relevant delay compensation methods, with particular emphasis on observers for time-varying delay systems. This is followed in Section III by a description of the design approach introduced in this paper. The approach is demonstrated on an example system in Section IV. Section V demonstrates a validation of the concept in hardware. This paper concludes with a summary of findings and suggestions for future research in Section VI.

II. BACKGROUND

A. Brief Overview of HIL Simulation

HIL platforms enable realistic evaluation of physical hardware in the context of a modeled system. Using real-time simulators, models are executed and exchange signals with a physical device under test (DUT) in a closed-loop fashion; an example configuration is depicted in Fig. 1.

In Fig. 1, the supply simulator is a controllable power supply, which is physically connected to the DUT. The real-time simulator computes the response of the simulated circuit (bubbled area in Fig. 1) and provides control signals to the supply simulator. Measurements from the experiment are fed back to the circuit model on the real-time simulator, thus “closing the loop.” In this way, the dynamic response of the DUT inside the model is simulated, i.e., the DUT appears to be inside the circuit model (depicted by gray box in Fig. 1).

Single location real-time HIL, including controller HIL and power HIL, have been developed extensively for closed-loop simulations. Examples include the simulation of physical controllers and power devices [1], [2] and for investigating demand-side management techniques for providing grid ancillary services [3]. These simulations have included multiphysics domains as well, e.g., see [4] and [5].

Closed-loop remote HIL consists of two or more of the individual HIL systems depicted in Fig. 1, which are physically separated, yet exchange state information through communication links to provide a virtually connected, closed-loop simulation. However, communication latencies between HIL instances are a critical factor in ensuring that the desired bandwidth of the emulated experiment is met. The following sections describe prior research in delay compensation techniques for time-varying delay control systems, which, as will be proposed, provides a basis for a method to mitigate communication latencies in remotely connected HIL systems.

B. Delay Compensation Methods in HIL Simulation

Prior research has been performed on delay compensation for single location HIL simulations. In [14], current filters were incorporated in the feedback path to improve the stability of the closed-loop software ↔ hardware simulation. Additional approaches to prevent instabilities of the resultant closed-loop system were proposed in [15]. However, the approaches in [14]–[23] focused on single HIL testbeds, and did not address the issue of connecting remote systems through HIL.

The real-time simulation platform developed in [24] was utilized to emulate the operation of cyber-physical energy systems, where control signals for energy resources were dispatched and exchanged through a communication network. The platform combined real-time simulations of dynamic electric models and communication systems, with the goals of assessing the impact of noninstantaneous communications on distributed control tasks and evaluating reconfiguration strategies for the communication systems. Communication protocols were emulated within the HIL platform, but remote communication between HIL instances was not considered.

Communication and processing delays in wide area control systems were evaluated in [25]. Communication delays were estimated from empirical measurements from a transmission system; operational delays were then estimated and assessed via (single location) HIL.

The work in [26] considered remote access to HIL simulators to control a dynamical system. The framework was demonstrated for remote control of a robotic arm. The main objective was to provide control and simulation capabilities without physical existence of the equipment in the laboratory. The framework was close in spirit to the method presented in this paper; however, the work in [26] did not consider utilizing HIL for testing and emulating remote HIL systems or assess the impact of communication delays on the control and emulation tasks.

Focusing on general (non)linear dynamical systems, an approach for the construction of a state observer for nonlinear systems when the output measurements have nonnegligible time
C. Observers for Time-Varying Delay Systems

This section provides a condensed summary of key research findings on observer design for systems with time-varying delays described in [27]–[30], which will be referred to in subsequent sections.

Notation: Upper-case (lower-case) boldface letters will be used for matrices (column vectors); $(\cdot)^T$ denotes transposition; $| \cdot |$ denotes the absolute value of a number or cardinality of a set; $\nabla$ stands for the gradient operator; $\langle x , y \rangle$ denotes the inner product of vectors $x$ and $y$; $|x|^2 := \sqrt{x^T x}$. $C^\infty$ denotes the field of infinitely differentiable functions.

