Received 25 July 2023, accepted 15 September 2023, date of publication 19 September 2023, date of current version 27 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317250

TOPICAL REVIEW

Overview of Interface Algorithms, Interface Signals, Communication and Delay in Real-Time Co-Simulation of Distributed Power Systems

ELUTUNJI BURAIMOH[1], (Member, IEEE), GOKHAN OZKAN[1,2], (Senior Member, IEEE), LAXMAN TIMILSINA[1], (Graduate Student Member, IEEE), PHANI KUMAR CHAMARTHI[1], (Member, IEEE), BEHNAZ PAPARI[1,3], (Senior Member, IEEE), AND CHRISTOPHER S. EDRINGTON[1], (Senior Member, IEEE)

[1]Real-Time COntrol and Optimization Laboratory (RT-COOL), Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA

[2]Dominion Energy Innovation Center, Clemson University, North Charleston, SC 29405, USA

[3]Department of Automotive Engineering, Clemson University, Greenville, SC 29607, USA

Corresponding author: Elutunji Buraimoh (eburaim@clemson.edu)

This work was supported by the Clemson University's Virtual Prototyping of Autonomy Enabled Ground Systems (VIPR-GS) through the U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC) under Grant W56HZV-21-2-0001.

ABSTRACT In recent years, there has been growing research interest in the virtual integration of hardware and software assets across different geographical locations. However, achieving joint real-time simulation in virtually connected laboratories presents several challenges that must be addressed. One critical aspect involves selecting suitable interface algorithms to conserve energy, ensure signal decomposition, and reconstruct data accurately, thereby preventing distortions. Additionally, communication latencies must be taken into account to maintain experiment fidelity. The establishment of fast and reliable communication between real-time simulators in different laboratories through coupling interfaces, is of utmost importance for successful real-time co-simulation. Nevertheless, assuming the constant availability of reliable and delay-free communication is unrealistic, which can lead to performance degradation and system instability. These requirements pose significant obstacles to implementing virtual integration involving various real-time simulators and hardware-in-the-loop setups across diverse laboratories. The real-time co-simulation of power systems, in particular, is highly susceptible to ineffective interface algorithms, signal decomposition, and reconstruction, as well as communication delays, potentially causing loss of synchronism and negatively impacting simulation fidelity. Consequently, these limitations render such setups unsuitable for dynamic and transient studies. In light of these challenges, this paper aims to provide an overview of interface algorithms, signal decomposition and reconstruction techniques, communication protocols, and delay compensation approaches. The goal is to ensure system fidelity, enhance coupling point modeling, and improve the accuracy of real-time co-simulation in virtual environments across long distances while preserving ongoing research confidentiality, safeguarding intellectual property, and facilitating collaborative research.

INDEX TERMS Communications latency, delay compensation, interface algorithms, power system real-time co-simulation, signal decomposition.

NOMENCLATURE

AITM	Advanced Ideal Transformer Model.
AM	Amplitude Modulation.
BMC	Best Master Clock.
CHIL	Contoller Hardware-in-the-Loop.
CPU	Central Processing Unit.
DIM	Damping Impedance Method.
DP	Dynamic Phasors.
EMT	Electromagnetic Transients.

The associate editor coordinating the review of this manuscript and approving it for publication was Youngjin Kim.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

FCF	Feedback Current Filter.
FPGA	Field-Programmable Gate Array.
GDRTCS	Geographically Dispersed Real-Time Co-Simulation.
GDRTCS	Geographically Dispersed Real-Time Simulation.
GPS	Global Positioning System.
HIL	Hardware-in-the-Loop.
HVDC	High-Voltage Direct Current.
IA	Interface Algorithm.
INL	Idaho National Laboratory.
IP	Internet Protocol.
IRIG-B	Inter Range Instrumentation Group Timecode B.
ITM	Ideal Transformer Model.
LPF	Low Pass Filter.
MATLAB	MATrix LABoratory.
MDIM	Modified Damping Impedance Method.
NREL	National Renewable Energy Laboratory.
NTP	Network Time Protocol.
PCD	Partial Circuit Duplication.
PHIL	Power Hardware-in-the-Loop.
PSCAD	Power System Computer-Aided Design.
PTP	Precision Time Protocol.
PV	Photovoltaic.
RMS	Root Mean Square.
RT	Real-Time.
RTP	Real-time Transport Protocol.
RTS	Real-Time Simulator.
SRF	Synchronous Reference Frame.
TCP	Transmission Control Protocol.
TFA	Time-Variant First-order Approximation.
TLM	Transmission Line Mode.
UDP	User Datagram Protocol.
VPN	Virtual Private Network.
VSS	Virtual Simulated System.
WAN	Wide-Area Network.
WV	Wave Variables.

I. INTRODUCTION

Real-time simulators are crucial in studying technically complex power system scenarios using predefined time steps. These predefined time steps are particularly advantageous in accurately analyzing high-frequency events like transients [1]. These simulators are typically integrated with hardware or a user interface. Depending on the underlying model dynamics, real-time simulations can be either Central Processing Unit (CPU)– or Field-Programmable Gate Array (FPGA)-based. The simulation time step for CPU-based implementations is constrained to 5-10 μs due to hardware limitations, while FPGA-based simulations can achieve much lower time steps, resulting in improved system dynamics. Nonetheless, FPGAs do have lower computing capability, which means they are best suited for smaller simulated systems [2]. The application of real-time simulators is especially valuable for power systems engineers as these simulators allow for the evaluation of control strategies and operations. Given the dynamic nature of power systems and the need to handle transients, real-time modeling and simulation become essential. While real-time simulators are versatile tools, they do have some limitations. Each individual real-time simulator unit is confined by its computational power and functionality. Moreover, they are designed for specialized applications, relying on specialized software and hardware interfaces to perform complex and specialized power system tasks [3].

Establishing a pool of real-time simulators necessitates significant investment and may prove challenging within a single laboratory setting. The geographical distribution of real-time simulators and system components being tested can pose additional challenges, as the local simulator may be insufficient to handle certain complex model simulations. To address this, a feasible approach involves partitioning or splitting the complex model to enable distributed and collaborative simulations [4]. Furthermore, the presence of real-time simulators scattered across the globe, each with its unique operational characteristics and limitations, calls for implementing wide area network-based real-time geographically distributed simulations. This approach, as highlighted by [5], allows researchers to leverage these distinct research facilities effectively. Also, there is a growing interest in integrating these engineering assets to achieve seamless collaboration and facilitate energy, control, and power systems research without infringing on intellectual property [6]. This integration not only reduces research costs but also promotes industry-level academic research. Thus, numerous national laboratories, universities, and industrial companies are now establishing virtually linked real-time Hardware-in-the-Loop (HIL) simulation setups. These setups enable participating HIL subsystems at different sites to communicate and share system parameters, effectively emulating a large and complex system for research purposes. This collaborative approach fosters a more unified and powerful research environment for tackling challenging power system scenarios.

The scientific community continually develops innovative approaches to address the challenges posed by complex power systems. Offline simulations have been instrumental in demonstrating the feasibility of these approaches. However, there is a growing need to evaluate these solutions' practical settings and real-time performance to showcase their scalability and robustness. Real-Time (RT) simulations, constrained by hardware limitations, often resort to using simplified models or focusing on specific sections of the power network model. In contrast, offline simulation programs such as MATrix LABoratory (MATLAB), MATLAB/Simulink, and Power System Computer-Aided Design (PSCAD), offer the potential for more detailed modeling. To overcome these limitations and expand real-time simulation capabilities across various locations, the concept of Geographically Dispersed Real-Time Co-Simulation (GDRTCS) has garnered considerable attention and investigation. GDRTCS facilitates the seamless integration of power electronics, improved communication structures, and novel control algorithms, all working towards ensuring grid controllability and stability [4]. Additionally, the power system domain has witnessed the emergence of ''co-simulation'' or ''hybrid simulation'' techniques. This involves simulating a single electrical network using two or more solvers [7].

Several suppliers provide software and hardware real-time simulators with diverse architectures and capabilities, offering distinctive power and energy system analysis environments. However, computational capacity and cost constraints limit the size of power systems that can be simulated in real-time. Many real-time simulators incorporate communication components to overcome this limitation, making geographically distributed real-time simulation a viable option. This approach enables the simulation of larger power systems and facilitates resource sharing. The communication components utilized in these setups support Internet Protocol (IP), enabling seamless data exchange. Geographically distributed real-time simulations offer numerous advantages, as highlighted by [5] and [8].:

- Encourage and promote collaboration and investment between power and energy research centers.
- Use distributed real-time simulation to simulate larger-scale electric power systems.
- Connect real-time simulators at universities, national laboratories, and electric utilities to common-use of renewable and distributed energy resources infrastructure.
- Allow remote testing and control of power systems and devices.
- Potential to share, access, and interconnect shared simulation software and hardware facilities.
- Sharing of expertise: several laboratories with different existing apparatus may benefit from the technical expertise of a collaborating different laboratory.
- Guaranteeing protection of intellectual properties and research data confidentiality.

In addition, one reason to perform a co-simulation utilizing two distinct simulators is to take advantage of existing model libraries and simulation resources [8]. GDRTCS has many benefits but also many challenges, including:

- Partitioning the power system model, determining the appropriate points of coupling, and exchanging information across subsystems.
- Interface algorithm; incorrect choice of Interface Algorithm (IA) and wrong choice of decoupling variables for a given simulated system causes instability, notwithstanding the real system's stability.
- Synchronizing various real-time simulators and hardware.
- Sustaining communication reliability among partitioned subsystems; connecting two distinct simulators necessitates a delay. Utilizing the internet as the communication medium for GDRTCS presents additional constraints.

Real-time model simulation runs or executes in time of the actual clock. For example, a fixed-time step solver is executed in real-time provided the computation time is not more than the time step [9]. Typical real-time simulators use milliseconds to microseconds for time steps, while Wide-Area Network (WAN) data latency can be several hundred milliseconds. As a result, communication latency between real-time simulators can cause inaccuracies and instability in geographically distributed simulations [5]. In addition to model partitioning and interface creation for parallel execution, the delay time could be longer than the Electromagnetic Transients (EMT) simulation time step; thus, delay-compensating interface algorithms can help mitigate and alleviate the effects of delay [4]. Similarly, synchronizing simulation time with a typical clock ensures externally connected hardware and other simulators communicate with a particular real-time simulator for HIL and GDRTCS.