Consider the following dynamical system describing the evolution of a state vector $x(t) \in \mathbb{R}^n$:

\[
\dot{x}(t) = f(x(t)) + g(x(t), u(t)) \tag{1a}
\]

\[
\dot{y}(t) = h(x(t) - \delta(t)) \tag{1b}
\]

where $u(t) \in \mathbb{R}^m$ is a known input vector, functions $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and $h : \mathbb{R}^n \to \mathbb{R}$ are in $C^\infty$, and $y(t)$ represents a measurement of the delayed state vector $x$ at time $t - \delta(t)$. Particularly, $\delta(t) \in [0, \Delta], \forall t$ represents a known, time-varying and bounded measurement delay. The objective of [27]–[30] is to design an observer to predict $\hat{x}(t)$ by processing the delayed measurement $\hat{y}(t)$.

Let $\xi : \mathbb{R}^n \to \mathbb{R}$ be an infinitely differentiable function, and consider a vector field $\nu : S \to \mathbb{R}^n$, $S \subseteq \mathbb{R}^n$. Suppose that $\nu(x) \in C^\infty$. The Lie derivative of the function $\xi(x)$ along the vector field $\nu$ is defined as follows:

\[
L_\nu \xi(x) := \langle \nabla \xi(x), \nu(x) \rangle = \sum_{i=1}^n \frac{\partial \xi(x)}{\partial x_i} \nu_i(x). \tag{2}
\]

The $k$th Lie derivative of $\xi(x)$, denoted $L^k_\nu \xi(x)$, is obtained by $k$-times repeated iteration of $L_\nu \xi(x)$ and $L^0_\nu \xi(x) := \xi(x)$.

Consider the following mapping associated with functions $f(\cdot)$ and $h(\cdot)$ in (1):

\[
\Phi(x) := [h(x) \ L_f h(x) \ \cdots \ L_f^{n-1} h(x)]^T. \tag{3}
\]

System (1) is said to be globally drift-observable if $\Phi(x)$ is a diffeomorphism on $\mathbb{R}^n$. Drift-observability of (1) implies that the Jacobian $J(x)$ associated with $\Phi(x)$ is nonsingular for all $x \in \mathbb{R}^n$, in which case the mapping $z = \Phi(x)$ defines a global change of coordinates.

Suppose that system (1) is globally drift-observable, and has the following additional properties: (P1) The triple $(f, g, h)$ has uniform observation degree at least equal to $n$ defined as follows:

\[
L_g L_f^k h(x) = 0, \quad k = 0, \ldots, n-2, \forall x \in \mathbb{R}^n \tag{4}
\]

\[
L_g L_f^{n-1} h(x) \neq 0, \quad \text{for some } x \in \mathbb{R}^n \tag{5}
\]

in which case, the following function is well-defined

\[
p(z, u) = (L_f^k h(x) + L_g L_f^{n-1} h(x) u)_{x=\Phi^{-1}(z)}. \tag{6}
\]

(P2) Function $p(z, u)$ in (6) is globally uniformly Lipschitz continuous with respect to $z$, and the Lipschitz coefficient $\gamma(||u||)$ is a nondecreasing function of $||u||$; i.e., for any $z_1, z_2 \in \mathbb{R}^n$, it holds that

\[
||p(z_1, u) - p(z_2, u)|| \leq \gamma(||u||)||z_1 - z_2||. \tag{7}
\]

If (1) is globally uniformly Lipschitz drift-observable (GULDO) with properties (P1) and (P2), then the following observer associated with system (1) can be constructed

\[
\dot{\hat{x}}(t) = f(\hat{x}(t)) + g(\hat{x}(t), u(t)) + J^{-1}(\hat{x}(t)) k_\delta [\tilde{y}(t) - h(\hat{x}(t) - \delta(t))], \quad t \geq 0 \tag{8a}
\]

\[
k_\delta = e^{-\rho \delta} k_0 \tag{8b}
\]

where vector $k_0 \in \mathbb{R}^n$ and scalar $\rho \geq 0$ are design parameters. Theorem 1 in [28] asserts that if input $||u(t)|| \leq u_M$ for some constant $u_M$, then for a decay rate $\rho \geq 0$ and bounded delay $\delta(t) \in [0, \Delta]$ there exists a vector $k_0 \in \mathbb{R}^n$ such that

\[
||x(t) - \hat{x}(t)|| \leq ce^{-\rho t}, \quad t \geq 0 \tag{9}
\]