Real-time digital simulators virtualized across long distances with distributed parallel subsystems increases computing capacity and scalability. However, this shift to power electronics technologies in power system engineering creates challenges; power grid design and assessment require large-scale, multidomain experiments and simulation [10], [11]. Secondly, a harmonized transition to future grids necessitates an overhauling approach to evaluating synergies and identifying unpredictable interactions. Research facilities use multiple digital real-time simulation units and comprehensive test benches to address these issues. However, a large-scale, complete testing and simulation facility at a single laboratory is prohibitively expensive and impractical [12], [13].

Power systems simulation in real-time aims to precisely emulate the power systems. Real-time power system simulation usually uses fixed-time steps, which is mandatory for RT simulation due to the fixed-step solver. The variable step solver is not utilized in RT because, within a single simulation experiment, the sample time is automatically updated several times based on the simulation dynamics; it entails repeated iterations of the same step and is indeterminate [14]. Hence, fixed-time steps are consistent with and applicable in real-time simulation when synchronized by a clock signal. However, nonlinear activities like power electronic switching can cause a jitter due to time-step synchronization; thus, such activities are only considered at the start of the next step and not during the occurrence. In addition, numerical instability requires specific techniques and solving methods [4].

Internet, with or without synchronization by Global Positioning System (GPS) or a Virtual Private Network (VPN), serves as a veritable medium for simulators participating in GDRTCS experiments to communicate and solve common complex problems, thus, permitting computational capacity sharing [15]. A virtual interconnected real-time laboratory consists of the following [12], [13].

- HIL setups,
- Real-Time simulators
- Co-simulation frameworks

These facilities ensure that collaborations and joint testing can be leveraged even with confidentiality constraints. Figure 1 shows how GDRTCS connects geographically

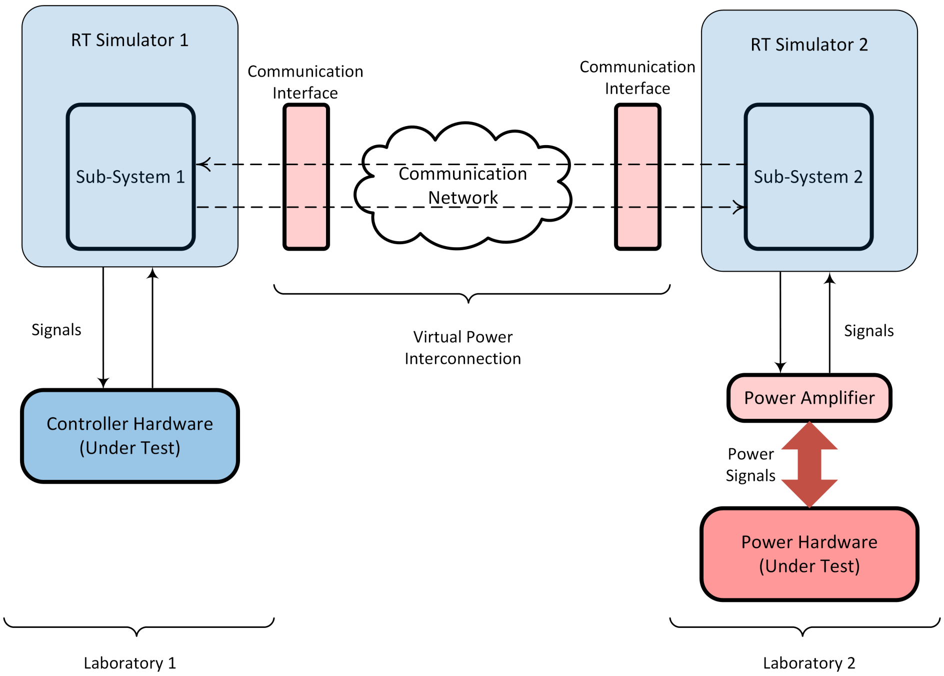

FIGURE 1. GDRTS schematic.

dispersed HIL setups interconnected and exchange data via the internet [12], [16].

This overview comprehensively examines various aspects critical for achieving accurate and reliable real-time co-simulation in a virtual environment, particularly when operating over long distances. The focus is on interface algorithms, signal decomposition, reconstruction techniques, and the communication and delay prediction required to compensate for latency to maintain system fidelity. Moreover, the study emphasizes the significance of improved coupling point modeling, which enhances the accuracy of the co-simulation process. It is important to note that the overview considers the utmost importance of preserving ongoing research confidentiality and safeguarding intellectual property. This overview is organized as follows; Section I introduces the background, real-time simulation, and the concept of GDRTCS, which is the virtual integration and interaction of several real-time simulators. Section II presents the various models of interface algorithms applicable in real-time co-simulations. Section III presents the interface signals' transformation and reconstruction and the various coupling and coupling points in real-time co-simulations. Section IV presents the communication network and protocols deployed in real-time co-simulation. Section V also presented the various frameworks deployed to implement geographically distributed real-time simulation experiments. Section VI presented latency associated with real-time co-simulation and communication delay compensation strategies used in addressing communication latency. Finally, Section VII presents the conclusion and recommended future work.

II. INTERFACE ALGORITHMS

Electrical signal coupling requires selecting a suitable interface: IA and interface signal [17]. The interfaces deployed to monolithic Power Hardware-in-the-Loop (PHIL) also apply in the virtual coupling of Geographically Dispersed Real-Time Simulation (GDRTCS) or GDRTCS. GDRTCS necessitates that the investigated system be detached into various subsystems. Different digital real-time simulators simulate subsystems and exchange interface signals or quantities according to a predefined IA. A co-simulation IA precision is crucial for obtaining high fidelity in geographically dispersed simulation [18].

IA design aims to account for the communication network's impact and ensure interface energy conservation. The IA defines the connecting strategy of the Geographically Distributed Real-Time Simulation subsystems [19]. This critical point signifies the interconnection set-up of two subsystems in co-simulation. The IA selection determines the co-simulation experiment stability [20]. Several PHIL interface strategies have been developed, as reviewed in [21] and [22], and these strategies are used in geographically distributed real-time simulation setups due to the similar electrical system partitioning and real-time requirements of these two types of systems [19].

In general, the architecture of a PHIL is represented as a Thévenin equivalent, as depicted in [23]. An adaptive power coupling element is required to interact between physical and virtual power domains. By bi-directionally connecting a Virtual Simulated System (VSS) to a, an IA must be used to remove disturbances and minimize inaccuracies caused by the coupling, such as measurement probes and power amplifiers in

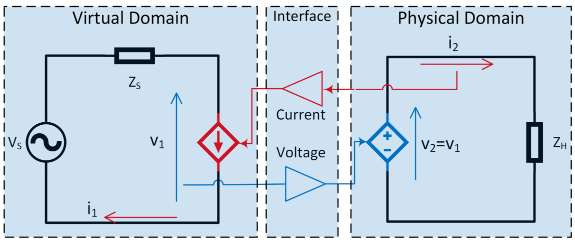

FIGURE 2. PHIL system scheme demonstrating virtual and physical domains and their interfaces.

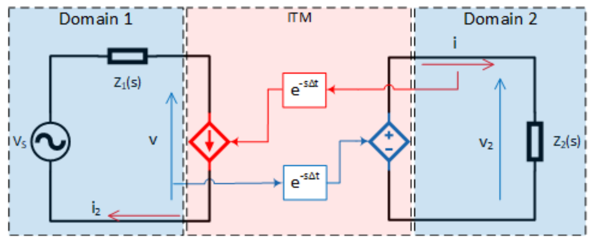

FIGURE 3. ITM interface scheme.

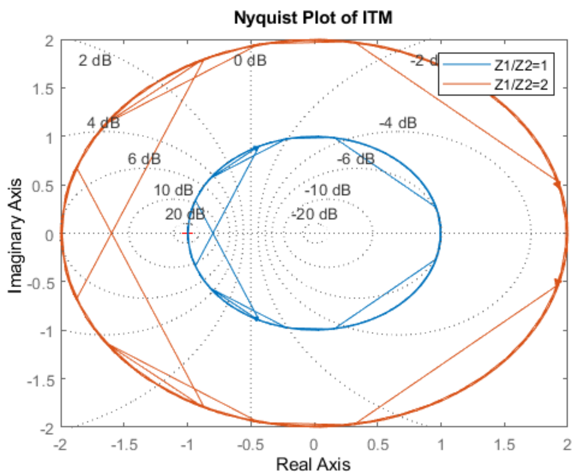

FIGURE 4. ITM range of operation.

order to run stable and realistic experiments [21]. Furthermore, in Figure 2, VSS, represented by V_0 and Z_S, is used to execute power system simulations for the stability study [21]. A similar representation of the PHIL test system with power amplifiers and sensors is given in [24]. Power amplifiers and probes are necessary for PHIL systems, but they can disrupt system dynamics.

Studies [22], [25], [26], [27], and [28] all listed eight PHIL interfaces, and Ideal Transformer Model (ITM) algorithm is adjudged to be the most simple and accurate, but time delay makes it unstable [24]. The following is the list of IAs as listed in the literature; ITM algorithm, Advanced Ideal Transformer Model (AITM), Time-Variant First-order Approximation (TFA) algorithm, Transmission Line Mode (TLM) algorithm, Partial Circuit Duplication (PCD) method, Damping Impedance Method (DIM) and IA Extendible Feedback Current Filter (FCF). A good IA should increase accuracy and stability while ensuring interaction between the subsystems simulated in two different simulations. In PHIL, the software PHIL IA can increase the system's stability and accuracy while contending with the challenges of delay, bandwidth, and noise. Two essential characteristics identify the IAs: the transmitted signal type (for example, power, current, voltage, force, torque) and the signal processing methods (for example, low pass filtered, unity gain, a lead-lag compensator) [22].

A. IDEAL TRANSFORMER MODEL TECHNIQUE

ITM is a widely used and simple technique for PHIL applications to link the simulated system and power device or hardware under test [3], [18]. Typically, most co-simulation IA approaches are based on ITM, which uses controlled voltage and current sources to impose the remote subsystem's behavior on the local subsystem [29]. The ITM algorithm is a conventional or basic power HIL modeling approach; thus, depending on the amplified signals, ITM is grouped into voltage-type or current-type. A power HIL simulation's error is assumed to be only lumped time. Figure 3 shows this interface's total time delay Δt (including filter and amplifier), and Equation 1 defines the ITM algorithm's corresponding open-loop transfer function. This technique is the simplest and most common, with minimum computation, good precision, and a relatively low stability region [30].

$$G_0 = \frac{Z_1(s)}{Z_2(s)} e^{s\Delta t} \tag{1}$$

$Z_1(s)$ and $Z_2(s)$ are the corresponding simulated and hardware network impedances. A Thevenin or Norton equivalent circuit can represent any electrical system at any time. Since its values change during the test, the ratio between simulated $Z_1(s)$ and hardware or device's impedance $Z_2(s)$ is crucial for PHIL experiments' consistent working range because $Z_1(s)/Z_2(s)$ determines the interface's stability. Using Nyquist, several impedance ratios were used to calculate the ITM's stable operational range by [21] and [31], as shown in Figure 4, which demonstrates the ITM's stable range; any value below the impedance ratio 1 surface is considered stable. The system is steady at $Z_1(s)/Z_2(s) = 1$; beyond this, the system is unstable, per Nyquist. As per the Nyquist criterion, a system's stability is contingent upon the fact that the Nyquist plot of its transfer function should not encircle the point $(-1 + 0j)$ on the complex plane. Even if this requirement ensures stability, a substantial latency of the pathways Δt could cause nonlinearities (phase shift) that affect the system's frequency domain and time-domain accuracy [27].

This interface can be used for co-simulation with a remotely regulated current or voltage source at each end. For implementation in GDRTCS, a controlled voltage source represents voltage from subsystem 1 at subsystem 2, and a controlled current source receives the sent current signal from subsystem 2 to impose on subsystem 1 [22], [32]. Furthermore, this technique's simple implementation and good stability made it suitable for co-simulation [20]. Adding a fixed series

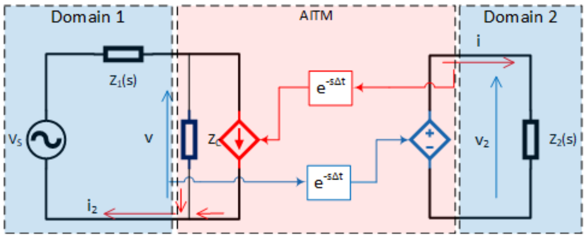

FIGURE 5. AITM interface scheme.

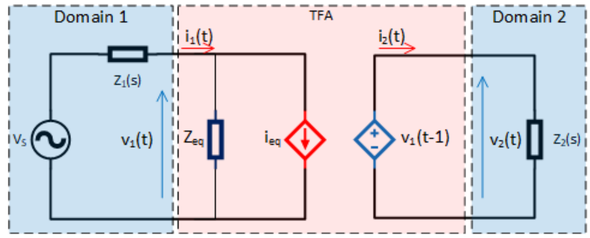

FIGURE 6. TFA interface scheme.

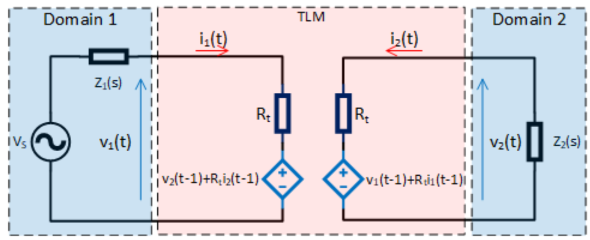

FIGURE 7. TLM interface scheme implementation.

resistance and shunt admittance to the voltage and current sources can prevent numerical instability.

B. ADVANCED IDEAL TRANSFORMER MODEL TECHNIQUE

To enhance the ITM stability to form a new AITM [24], an extra compensation impedance Z_C is added to subsystem 1, as shown in Figure 5. Equation 2 explains the concept of AITM, where Z_C must be bigger than 0 for the AITM to avoid short-circuiting the current injecting source. Figure 6 displays the AITM's stable impedance ranges. Only at 0.5 Z_C does the system enter an unstable condition when $Z_1(s)/Z_2(s)$ is over 1.5. (red-yellow surface). It must be noted that an additional component to the system can alter accuracy; thus, Z_C must be minimal [21].

$$G_0 = \frac{Z_1(s)}{Z_c(s) + Z_2(s)} e^{s\Delta t} \tag{2}$$

C. TIME-VARIANT FIRST-ORDER APPROXIMATION

This is a predictive approach for approximating the dynamics of the device under test using linear systems of the first order [25]. The historical simulation data is stored, updated online, and utilized in real-time to solve and update the system and hardware coefficients [33]. The simulator can then rectify interface errors. Considering a typical RL circuit application and assuming the hardware has a first-order Equation 3, the current i_2 value is derived from the initial values voltage v_1 and current i_2, by discretizing with a trapezoidal approximation, as illustrated in Equation 4. Furthermore, Equation 4 can be solved using Equation 5. Lastly, Figure 6 depicts RL-type TFA implementation [22].

$$\frac{di_2(t)}{dt} = ai_2(t) + bv_2(t) \tag{3}$$

$$i_2(t) \approx \alpha[v_1(t-1)] + \beta[i_2(t-1)] = \frac{v_1(t-1)}{Z_{eq}} + I_{eq} \tag{4}$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} v_1(t-2) \; i_2(t-2) \\ v_1(t-3) \; i_2(t-3) \end{bmatrix}^{-1} \begin{bmatrix} i_2(t-1) \\ i_2(t-2) \end{bmatrix} \tag{5}$$

D. TRANSMISSION LINE MODEL

TLM links represent discrete reactive components modeled as transmission-line sections [34]. This method considers reactive components as two-port devices. The capacitor voltage or inductor current change within each simulation time step is minimal; thus, two circuit models can be decoupled and appear as though they are connected to a constant current or voltage source, as shown in Figure 7. The TLM uses a linking inductor or capacitor using the Bergeron line model to replace L or C with R_t. To replace L with R_t, L is divided by Δt is the transmission line propagation time (also interface time delay), while to replace C with R_t, Δt is divided by C. This method is useful for decoupling big systems for parallel computation since the transmission line may be described in the Bergeron line model [22]. TLM's transfer function is given in Equation 6:

$$G_0 = \frac{1 - \alpha e^{-2s\Delta t}}{1 + \alpha e^{-2s\Delta t}} \times \frac{Z_1(s)}{R_t} \tag{6}$$

where $\alpha = \frac{Z_L - R_t}{Z_L + R_t}$

E. PARTIAL CIRCUIT DUPLICATION

Dmitrie Zodorov and Bernhard Klaassen, in reference [35], [36], proposed two IAs, namely the PCD and DIM. The technique splits the initial circuit into subcircuits and iterates for a solution. Figure 8 depicts PCD's algorithm scheme. The two Z_{12} are extra coupling impedances in the two subsystems. Thus, the link Z_{12} is duplicated in the simulated network and test hardware. PCD loop transfer function is given by Equation 7 [21], [22].

$$G_0 = \frac{Z_1(s)Z_2(s)}{[Z_1(s) + Z_{12}(s)][Z_2(s) + Z_{12}(s)]} \times e^{s\Delta t} \tag{7}$$

The PCD method is premised on the reduction technique, which divides the entire circuit under test into distinct units and iteratively approaches a solution. The loop equation concludes that the PCD technique is more stable than the ITM for resistive systems because keeping the transfer function below unity is easier. If this technique converges,

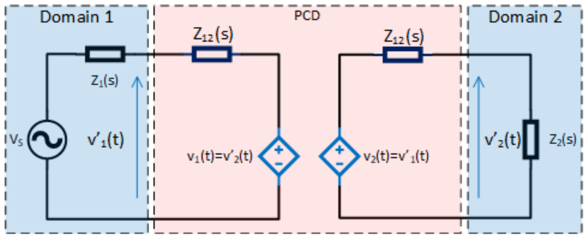

FIGURE 8. PCD interface scheme implementation.

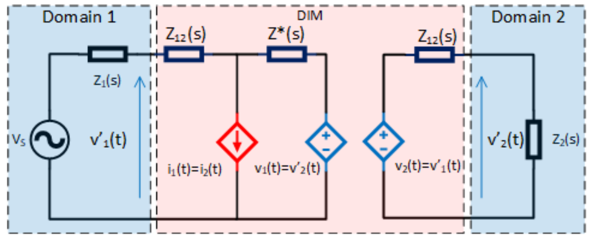

FIGURE 10. DIM interface scheme implementation.

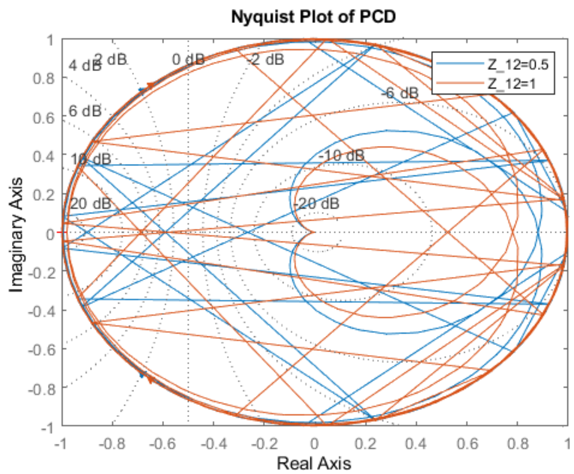

FIGURE 9. ITM range of operation.