for some constant $c$, i.e., $\hat{x}(t)$ asymptotically converges to the ideal (nondelayed) system state $x(t)$. Furthermore, the theorem states that if vector $k_0$ satisfies the matrix inequality

\[
(A_n - k_0 C_n)^T P + P (A_n - k_0 C_n) + (2 \rho + \beta + \kappa) P + \gamma_M^2 (B_m^T P B_m) I_{n \times n} \leq 0 \tag{10}
\]

where $(A_n, B_n, C_n)$ are a Brunovski triple of order $n$, $\gamma_M$ is the Lipschitz coefficient associated with $u_M$, $\beta > 0$ and $\kappa > 1$ design parameters, and $P$ is symmetric positive definite, then exponential-decay state tracking [cf., (9)] is guaranteed for

\[
\Delta \leq \Delta := \frac{\beta}{2 + (k_0^T P k_0) \left[ P^{-1} \left[ I + \rho^2 (C_n^T C_n) \right] \right] \kappa}. \tag{11}
\]

Remark: it is important to note that the theory derived in [27]–[30] and summarized in (1)–(11) assumes control of a single system. While we use the structure of the observer design...
in (8) as a guide for the design of observers for remote HIL applications (with justification explained in the next section), convergence for multiple HIL systems must be verified in simulation until the convergence guarantee in (11) is extended and rigorously proven for multiple HIL instances; this an objective of future research.

III. EMBEDDED OBSERVERS FOR REMOTE HIL APPLICATIONS

This section explains how the theory in the previous section can be adapted to help mitigate communication delays in remote HIL applications, hereafter referred to as the observer delay compensation (ODC) approach. The central concept is depicted in Fig. 2. As shown in Fig. 2, the mathematical description of System A(B) is a function of both locally obtained (measured or computed) state information $x_a(b)$ and estimates of remote states $\hat{x}_b(a)$. Estimates for remote states occurring at System A(B) are computed in observers located at System B(A). In control theory, observers are commonly used to estimate actual state variables based on known information, such as measurements [31]. The notation $O_{x \rightarrow y}$ denotes an observer at System X that estimates delayed states received from System Y; $\delta_{x \rightarrow y}$ denotes the communication delay between System X and System Y.

In Fig. 2, delayed measurements from the remote system(s) along with knowledge of its mathematical representation are used to provide an estimate of the nondelayed remote states, through the use of embedded observers. Errors in the predicted states are then dynamically corrected as new information is obtained from the remote system(s). For simplicity, we assume herein that locally measured signals do not contain delay; only communication between remote systems is considered.

Lacking a formal convergence guarantee for multiple HIL instances corresponding to [27]–[30] at the time of the experimental work, the observer designs were performed quasi-heuristically with (1)–(8) as a guide. In particular, we assumed that if: 1) Systems A and B in Fig. 2 could both be shown to be GULDO and possess properties (P1) and (P2); 2) observers for the system were constructed using (8) (which is always possible if the first condition is satisfied); and 3) the upper bound on the communication delay between the two systems could be estimated with reasonable certainty. Then, design parameter $k_0$ in (8) may exist such that all states in the complete system converged asymptotically, which we could validate through computer simulation before hardware implementation.

The assumptions above were based on the following reasoning. When designing observers to estimate the state vectors $x \in \mathbb{R}^N, N \leq n$ at each location for an $n$-dimensional total system, drift-observability of each subsystem ensures the Jacobian $J(x)$ in (8) is nonsingular for all $x \in \mathbb{R}^N$, the guarantee of observation degree $\geq n$ ensures the subsystems are observable, and global uniform Lipschitz continuity ensures adjustment (i.e., feedback control) of states occurs along smooth and well-defined trajectories.

Guided by this reasoning, the design approach for constructing embedded observers to implement the ODC method consists of the following steps.

1) Derivation of the mathematical description of the virtually connected subsystems.
2) Check of GULDO and properties (P1) and (P2) for each subsystem.
3) Construction of observers using (8) and selection of $k_0$ for each location.
4) Validation of system convergence (e.g., in simulation).

What follows is a demonstration of the above steps on an example system partitioned into two subsystems.

IV. DESIGN EXAMPLE

A. Mathematical Description

Consider the electrical circuit shown in Fig. 3 containing two current sources $i_{1(2)}$ in parallel with shunt capacitors $C_{1(2)}$, and connected through a series $RL$ impedance.