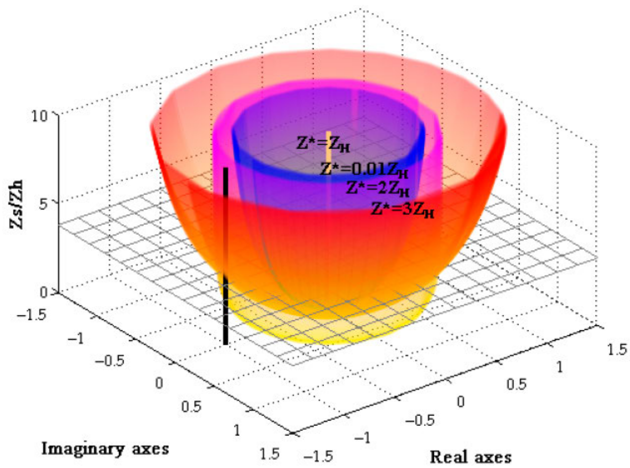

FIGURE 11. DIM Stable operational ranges [21].

many iterations can accomplish accurate simulation. Real-time simulations allow only one iteration per time step. Each iteration's error must be minimal. To preserve accuracy, Z_{12} should be larger than Z_1 and Z_2. In practice, this is difficult to do; hence PCD accuracy is generally low. Figure 9 indicates the system is stable even with low coupling impedance Z_{12}. Therefore, Adding impedances to the setup results in additional power consumption, affecting fidelity.

F. DAMPING IMPEDANCE METHOD

The DIM incorporates an impedance that optimizes the convergence factor, hence enhancing the PCD algorithm's stability and precision [35], [36] [37]. DIM is typically a hybrid of the PCD and ITM, is achieved by placing a damp impedance, Z^*, in between current and voltage sources. The open-loop transfer function is depicted in Equation 8, and it is evident that when Z^* approaches infinity, the DIM assumes the ITM configuration an,d when Z^* approaches zero value, the PCD configuration [37]. The DIM technique combines the PCD and ITM methods with Z^*. Figure 10 shows its implementation. Depending on the Z^* estimate, this interface's stability lies between PCD and ITM; thus, absolute stability is ensured when Z^* matches Z_2.

$$G_0 = \frac{Z_1\ (s)[Z_2\ (s) - Z^*(s)]}{[Z_1(s) + Z_{12}(s)][Z_2(s) + Z_{12}(s) + Z^*(s)]} \times e^{-s\Delta t} \tag{8}$$

When $Z = Z_2$, G_O for DIM becomes 0, the PHIL system IS stable, and simulation errors will not aggravate. Figure 11 shows the DIM's stable impedance range. The exact $Z = Z_2$ value may be difficult to estimate without a proper typical hardware model. Such a model would make HIL simulation obsolete. Many studies imply that the DIM interface algorithm is the most accurate. Unlike previous approaches, DIM updates the damping impedance component to adjust to load variations [38], [39]. Modified Damping Impedance Method (MDIM) matches the load and damping impedance component dynamically to improve PHIL accuracy [35]. Modified DIM offers an adaptative approach to improve PHIL-based experiments. MDIM can track load impedance and match $Z = Z_b$. Hence, the updated damping impedance component is extremely load-dependent, as shown in Figure 12.

G. FEEDBACK CURRENT FILTER TECHNIQUE

FCF helps extend the IA's stable functioning ranges by adding band pass or low pass filters to remove harmonics and noise from the feedback current or voltage, as shown in Figure 13. Equation 9 shows the transfer function. The transfer function shows the system's low pass filter's dynamic influence on the ITM; thus, adding an FCF improves ITM's different frequencies f_C. Without a filter, the ITM stable area was violated at a ratio above 1, and with 1000 Hz FCF, the ratio is improved to above 3 [21]. The FCF improves every IA but can impair the signal exchange accuracy depending on the cut-off

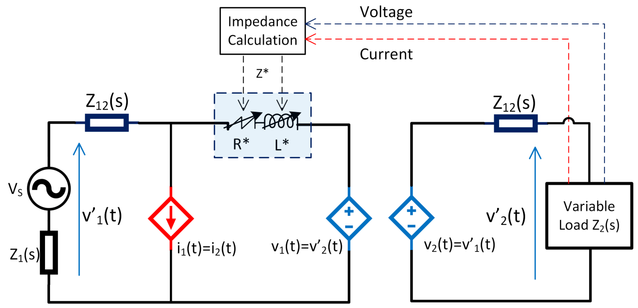

FIGURE 12. Adaptive and modified DIM.

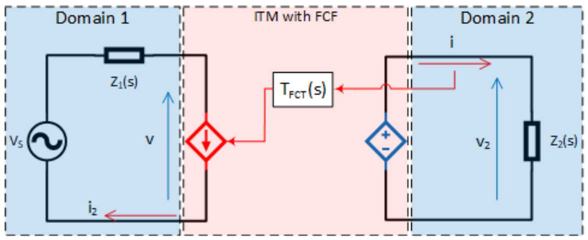

FIGURE 13. ITM interface with FCF scheme.

frequency f_C.

$$G_0 = T_{FCF}(s)\frac{Z_1(s)}{Z_2(s)}\, e^{s\Delta t} \tag{9}$$

H. SUMMARY OF INTERFACE ALGORITHMS

Table 1 shows the merit and demerits of the earlier discussed interface algorithms, while Table 2 shows the review of selected references on interface algorithms.

Digital PHIL simulators have been used in several works of literature to co-simulate with grid emulators and physical systems; however, due to the real interface's low voltage, sensors and amplifiers are needed to send signals at the power side. Providentially, for simulator-to-simulator data sharing, which is also applicable in GDRTCS, the communication network and interface simplify the interface of IA [41]. Thus, some challenges listed in Table 1 may not exist in simulator-to-simulator data sharing.

Most power equipment is connected to a PHIL real-time simulation using the ITM or DIM [39], [40]. DIM and ITM use a Low Pass Filter (LPF) on the Real-Time Simulator (RTS) output to stabilize the system when hardware like Photovoltaic (PV) inverters are connected. Furthermore, the simulation's discrete time step and LPF "phase shift" power amplifier voltage output inject synthetic reactive power into the simulation at the real-time simulator connection point. Consequently, a low-pass filter lead compensator can modify the phase shift to decrease synthetic reactive power injection in the simulation without losing hardware features [3]. Lastly, interface algorithms for subsystem connection must be accurate and stable. The interface techniques utilized for geographically dispersed real-time co-simulations are similar to those used for monolithic Contoller Hardware-in-the-Loop (CHIL) and PHIL simulations, but due to the geographic isolation of the infrastructures, they are slower and less stable [44]. Geographically dispersed real-time co-simulations are more challenging to partition than monolithic PHIL simulations with hardware coupled to a real-time simulator. Consequently, an optimum approach to partitioning a system for geographically dispersed simulation needs additional study [44].

III. INTERFACE SIGNAL AND COUPLING

A. INTERFACE SIGNAL TRANSFORMATION

GDRTCS IAs do not use instantaneous current and voltage waveforms like PHIL because Time-varying delays, packet loss, and reordering caused by a shared network for communication would degrade waveforms and introduce dynamics. Instead, GDRTCS IA transforms voltage, and current waveforms to available quantities exchanged between dispersed real-time simulators to solve this problem [18]. Thus, voltage and current signal equivalents that vary slowly with respect to time are generated and utilized in rebuilding respective time-varying equivalent signals at the receiving end. The sending end transforms voltage and current waveforms, the receiving end reconstructs them, and the controlled source of ITM receives the input [18].

1) ROOT MEAN SQUARE (RMS), FREQUENCY, AND PHASE ANGLE

This technique uses Root Mean Square (RMS), frequency, and phase angle to express interface quantities. Figure 14 shows the interface bus voltage waveform at subsystem 1, a transmission system decomposed into equivalent RMS, frequency, and phase angle. RMS value V_{TS} determined at the fundamental frequency before transferring interface quantities to participating real-time simulator simulating subsystem 2, a distribution network. Expected frequency f_{TS} is the next interface quantity in the co-simulation IA θ_{TS} is the phase angle between interface voltage and slack buses. Thus, three finite ordered lists of elements $[V_{TS}, f_{TS}, \theta_{TS}]$ represents distribution simulator interface quantities [45]. These signals at the interface are sampled at the transmitting rate and sent to a remote real-time simulator. The subsystem two simulators reconstruct time-domain regulated voltage values from $[V_{TS}, f_{TS}, \theta_{TS}]$ data [18].

$$v_a(t_k) = \sqrt{2}V_{TS}\sin(\varphi(t_k) + \theta_{TS}) \tag{10}$$

$$v_b(t_k) = \sqrt{2}V_{TS}\sin\left(\varphi(t_k) + \theta_{TS} - 2\frac{\pi}{3}\right) \tag{11}$$

$$v_c(t_k) = \sqrt{2}V_{TS}\sin\left(\varphi(t_k) + \theta_{TS} - 4\frac{\pi}{3}\right) \tag{12}$$

Simulation time is $\phi(t_k) = \int_0^{tk} 2_{TS}dt$. Signal reconstruction in Equations 10, 11, and 12 uses simulation time t_k. This method adjusts for phase shifts caused by system latency. It aligns the voltage at the transmission system's interface and the reconstructed waveform for the distribution system's regulated voltage source. Power balance preservation is critical for the interface's energy conservation. Correspondingly,

TABLE 1. Merits and demerits of interface algorithms.

IA	Merits	Demerits
ITM Algorithm	1) ITM Simple and high in accuracy without delay [24] 2) It does not affect the electrical conditions in the simulated circuit by adding elements [3] 3) This interface performs best in steady state error [34] 4) Stable [37]	1) Sensitive to initial conditions and hardware dynamics [34] 2) Has stability boundaries, limiting its applicability [3]. 3) Becomes unstable if partitioned circuit impedance criteria are unmet [3] 4) Poor simulation accuracy [37] 5) Vulnerable to the effects of time delay, which causes the system to become unstable [24]
TFA Algorithm	1) Performs best in steady state with minimized steady state error [34] 2) High precision in steady state [37] 3) Predicts and models the tested hardware to predict state variables	1) Complex in computation 2) Poor stability or instability [37] 3) Inaccuracy or low accuracy due to the vulnerability to sensor noise 4) Accuracy is limited for nonlinear systems and high-frequency signals; thus, it cannot implement them 5) Fails when nonlinear systems go through rapid dynamic changes [37] 6) Extremely noise sensitive, necessitating a configuration adjustment if the load changes from RL to RC [37]
TLM Algorithm	1) Highly stable [19] 2) Connects analog and digital simulators for real-time experiments 3) Possesses significant stability due to its trapezoidal approximation but has PHIL restrictions [22]	1) Hard to implement in high-power applications due to extra impedance replacement [19], which causes excessive power consumption [22] 2) Substitutes the L and C with Rt. This means excessive power consumption in high-power applications, which is often unacceptable [22] 3) Similarly, since Rt value relies on linking components, the simulated system must modify Rt. Thus, the system has low flexibility and expensive maintenance costs [22] 4) Interface's time delay Δt may depend on the load or frequency of the signal. Therefore, assuming a steady t reduces simulation accuracy [22]
PCD Method	1) Solves the simulation by splitting it into two subsystems, thereby saving computational resources [37] 2) Additional impedances or partial circuit duplication stabilize the interface to address instability [3]	1) Its precision is quite low
DIM	1) Hybridize ITM and PCD to stabilize the interface [3]	1) Damping impedance equals the hardware impedance for optimum value to avoid error propagation; however, the hardware's inadequate information makes it difficult to predict hardware's inadequate changes [37]
Feedback Current Filter	1) Stabilizes IA and improves stability while maintaining a compromise between accuracy and complexity [40]	1) Stabilization is always feasible if the filter cutoff frequency is low enough. Low cutoff frequencies stabilize the technique but delay the feedback signal, thereby reducing accuracy [40]

the measured currents on the distribution system interface bus are translated to the five finite ordered list of elements $[I_{DS,A}, I_{DS,B}, I_{DS,C}, f_{DS}, \varphi_{DS}]$ and passed to the real-time simulator for signal reconstruction for the controlled current source.

Figure 14 shows how co-simulation IA using the asynchronous method of coupling using RMS values of voltage as used in [46]'s IA connecting Idaho National Laboratory (INL) and National Renewable Energy Laboratory (NREL) submodels. This case study examines wind energy system grid integration. INL lab simulates IEEE 14 bus test system on RTDS, whereas NREL lab simulates wind turbine. Based on lab studies, a monolithic model was partitioned. The receiver reconstructs interface signals from RMS, frequency, and phase angle equivalent quantities [46]. A similar approach is employed in reference [47], where the RTDS simulator runs a modified balanced IEEE 13-bus in RSCAD and hardware under test connected at node 634.

2) DYNAMIC PHASORS

Dynamic Phasors (DP)s are sophisticated simulation tools based on time-varying Fourier quantities [48], [49]. As the size and quantity of new power system components increase, especially with the introduction of power electronics devices, simulation techniques are required to depict dynamic grid state changes without the computational cost of EMT simulations. Larger grids can be simulated and investigated under various conditions [4]. Dynamic Phasor Approach uses the Fourier series to estimate a complex time domain waveform $x(\tau)$ in the interval $\tau \in (t \smile T, \tau)$. Dynamic phasor representation of system quantities incorporates nonstationary situations [50].

TABLE 2. Review of selected references on interface algorithms.

References	Concept	Furtherwork for GDRTCS
Reference [42]	This study analyzes a Smith predictor (SP) compensator performance in eliminating system delay using the ITM interface model implemented on a real-time digital simulator platform.	A similar approach can be generalized and extended to other interface algorithms and configurations.
Reference [43]	The dynamic stability and accuracy of ITM and DIM using grid-support functions were tested using them to connect a PV emulator with the OPAL-RT-PHIL system. The grid-support functions include frequency-watt, volt-var, and power factor correction. For unity and negative power factor, the DIM low pass lead filter (LPF) was better; however, the ITM's LPF was better under positive power factor conditions.	Improve the DIM interface approach by incorporating a series RL in parallel with a series RC circuit that adaptively switches based on the hardware's operating regions and conditions.
Reference [3]	This study introduces a delay differential equation-based representation of the ITM. Thus, this work proposed a stabilization approach to improve the time-delay system's stability.	A similar approach can be generalized and extended to other interface algorithms and configurations.
Reference [37]	This study improves the accuracy and stability of DIM by evaluating the hardware's impedance and including the estimated value as part of the DIM	The delay time considered 180 microseconds is significantly small.
Reference [39]	This study compares and analyzes the accuracy of three interface algorithms (ITM, DIM, and modified DIM given in [37]) using simulation, experimental data, and FFT analysis.	Additional analysis and testing can be conducted for more dynamic and nonlinear loads to address the challenges of HIL at an increased power range.

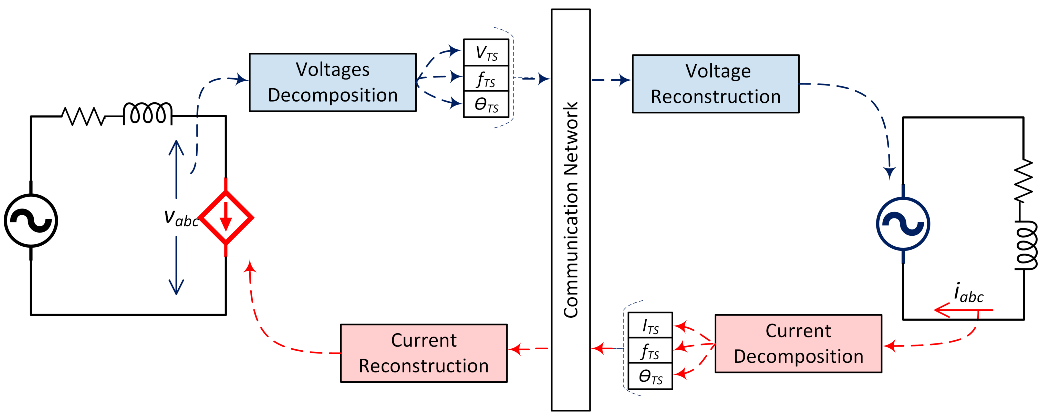

FIGURE 14. GDRTCS using Root Mean Square (RMS), frequency, and phase angle.

The concept is:.

$$x(\tau) = \sum_{k=-\infty}^{\infty} X_k(t) e^{jk\left(\frac{2\pi}{T}\right)\tau} \tag{13}$$

$$x(\tau) \approx \text{Re}\left\{\sum_{k\in K} X_k(t) e^{jk\omega\tau}\right\} \tag{14}$$

T is the system's period. Time-varying $X_k(t)$ is DP. The IA estimates the $k-th$ coefficient of DPs based on interface voltage or current waveforms. Where $\omega = 2\pi/T, X_k(t)$ is the kth complex form time-varying Fourier coefficient, and K approximates the waveform using specified Fourier coefficients [45].

$$X_k(t) = \frac{1}{T}\int_{t-T}^{t} x(t)e^{-jk\omega\tau} d\tau = \langle x\rangle k(t) \tag{15}$$

Four finite ordered lists of elements $[X_0, X_1, X_2, X_3]$ representing dynamic phasor coefficients are delivered to the remote real-time simulator for each three-phase. The controlled source's time-domain estimate is reconstructed based on interface quantities at the receiving end. As shown in Figure 15, the dynamic phasors method of decomposition of the signal can help overcome delay and integrate frequency into signals [4], [51], [52].

Figure 15 shows the GDRTCS idea with a co-simulation interface algorithm and signal exchange dynamic phasor domain. Subsystems are time-simulated. Depending on application and precision, transformation to the DP domain can involve more DP coefficients than basic components. Phase shift compensates for time-varying delay in DP-based co-simulation IA [53]. Furthermore, DP allows the integration and co-simulations of two or more subsystems

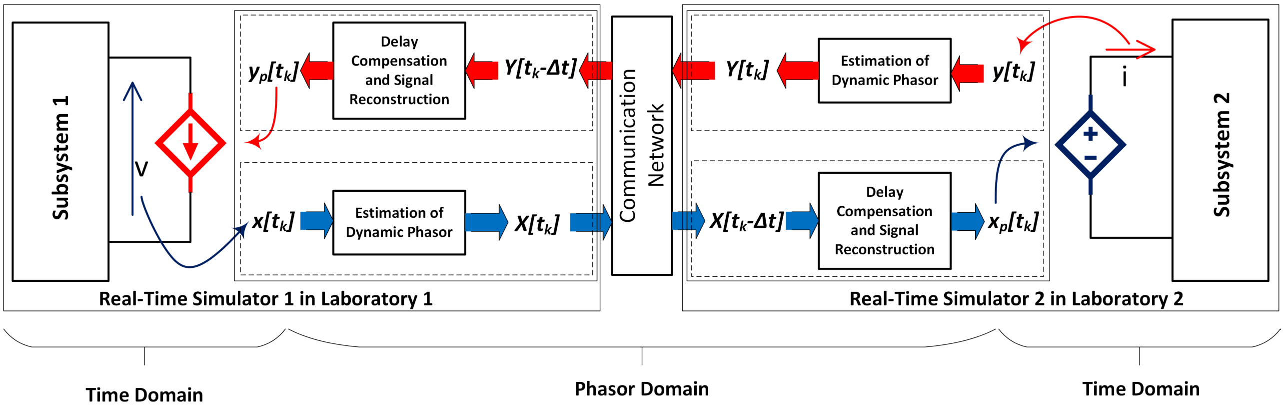

FIGURE 15. Dynamic Phasors GDRTCS IA.

simulated in different domains. In reference [45], a subsystem is the dynamic phasor domain, while the second subsystem is simulated in the electromagnetic transient domain.

In [45], a complex system is partitioned into two subsystems, with Subsystem 1 simulated in DP and Subsystem 2 simulated in EMT. Consequently, a line-to-ground unbalanced fault at a particular time and point within the EMT subsystem which disturbed the entire network. Different dynamic phasor extraction techniques acquire the current at the interface to test each method's accuracy [45]. The EMT simulator simulates fast transients like fast-response control and power-electronic switches; these segments in a model demand detailed representation and commensurate simulation time steps. Consequently, the dynamic-phasor solver can depict the rest of the model, where quick transients are insignificant, or their re-presentation can be avoided for computational advantages [50].