The equation describing each of the $k \in \{1, 2\}$ sources is

$$\frac{di_k}{dt} = \eta_k (v_k - e_k) + \alpha_k i_k$$

(12)
where it is assumed that the sources represent composite current supplies which include externally connected voltage sources downstream; this is introduced to simulate system inputs to the simulation. Here, $e_i^k$ represents connected (and potentially time-varying) voltage input signals, $\eta_k$ are voltage error gains, and $\alpha_k < 0$ are damping factors.

This system can be represented in state-space as

$$\frac{dx}{dt} = Ax + Bu$$  \hspace{1cm} (13)

where the vector of states $x = [i_1 \ i_2 \ v_1 \ v_2]^T$

$$A = \begin{bmatrix} \alpha_1 & 0 & \eta_1 & 0 & 0 \\ 0 & \alpha_2 & 0 & \eta_2 & 0 \\ 1/C_1 & 0 & 0 & -1/C_1 & 0 \\ 0 & 1/C_2 & 0 & 0 & 1/C_2 \\ 0 & 0 & 1/L & -1/L & -R/L \end{bmatrix}, \quad B = \begin{bmatrix} -\eta_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$  \hspace{1cm} (14)

and the input vector $u = [e_1^k \ e_2^k]^T$.

Suppose we wish to partition the system in Fig. 3 so portions can be simulated remotely, yet remain coupled through mutual exchange of state information. First, consider a partitioning of the system into two (sub)systems, referred to herein as “System A” and “System B,” as shown in Fig. 4.

Next, consider the physical separation and control of Systems A and B where each is controlled in the following manner. First, replace System B in Fig. 4 with a controllable current source that injects estimated input current $\hat{i}_2$ into System A, as shown in the left of Fig. 5. Then, replace System A in Fig. 4 with a controllable voltage source that provides estimated voltage $\hat{v}_2$ to System B as shown on right of Fig. 5.

System A in Fig. 5 can be described in state-space as

$$\frac{dx_a}{dt} = A_a x_a + B_a u_a$$  \hspace{1cm} (15)

where the vector of states $x_a = [i_1 \ v_1 \ i \ v_2]^T$

$$A_a = \begin{bmatrix} \alpha_1 & \eta_i & 0 & 0 \\ 0 & 1/C_1 & -1/C_1 & 0 \\ 0 & 0 & 1/L & -R/L \end{bmatrix}, \quad B_a = \begin{bmatrix} -\eta_i \\ 0 \\ 0 \\ 0 \end{bmatrix}$$  \hspace{1cm} (16)

and the input vector $u_a = [e_1^\ast \ i_2^\ast]^T$. The nominal output of System A at System A is $v_2(t)$, the delayed output $\bar{y}_a(t) = v_2(t - \delta(t))$ in (1), is obtained by defining matrix $C_a$ as

$$\bar{y}_a(t) = C_a x_a(t - \delta(t)) = [0 \ 0 \ 0 \ 1] x_a(t - \delta(t)) = v_2(t - \delta(t)).$$  \hspace{1cm} (17)

For System B, the state-space representation is

$$\frac{dx_b}{dt} = A_b x_b + B_b u_b$$  \hspace{1cm} (18)

where the state vector $x_b = [i_2]$  

$$A_b = \begin{bmatrix} \alpha_2 \\ 0 \\ 0 \end{bmatrix}, \quad B_b = \begin{bmatrix} -\eta_2 \ \eta_2 \ \eta_2 \end{bmatrix}$$  \hspace{1cm} (19)

and the input vector $u_b = [e_2^\ast \ \bar{i}_2^\ast]^T$. The nominal output of System B at System B is $i_2(t)$, so matrix $C_b$ is defined as

$$y_b(t) = C_b x_b(t - \delta(t)) = [1][i_2(t - \delta(t))] = i_2(t - \delta(t)).$$  \hspace{1cm} (20)

For System A note from comparison of (1) and (15)–(17)

$$f(x(t)) \rightarrow A_a x_a(t)$$  \hspace{1cm} (21)

$$g(x(t))u(t) \rightarrow [-\eta_i e_i^k(t) \ 0 \ 0] \ \ \ \ (22)$$

$$h(x(t) - \delta(t)) \rightarrow C_a x_a(t - \delta(t)) = v_2(t - \delta(t)).$$  \hspace{1cm} (23)