3) COMPARISON OF RMS AND DYNAMIC PHASORS IN GDRTCS

Two methods are considered for representing interface quantities in GDRTCS, and the Phasor-based simulation is more accurate than RMS as given in reference [18] where two racks of real-time simulators are used for transmission-distribution co-simulation. Both IA co-simulation approaches provide simulation fidelity, and the time delay must be compensated. However, simulation fidelity can be considerably degraded under transients, leading to inaccurate simulation results. In [18], the fidelity was high under normal testing but considerably worsened under transients.

DP-based GDRTCS IA naturally compensates for time delay with a phase shift. Literature has studied phase-shift-based correction of GDRTCS's much more significant time delay. Reference [49] compared stability for GDRTCS IA based on time domain waveforms and dynamic phasors in which the DPs approach provided better stability.

4) SYNCHRONOUS REFERENCE FRAME INTERFACE

RMS and Fourier coefficients-based dynamic phasors are used to exchange interface signals to eliminate delay impact on interface signal waveforms. Although phase information in RMS allows time delay compensation, system transients, and dynamics, the fidelity and waveform reproduction is impaired [18]. Phasor interface has these inadequacies: reduces response speed, phase shifts [54], and high computational complexity [55]. A Synchronous Reference Frame (SRF) interface is proposed in [19] to resolve the phasor interface drawbacks. SRF is a traditional transformation, but its assimilation into the IA and the consequent use to enable geographically dispersed real-time simulation present knowledge applicable to real-time co-simulation and PHIL setups. Furthermore, [19] claimed that the comparative performance evaluation with the DP interface reveals its superiority. Figure 16 shows two geographically dispersed real-time simulation experiment subsystems with a synchronous reference frame interface.

The interface algorithm determines the signals exchanged in the given experiment of Figure 16; subsystem 1 outputs three-phase voltages, and subsystem 2 outputs three-phase currents. As time-domain signals are unsuitable for direct packet-based transmission [49]; thus the interface signals must be transformed. Time-delay compensation is required to choose the interface signal transformation method [56]. The method used at one end determines the method used at the other. In [19], Park's transformation ($dq0$) decomposes signals. The matrices in Equations 16 and 17 give the SRF transformation of three-phase voltage and current signals in Figure 16.

$$\begin{bmatrix} u_d \\ u_q \\ u_0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \cos(\omega t) & \cos(\omega t - \frac{2\pi}{3}) & (\omega t + \frac{2\pi}{3}) \\ -\sin(\omega t) & -\sin(\omega t - \frac{2\pi}{3}) & -\sin(\omega t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (16)$$

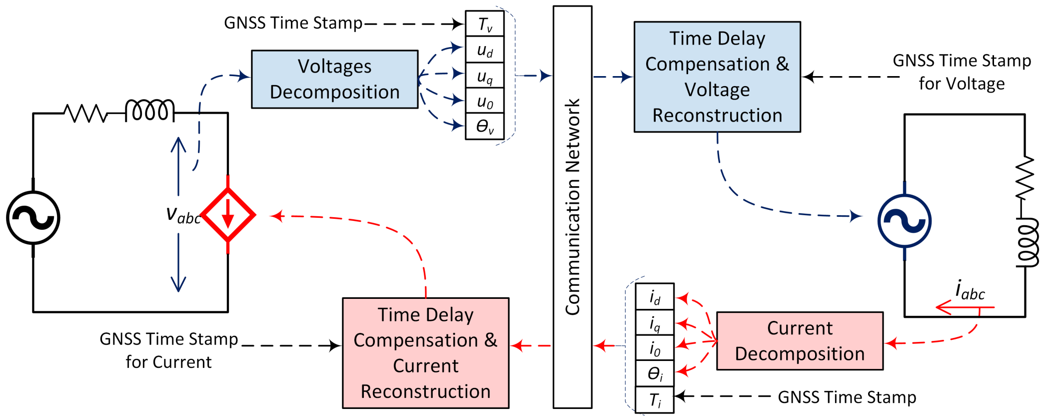

FIGURE 16. **Synchronous reference frame interface.**

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \cos(\omega t) & \cos(\omega t - \frac{2\pi}{3}) & (\omega t + \frac{2\pi}{3}) \\ -\sin(\omega t) & -\sin(\omega t - \frac{2\pi}{3}) & -\sin(\omega t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{17}$$

5) WAVE VARIABLE TRANSFORMATION

Teleoperation is remote device operation; teleoperation allows humans to remotely manipulate items by simulating remote conditions, as shown in Figure 17. The master, a manipulator or joystick at the human end, sends motion orders to the slave, doing the task. Generally, the human applies force to the master manipulator, causing the slave to move [57]. Bilateral teleoperation lets the user control the slave with a joystick or robot and feel the slave interaction force [58]. Thus, if the slave has force sensors, it can communicate or reflect response forces from the job to the master, which inputs the master's input torque and controls the teleoperator bilaterally [57].

Teleoperation's control-theoretic goals are [57];

- Stability: Ensuring closed-loop system stability regardless of operator or environmental behavior.
- Telepresence: Providing a sense of telepresence to the operator, which is system transparency between the operator and the environment.

Wave Variables (WV) Transformations are presented for bilateral teleoperation to guarantee stability during time delays. Passivity formalism uses wave variables [53]. Bilateral teleoperation uses it to transfer communication, not power, between environment WVs and the operator. Figure 17 shows the force and velocity F_m and $\dot{x}_m$ at the human operator end and F_s and $\dot{x}_s$ at the environment terminal.

The wave variables (u_m, u_s, w_m, w_s) are obtained by the wave transformation that is given by Equations 18, 19, 20, and 21. Characteristic impedance is given by parameter b; b is not necessarily physical impedance to be stable, but it is

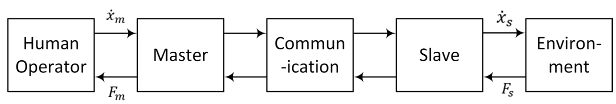

FIGURE 17. **Principle of teleoperation.**

crucial for obtaining a required response.

$$u_m = \frac{1}{\sqrt{2b}}(F_m + bx_m) \tag{18}$$

$$u_m = \frac{1}{\sqrt{2b}}(F_m + bx_m) \tag{19}$$

$$u_m = \frac{1}{\sqrt{2b}}(F_m + bx_m) \tag{20}$$

$$w_s = \frac{1}{\sqrt{2b}}(F_m + bx_m) \tag{21}$$

Equations 22, 23, 24, and 25 relate wave variables to power variables which determine each terminal's wave and power variables. Figure 17 shows the system with wave variables if $\dot{x}_m$ is defined on the master side and F_s is given on the slave side.

$$F_m = \sqrt{\frac{b}{2}}(u_m + w_m) \tag{22}$$

$$F_s = \sqrt{\frac{b}{2}}(u_s + w_s) \tag{23}$$

$$x_m = \frac{1}{\sqrt{2b}}(u_m - w_m) \tag{24}$$

$$x_s = \frac{1}{\sqrt{2b}}(u_s - w_s) \tag{25}$$

Transferred wave variables are affected by time delay, as shown in Equations 26 and 27.

$$w_s(t) = u_m(t - \tau) \tag{26}$$

$$w_m(t) = u_s(t - \tau) \tag{27}$$

Using the velocity-current- and force-voltage analogy, three-phase waveforms were transformed into wave variables to

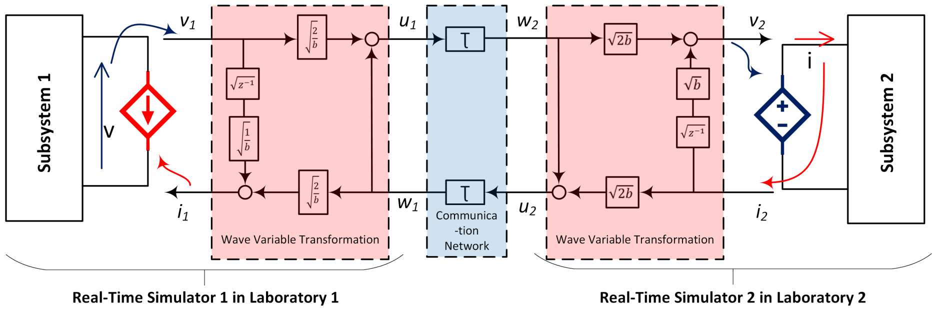

FIGURE 18. **Wave Variable Transgormation GDRTCS.**

improve geographically dispersed real-time co-simulation stability in the presence of delays for IA. Reference [53] showed that this transformation ensures stability in simulation even when ITM is unstable by analyzing the fidelity of a wave variable system with deterministic large delays. Consequently, both GDRTCS and bilateral teleoperation approaches seek to synchronize two geographically distant and decoupled subsystems to maintain the system's accuracy, reliability, and stability.

Wave Variables and ITM: As wave variable transformation ensures stability, wave transformation is of great interest in geographically dispersed real-time co-simulation IA. Figure 18 shows wave-based distributed simulation architecture. The wave transformation is modified by comparing current to velocity and voltage to force. Interface quantities (voltage and current) transformation into wave variables improve simulation stability and accuracy for small time delays; however, simulation fidelity deteriorates this higher delay and renders the results unacceptable [53].

In [59] as shown in Figure 19, WT-based GDRTCS IA is utilized in an High-Voltage Direct Current (HVDC) link with a DC cable partitioning point where ITM is unstable for high time delay; wave transformation provides simulation stability [59].

Wave Variables and Dynamic Phasors: In [53], the simulation fidelity degrades significantly with a high delay and beyond. In reality, for example, Europe's geographically dispersed real-time co-simulation delays can reach 50 ms. Wave variables provide simulation stability for large delays, which motivates improved IA design for fidelity. Reference [46] proposed using dynamic phasors to improve co-simulation IA using wave variables. Figure 20 shows WV conversion to DP for sending and DP coefficients reconstruction on reception, while wave and power variables do not change [53]. Further Simulation fidelity analysis with higher delay in [53] shows that delays have not affected simulation stability; steady-state simulation fidelity is high, but transients are lower.

The process of GDRTCS commences by dividing a large network into smaller, manageable subsystems.. The partitioning and interface technique influences simulation accuracy, speed, and stability [7]. For precision, three-phase signals measured at the interface are decomposed and sent as transformed signals (RMS, DP, or dq0) rather than time-domain signals in real-time. Real-time simulators have communication interfaces that enable data exchange via a gateway server, converting and managing data in various protocols from various sources and destinations, enabling compatibility between multiple simulators in GDRTCS. The remote side then reconstructs signals and provides a delay compensation and synchronization. Lastly, the reconstructed signals are introduced into the distributed subsystem [41].

B. VIRTUAL INTEGRATION COUPLING POINT SIGNALS

Real-time co-simulation involves simulating a complicated system model employing two or more diverse, interconnected simulations divided into many simulators. For example, two types of couplings are used in the power systems sector for Geographically Distributed Real-Time Co-simulation [60].

- Control Signals Coupling.
- Electrical Signals Coupling.

Authors in study [61] created a framework to study how coupling point selection affects distortion and the partitioning requirements and design considerations that determine the choice of a coupling point. Furthermore, reference [61] proposes system parameters that make a coupling point location with low distortion in GDRTCS.

Control Signals Coupling: This coupling is utilized for exchanging control signals between two or more GDRTCS systems. This type of coupling is utilized, modeled, and illustrated by [62], [63], [64], and [65]. When two separate domains are coupled, as in the thermo-electrics, there is no concrete simulation coupling, and the coupling is comprised exclusively of control signals. In thermo-electric co-simulation, the electric subsystem is modeled differently from the thermal subsystem. Similarly, control signals coupling involves the exchange of control signals between the simulation models representing hardware and software components; these control signals typically represent commands, setpoints, or other instructions that control the behavior of the simulated system. However, the overall simulations are connected to share feedback signals synchronously or asynchronously [15].

In the context of study [66], the primary contrast between CHIL and PHIL is the coupling type between the simulated

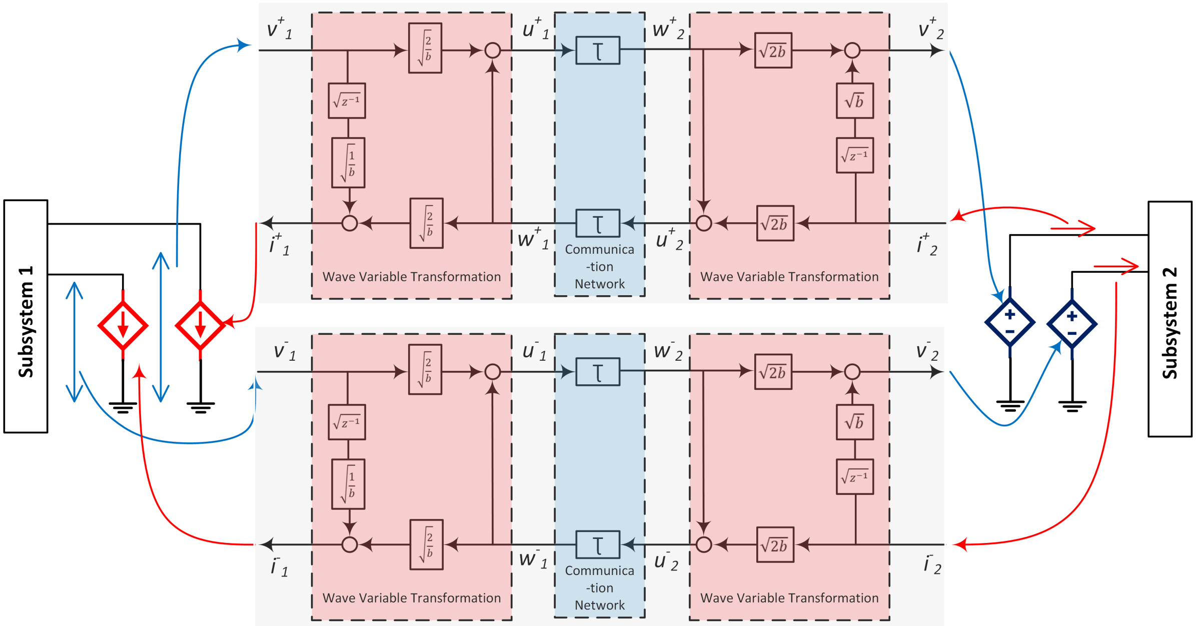

FIGURE 19. Wave Transformation for ITM of HVDC interface.

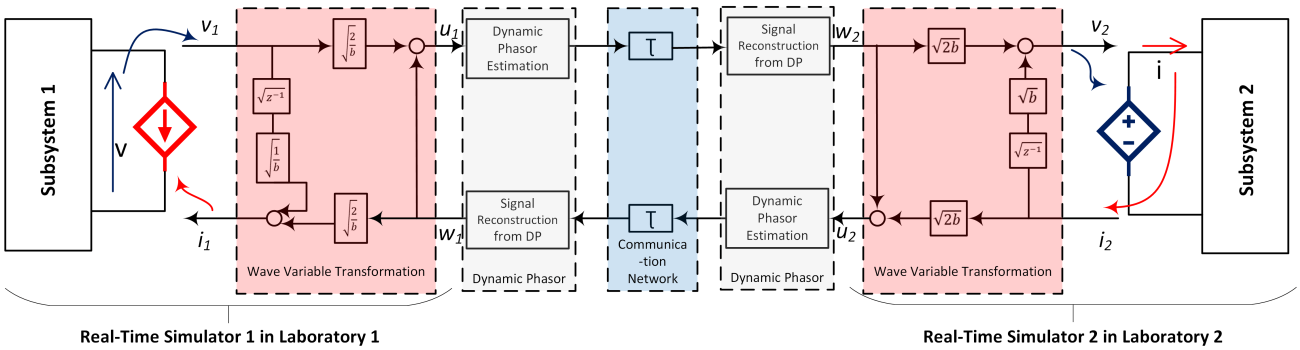

FIGURE 20. Wave transformation with Dynamic Phasor.

virtual system and the hardware device under test. In the HIL testing scenario, only low-power or control signals (digital control signals and encoder signals) are transmitted between software (simulator) and hardware (for example, external controller). In the PHIL, on the other hand, a natural coupling is enabled, resulting in the power exchange between the hardware device being tested and the virtual simulated system.

For real-time cyber-physical systems, a control signal is at the interface of the two dissimilar simulation systems (for example, power and communication) in the microgrid co-simulation testbed [67]. This work provided a validation platform for the microgrid control schemes and cyber incidents' impact on microgrid operation. References [68] and [69] used a similar real-time co-simulation method for power systems and communication network emulation on the power and communication simulators.

Electrical Signals Coupling: This coupling refers to the coupling that occurs when real-time simulators exchange electrical signals and is an example of natural coupling [66]. Natural coupling is the interconnection of physical systems with a strict energy conservation requirement. Electrical signals coupling focuses on the exchange of electrical signals between the simulation models. These electrical signals represent the dynamic behavior of the electrical or electronic components being simulated [29]. The three forms of electrical signal coupling and exchange are asynchronous AC coupling and synchronous AC coupling [60], [70].

Asynchronous AC Coupling: When the current and voltage phase angles at the coupling point of the two power subsystems are not out of phase, asynchronous AC coupling is present. Thus, voltage and current phase angles are not synchronized with respect to time in GDRTCS [46], [71]. In [72], two distinct subsystems, each corresponding to 50 Hz and 60 Hz distribution network models in Australia and the United States, were utilized to examine the effects of synchronized inverter management. In addition, asynchronous AC coupling was used in reference [46] to establish a virtual link between real-time simulators at INL and the NREL for wind turbine characterization and wind power large-scale grids integration assessment. Figure 21 shows this approach is a much simpler alternative to synchronous coupling. In asynchronous coupling, the feed-forward loop facilitates the transfer of the RMS value and frequency of each phase of the three-phase voltages from Subsystem 1 to Subsystem 2 [73].

Because the two subsystems are not time-synchronized, Subsystem 2's real-time measured current cannot be injected into Subsystem 1; however, Subsystem 2's measured power (active and reactive) is sent to Subsystem 1 and injected as current after re-estimation using the coupling Subsystem 1's voltage. Thus, the feedback loop uses the coupling voltage's phase angle in Subsystem 1 rather than that of currents from Subsystem 2 without providing time delay compensation [73].

Synchronous AC Coupling: This coupling exists when two subsystems are split at an AC line, and voltage, current, and phase angles are synchronized to time. However, the GDRTCS framework does not guarantee time and start-up synchronization between two real-time simulators [74]. Thus, there is a relative time shift between results logged in each real-time simulator, as the signal reception depends on each simulator's absolute time relative to simulation commencement. Thus, simulators track data with a time shift relative to another simulator's absolute time perspective [74].

The synchronous reference frame shows that the time-varying current (feedback loop) and voltage (feed-forward loop) are transformed into equivalent DC quantities in the Synchronous Coupling approach. In addition to exchanging these equivalent DC quantities between subsystems, this technique necessitates the exchange of phase angle between subsystems to achieve precise signal reconstruction with compensation for delay and time-synchronization [73].

DC Coupling: DC coupling exists when two power subsystems are split across a DC line coupling point. In [12], Eight geographically dispersed laboratories executed a complete United States-European transatlantic HVDC interconnection simulation experiment, which serves as a benchmark for transmission-distribution interactions and an evaluation of the grid support capability of distributed energy resources. This author [75] concentrates on geographically dispersed real-time simulation experiments of an HVDC link with a DC cable partitioning coupling point.

IV. COMMUNICATIONS AND FRAMEWORK

The ability to communicate over long distances between geographically separated assets distinguishes monolithic HIL simulations from geographically dispersed real-time simulations [44]. Communication refers to data exchange between interconnected simulators, which is typically handled by telecommunication protocols [30]. The communication medium must facilitate data integration and exchange across geographically dispersed real-time co-simulation. Fiber optics is preferred as a dedicated communication medium; however, fiber optics as a dedicated communication medium is expensive over great distances. Thus, the Internet is the most appropriate communication platform for conducting such experiments [5]. In GDRTCS, the physical limits of the communication network and the intrinsic communication protocol features become crucial considerations that impact the simulation's accuracy and stability.