The mapping $z_b = \Phi(x_a)$ for System A, with $n = 4$ is

$$\Phi(x_a) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1/C_2 & 0 \\ 0 & 1/LC_2 & -R/LC_2 & -1/LC_2 \\ \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 \end{bmatrix} x_a = A_a x_a$$  \hspace{1cm} (24)

where $\sigma_1 = 1/LC_1 C_2$, $\sigma_2 = 1/LC_1 C_2 - R/LC_2$, $\sigma_3 = -R^2/L^2C_2 - 1/LC_2^2$, and $\sigma_4 = R/L^2C_2$. Note that the mapping $\Phi(x_a)$ is diffeomorphic if $A_a$ defined in (24) is invertible.
which is determined by the values of the circuit parameters. From (24)

\[
J_a(x_a) = \begin{bmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 1/C_2 & 0 \\
0 & 1/LC_2 & -R/LC_2 & -1/LC_2 \\
\sigma_1 & \sigma_2 & \sigma_3 & \sigma_4
\end{bmatrix} = \Lambda_a
\]

(25)

which is indeed nonzero for all \(x_a \in \mathbb{R}^4\). For property (P1), it is easy to verify that \(L_gL_f^\dagger h(x) = 0\) for \(k = 0, \ldots, n - 2, \forall x \in \mathbb{R}^n\). The second condition in (P1) holds since \(L_gL_f^\dagger h(x_a) = \Gamma g(x(t))u(t)\) where \(\Gamma := [\varsigma_1 \varsigma_2 \varsigma_4], \varsigma_1 = 1/LC_1C_2, \varsigma_2 = -R/LC_2C_2, \varsigma_4 = -1/LC_1C_2 + R^2/LC_2C_2 - 1/LC_2C_2\). Therefore, \(L_gL_f^\dagger h(x_a) = -\eta_1 e_1^\dagger(t) \neq 0\) for some \(x_a \in \mathbb{R}^4\), in particular whenever the signal \(e_1^\dagger(t) \neq 0\). As expected, \(p(z_a, u_a)\) is well-defined

\[
p(z_a, u_a) = \left( L_f^\dagger h(x_a) + L_gL_f^\dagger h(x_a)u_a \right)_{x_a = \Phi^{-1}(x_a)} = \Gamma x_a - \eta_1(\sigma_1^2 - \epsilon_1^2(t))^2
\]

\[
= \Gamma \Lambda_a^{-1} z_a - \eta_1(\sigma_1^2 - \epsilon_1^2(t))^2.
\]

(26)

Considering property (P2) for System A, note that \(p(z_a, u_a)\) is globally uniformly Lipschitz continuous with respect to \(z_a\); its derivative with respect to this variable is equal to (constant) \(\Gamma g_{a} \). The Lipschitz coefficient can be determined from (7) to derive that \(\gamma \geq \|\Gamma g_{a}\|\) and is therefore nondecreasing in \(\|u_a\|\).

For System B, note from comparison of (1) and (18) and (20),

\[
f(x(t)) \mapsto A_b x_b(t) = \alpha_b i_2(t)
\]

\[
g(x(t))u(t) \mapsto -\eta_2 e_2(t)
\]

\[
h(x(t) - \delta(t)) \mapsto C_b x_b(t - \delta(t)) = i_2(t - \delta(t)).
\]

(29)

The mapping \(\Phi_b(x_b)\) for System B with \(n = 1\) is

\[
\Phi_b(x_b) = x_b = i_2(t).
\]

(30)

Functions of the form \(\Phi(x) = x, x \in \mathbb{R}\) are diffeomorphic and \(\Phi(x_b)\) is a diffeomorphism for \(x_b \in \mathbb{R}\). The Jacobian for System B

\[
J_b(x_b) = 1
\]

(31)

which is indeed nonzero for all \(x_b \in \mathbb{R}\).

We now examine whether System B possesses properties (P1) and (P2). For the first property, note that the first condition in (6) doesn’t apply since \(n = 1\). The second condition is true since \(L_gL_f^\dagger h(x_b) = -\eta_2 e_2(t) \neq 0\), for some \(x_b \in \mathbb{R}\), in particular whenever \(e_2(t) \neq 0\). Note the function

\[
p(z_b, u_b) = \left( L_f^\dagger h(x_b) + L_gL_f^\dagger h(x_b)u_b \right)_{x_b = \Phi^{-1}(x_b)} = \alpha_b z_b - \eta_2(\sigma_2^2 - \epsilon_2^2(t))^2
\]

(32)

is indeed well-defined. Considering property (P2) for System B, note that \(p(z_b, u_b)\) is globally uniformly Lipschitz continuous with respect to \(z_b\); its derivative with respect to \(z_b\) is equal to constant \(\alpha_b\) for all \(z_b \in \mathbb{R}\). The Lipschitz coefficient can be determined from the inequality

\[
\|p(z_{b,1}, u_b) - p(z_{b,2}, u_b)\| \leq \gamma(\|u_b\|) \|z_{b,1} - z_{b,2}\|
\]

(33)

from which we derive that \(\gamma \geq \|\alpha_b\|\), and is therefore nondecreasing in \(\|u_b\|\).

B. Observer Construction and Parameter Selection

The observer \(\hat{O}_{a-b}\) estimates input current \(i_2\) at System A based on measurements of signal \(i_2(t - \delta_{b-a}(t))\) received from System B and is constructed as

\[
\frac{di_2(t)}{dt} - \mathbf{A}_b i_2(t) + \mathbf{B}_b u_b + \mathbf{J}_b^{-1}(i_2(t)) k_{0b} e^{-\rho_{b-a}\tau_{u}} - \mathbf{J}_b^{-1}(\hat{i}_2(t - \delta_{b-a}(t)))
\]

(34)

The observer \(\hat{O}_{b-a}\) estimates voltage \(v_2(t)\) based on measurements of signal \(v_2(t - \delta_{b-a}(t))\) received from System A and is constructed as

\[
\frac{dv_2(t)}{dt} - \mathbf{A}_a \hat{x}_a(t) + \mathbf{B}_a u_a + \mathbf{J}^{-1}_a(\hat{x}_a(t)) k_{0a} e^{-\rho_{a-b}\tau_{u}} - v_2(t - \delta_{b-a}(t))
\]

(35)

where \(\hat{x}_a = [\hat{i}_1 \hat{v}_1 \hat{i}_2 \hat{v}_2]^T\). In (35) and (36), \(k_{0a(b)}\) denotes the parameter \(k_0\) in (8) for the observer that estimates states physically located at System A(B).

A reasonable guess for selecting suitable \(k_{0a(b)}\) (which must be validated, as shown in the next subsection) is to compute the feedback vector for the so-called Luenberger observer [32] corresponding to each of the subsystems. This is the method that was used in this work, where the MathWorks place command was used to place closed-loop poles at desired locations to yield satisfactory total system response. Note that it is always possible to obtain these feedback vectors since the subsystems are confirmed to be observable in step B) of the design process.

C. Validation in Simulation

The delay compensation technique was applied to the partitioned circuit in Fig. 4 using the parameter values shown in Table I. [Note that these parameters are not intended to represent a specific system—they were chosen only to clearly visualize the difference in the compensated and uncompensated responses.] The simulations were performed in MathWorks SimPowerSystems toolbox, version 5.7 [33].

The passive circuit elements, error gains and damping factors in Table I were chosen arbitrarily, except that \(A_{a} \) in (25) was valideted to be invertible. The communication delay \(T_d\) was chosen to be several times greater than the empirically measured network delay (approximately 30 ms per round trip, as described in Section V). The update rate \(\tau_u\) was chosen to be an order of magnitude greater than \(T_d\); the desired convergence rate \(\rho\) was arbitrarily chosen to visually observe the response. The vector \(k_{0a}\) was calculated as \([4.06 6.18 4.18 1.06]^T\) for poles placed at \([-1.00 -1.01 -1.02 -1.03]^T\). Similarly, \(k_{0b}\) was calculated as \([1.0]\) for a pole placed at \([-1.0]\). The selection of
TABLE I
SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacitances, $C_1$, $C_2$</td>
<td>1.0 [F]</td>
</tr>
<tr>
<td>Resistance, $R$</td>
<td>0.5 [Ω]</td>
</tr>
<tr>
<td>Inductance, $L$</td>
<td>1.0 [H]</td>
</tr>
<tr>
<td>Voltage Error Gain, $\eta_1$</td>
<td>$-1.0 \text{ [sΩ]^{-1}}$</td>
</tr>
<tr>
<td>Voltage Error Gain, $\eta_2$</td>
<td>$-1.0 \text{ [sΩ]^{-1}}$</td>
</tr>
<tr>
<td>Damping Factor, $\alpha_1$</td>
<td>$-1.0 \text{ [s]^{-1}}$</td>
</tr>
<tr>
<td>Damping Factor, $\alpha_2$</td>
<td>$-1.0 \text{ [s]^{-1}}$</td>
</tr>
<tr>
<td>Communication Delay, $T_d$</td>
<td>0.20 [s]</td>
</tr>
<tr>
<td>External Command Update Rate, $\tau_u$</td>
<td>2.0 [s]</td>
</tr>
<tr>
<td>Observer Convergence Rate, $\rho$</td>
<td>4.0 [ms]</td>
</tr>
</tbody>
</table>

Fig. 6. Signal representing external sources.

Fig. 7. Simulation results from example partitioned system. Blue lines signify the ideal response (from the ideal circuit in Fig. 3), green lines the uncompensated response, and red lines the observer output.

For simplicity, the input signals $e^*_1(t)$ and $e^*_2(t)$ representing externally connected sources in (12) were both set equal to the waveform $e^*(t)$ depicted in Fig. 6. This function held a (randomly sampled) constant value which was updated every time interval $\tau_u$ after the start of the simulation.

The blue lines of Fig. 7 signify the ideal response (from the ideal circuit in Fig. 3), green lines the uncompensated response, and red lines the observer output. As can be seen, the ideal and observer-based (“compensated”) signals converge for both voltage and current, despite continued changes in the input signals $e^*_1(t) = e^*_2(t) = e^*(t)$ throughout the simulation.

Fig. 8. Errors in $i_2$ with (upper) and without (lower) delay compensation.

Errors between compensated and uncompensated responses are plotted in Fig. 8 (for current $i_2$). In particular, $\phi(t)$ is defined as the difference between the compensated and ideal response (blue and red lines in Fig. 7, upper); $\hat{\phi}(t)$ is the difference between the uncompensated and ideal response (green and red lines in Fig. 7, upper).

As shown in Fig. 8, the error in the compensated current signal decreases exponentially to zero, as predicted by (9). Remarkably, once the controller error converges to zero it stays at zero even after the external inputs $e^*_1(t)$, $e^*_2(t)$ continue to change. Note that continued convergence to zero in the compensated systems, despite changing external input signals, are likely possible in this example because models for the full and partitioned systems were known exactly.

V. HARDWARE RESULTS

This section describes experimental validation of the ODC method in hardware. During this experiment, a closed-loop, remote HIL experiment emulating the system shown in Fig. 5, employing the architecture shown in Fig. 2, was performed. This experiment provided a virtual connection of equipment physically located at Colorado State University (CSU), Fort Collins, CO, USA, with equipment located remotely at the National Renewable Energy Laboratory (NREL), Golden, CO, USA. The physical separation between these locations was approximately 115 km. A depiction of the overall experiment is shown in Fig. 9.

In Fig. 9, the System A (“Sys A”) and System B (“Sys B”) portions of the electrical circuit in Fig. 4 were simulated on OPAL-RT real-time digital simulators [34] at CSU and NREL,
This paper introduced a potential method and architecture for helping to mitigate communication delays between remotely connected HIL experiments. Motivation for this paper was the desire to connect geographically distant software/hardware testbeds while minimizing the loss of experimental fidelity arising from communication latencies. The approach was validated through computer simulation and a remote HIL experiment where the subsystems were separated by a physical distance of 115 km and communication delay of 30 ms. However, we note that the convergence results shown in this paper assumed perfect knowledge of the underlying system models. Primary challenges in implementing the method are generating sufficiently accurate mathematical descriptions of the subsystems and determining controller parameters. Automated state-model generation or average value models can possibly be used for more complex systems to address the first challenge. Theoretical development to extend the convergence guarantee for drift-observable, time-delayed observer design for multiple remote systems and design guidelines for optimal parameter selection are needed to resolve the second challenge. Additional future research topics include the addition of robust control techniques to account for potential parameter variations and imperfect system knowledge.

VI. CONCLUSION AND FUTURE RESEARCH

Fig. 10. Measurements from hardware validation of ODC method: compensated signals (red); ideal signals (black); and external input signals (blue).

respectively. The observers for both systems were executed on Arduino EtherDUE [35] at each location. The EtherDUE microcontrollers had a built-in Ethernet port in addition to analog and digital I/O channels.

The observer calculations were executed on the microcontroller boards rather than on the real-time simulators in order to isolate the electrical simulations from the observer computations. This approach provided the following advantages:

1) software for network packet manipulation and time functions were accessible through standard libraries readily installed on the microcontroller boards;

2) processing network signals externally to the real-time simulators eliminated the need for using limited memory and computational resources on the real-time simulators;

3) isolation of the electrical simulation and observers made the architecture more maintainable, since components of the system could be modified and tested independently.

Measurements and observer values were passed between colocated OPAL-RT and Arduino microcontrollers using digital and analog I/O channels. State information was passed between remote locations through the EtherDUE boards using TCP/IP communication. Synchronization between systems was implemented using algorithms on the Arduino boards in a master/slave, hand-shaking configuration.

As mentioned in Section IV, selection of observer parameters and stability analysis for this experiment required an estimate of the network delay between Systems A and B. Since network delays between two systems are generally dependent on many factors (e.g., network load and routing algorithms), the estimated delay was determined experimentally using repeated ping algorithms on the microcontrollers. The sample mean of the round-trip communication delay was found to be \( \approx 30 \) ms.

Measurements obtained during the experiment are shown in Fig. 10. The compensation algorithms were activated at each location at approximately \( t = 57 \) s. The measurements show that the compensated currents and voltages (red lines) converged to the ideal response signals (black lines) with little error after the delay compensation algorithms converged. Note that as predicted in simulation, once the observer-based delay compensators converged, the error remained near zero even when the external input signal continued to change.

REFERENCES


Gerald Duggan received the B.S. degree in math and computer science from New Mexico State University, Las Cruces, NM, USA, and the M.S. degree in computer science from the University of Utah, Salt Lake City, UT, USA, in 1983 and 1985, respectively.

From 1985 to 2009, he was a Senior Software Architect with Hewlett Packard, defining standards and implementing software for large-scale IT management systems. He is currently the Lab Manager of the Smart Village Microgrid Lab, Energy Institute of Colorado State University, Fort Collins, CO, USA. His research interests include village-scale microgrids, dc power distribution systems, and ac–dc efficiency studies.

Poorva A. Bedge (S’15) received the B.S. degree in electrical engineering from the Government College of Engineering Pune, Pune, India, in 2004, and the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2008. She is currently working toward the Ph.D. degree in systems engineering with Colorado State University, Fort Collins, CO, USA.

She worked in Mumbai and Delhi, India with Reliance Energy as a Senior Engineer on the electric power distribution system. At Midwest ISO, she was an Outage Coordination Engineer, and at California ISO she was a Senior Operations Planning Engineer. Her research interests include developing tools and solutions to modernize the electric grid to accommodate the rapid increase in renewable energy.

Daniel Zimmerle received the B.S. and M.S. degrees in mechanical engineering from North Dakota State University, Fargo, ND, USA, in 1983 and 1984, respectively.

From 1985 to 2005, he worked in various roles at Hewlett Packard and Agilent Technologies including both a division R&D and General Manager. From 2006 to 2007, he was the Chief Operating Officer with Spirae, Inc. He is currently a Senior Research Associate and the Director of the Electric Power System Laboratory, Energy Institute, Colorado State University, Fort Collins, CO, USA. His research interests include microgrids and the development, integration, and field assessment of distributed, renewable, and waste heat electrical generation systems.

Leah Holton received the B.S. degree in mathematics and the B.S. degree in engineering from James Madison University, Harrisonburg, VA, USA, in 2012, and the M.S. degree in mechanical engineering from Arizona State University, Tempe, AZ, USA, in 2014.

She is currently a Solar Engineer with the Independent Engineering Group, ICF, Denver, CO, USA. She supports solar energy project owners and investors with due diligence and owners engineering services.