A. NETWORK

Internet is used to exchange interface signals between real-time simulators performing geographically dispersed co-simulation experiments. The associated latencies depend on the technique or mix of technologies used to connect the real-time simulators. The estimated round-trip latency between European RIs is around 50 milliseconds, limited by the following physical restrictions [56]: the light propagation in optical fiber and real-time simulators' location distances.

GÉANT connects European national research and education networks (NRENs), allowing partnership on research initiatives with a 50,000-km high bandwidth with increasing daily services. In addition, these features let researchers cooperate remotely and use roaming services like Eduroam. Eduroam is a free worldwide Wi-Fi roaming service for education consumers. It gives visiting researchers, lecturers, and students network access.

1) ESnet

The Energy Sciences Network (ESnet) serves Department of Energy (DOE) scientists and their global collaborators [77], [78]. This network connects more than forty DOE research centers and laboratories, promoting global collaboration. ESnet 5, a 100 Gbit/s network by ESnet with a coverage map shown in Figure 22, was launched in October 2011. ESnet has teamed with GÉANT to enhance ESnet's network capacity in Western Europe to 200 Gbit/s to ensure that this data reaches scientists worldwide as quickly as possible. In addition, this upgraded network enables rapid scientific collaboration between researchers on both continents. The coverage map of this intercontinental coverage is shown in Figure 22.

2) Internet 2

The Internet2 Network [80] is an IP network employing fiber optics that provides education and research network services. Internet2 is designed to meet the bandwidth-intensive needs of US research and education. The network is dynamic, strong, and cost-effective. ESnet teamed with Internet2 to connect research centers and universities in the United States.

B. COMMUNICATION PROTOCOL

Protocol as a set of rules lets two or more units of a communication system exchange data using any physical quantity variation. The protocol sets the rules, semantics,

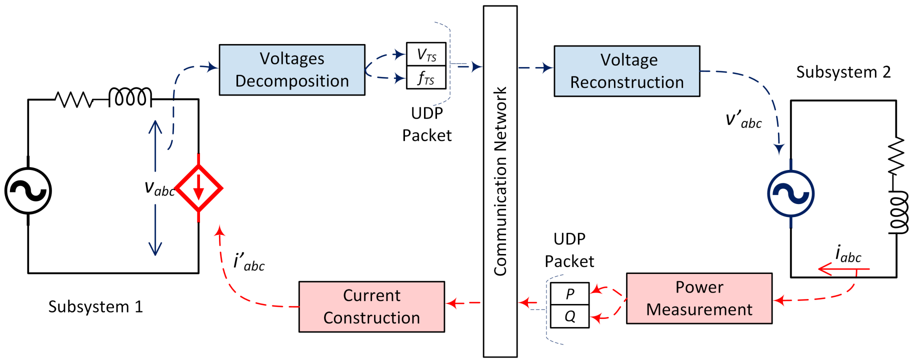

FIGURE 21. Asynchronously coupled GDRTCS.

TABLE 3. Summary of coupling used in real-time co-simulation.

Coupling	Application	Remarks
Control Signals	Signal exchange between dedicated simulations/simulators	Only applicable to two dissimilar subsystems
Asynchronous AC Electrical Signals	Model split across an AC line where voltage or current and phase angles are not synchronized to the time.	Useful in the case of a high communication delay and a slow data exchange rate. Limited to steady-state testing and slow dynamic study.
Synchronous AC Electrical Signals	Model split across an AC line where voltage or current and phase angles are synchronized to the time.	Capable of transients and dynamic simulations. However, system dynamics and transients shorter than the time delay may not be replicated accurately.
DC Electrical Signals	Model split or portioned across a DC line (for example, HVDC) for simulation.	Only applicable to DC link between two subsystems

TABLE 4. Examples of network delays in different communication network applications [76].

Communication Networks	Network Delays
Wired/wireless domestic data transmissions in commercial	25 to 110 ms (round-trip)
Intercontinental data transmissions	order of 270 ms
Image data transmissions	700 ms
Intercontinental 4G networks	up to 3 s (round-trip)

syntax, synchronization, and error recovery procedures for communication. Hardware, software, or a combination of both may be used to implement protocols [81]. For example, Transmission Control Protocol/Internet Protocol (TCP/IP) and User Datagram Protocol (UDP) are protocols used in co-simulation. Therefore, most efforts focus on improving Transmission Control Protocol (TCP) and TCP's relationship with the IP [82]. Choosing the right protocol is critical to creating an accurate and dependable co-simulation infrastructure capable of ensuring numerical solution stability. The main issue is usually caused by latency (or lag) [30].

1) TCP/IP PROTOCOL

TCP/IP is a sophisticated Internet communication protocol that specifies a standard connection approach and data transfer

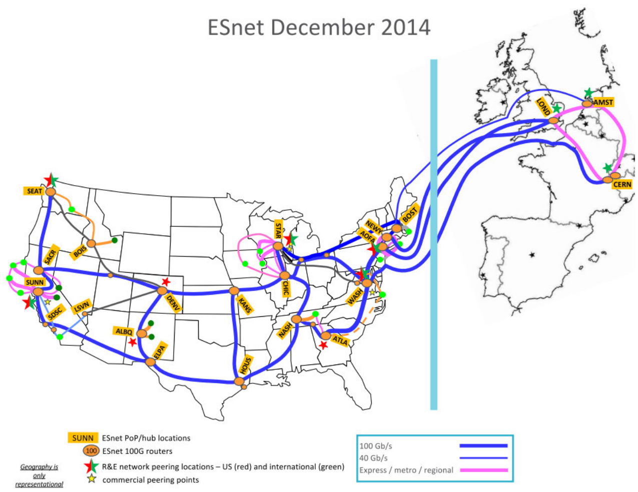

FIGURE 22. ESnet Coverage map and connection to Europe [79].

between electronic devices and the internet [77]. TCP is one of the fundamental protocols for the internet. It originated with the initial network construction and was designed to complement IP. Consequently, the full suite is often known as TCP/IP. TCP enables dependable, well-ordered, and error-free transfer of an octet (byte) stream between software applications operating on a host PC which is interacting with another

Internet network-based IP application on TCP, a TCP/IP suite's Transport Layer component. Before sending data, TCP must establish a connection connecting the server to the client, and before connecting, the server must be passively open (listening). Error detection, retransmission, and three-way handshake increase dependability; however, these lengthen latency. Reference [83] shows OPNET's and RT-LAB's real-time co-simulation structure where TCP/IPs are employed as the data interaction interfaces between the RT-LAB and OPNET, where Ethernet sockets communicate power grid status and control command information.

The innovative technology in [84] uses TCP/IP to co-simulate track dynamics and vehicle simulators. Similarly, the interaction interface in [67] uses TCP/IP for data transmission, coordinating three RT-LAB modules with OPNET's SITL module. The three modules that control data transmission in [67] are OpIPSocketCtrl, OpAsyncRecv, and OpAsyncSend.

2) UDP PROTOCOL

UDP is a key Internet protocol that conveys messages (datagrams in packets) to other hosts on an internet protocol network. UDP does not need prior communication in setting up IP data pathways or channels. UDP is a connectionless protocol with few mechanisms for protocol. This protocol has the ability to provide checksums for data integrity and port numbers for addressing source and destination functions. Consequently, there are no handshaking dialogues in UDP, and this causes network unreliability as ordering, delivery, and duplication protection are not guaranteed. Any application needing error correction at the network interface level can use TCP or a computer networking communications protocol called Stream Control Transmission Protocol.

UDP avoids the protocol stack overhead of error checking and correction when it's not needed or done in the application. For example, when time sensitivity like a real-time experiment is required, UDP is employed due to its packet discarding ability, which is preferable to waiting for retransmitted packets [85]. Reference [86] describes the first observer-free integration of two geographically distributed HIL experiments (engine-in-the-loop simulation) for an automotive application.

In study [86], there are two setups that exist for co-simulation purpose. One is the engine-in-the-loop simulation setup located at the University of Michigan in Ann Arbor, Michigan, United States. The other setup is the driver-in-the-loop ride motion simulator situated at the US Army Tank-Automotive Research, Development and Engineering Center (TARDEC) in Warren, Michigan, United States. In this framework, the arrangement at TARDEC serves as a client, while the UM arrangement serves as a server; whether or not it receives a response, the client transmits an updated transmission speed and throttle signal via the UDP at a rate of 20 Hz. UDP indicates that packets may be lost or out of order when it arrives; however, the delay is minimized, which is more important for GDRTCS. On the other hand, the server only provides an updated transmission torque on getting packet data from the client. Therefore, all packets sent are stamped with the time; thus, only the latest packets are utilized, with older packets that arrive after the time stamp being ignored. This strategy assures that both parties respond to the most recent available signals [86].

TABLE 5. Summary of comparison of UDP and TCP.

Item	TCP	UDP
Overview	Connection-oriented protocol	Connectionless protocol
Reliability	Reliable	Unreliable
Orderliness	Ordered	Not ordered
Weight	Heavyweight	Lightweight
Reading	Streaming	Datagrams
Congestion	Congestion control	No congestion control
Broadcasting	No broadcast	Broadcasts
Multicasting	No multicast	Multicast

3) COMPARISON OF UDP AND TCP

As a stateless protocol, UDP offers a connectionless datagram service that promotes time over dependability. On the other hand, as a connection-oriented TCP avoids congestion but has drawbacks such as service denial, hijacking, reset attacks, and TCP veto. Moreover, TCP requires establishing a physical connection before transmitting data, unlike UDP, which requires none for transmission. Therefore, the upper layer of the UDP protocol is equipped with the ability to send data at any required time with a high speed and inherent ability to unicast, multicast, and broadcast. Thus, UDP is ideal for situations with a large number of terminals, small amounts of data to transmit, and real-time requirements [87].

In UDP, each interface sends packets to the remote interface at a predetermined rate, whether remote laboratory packets are received or not. This prevents excessive waiting and guarantees that one-way communication issues do not influence the other. In addition, UDP transfers data between laboratories. Furthermore, UDP is preferable and better for real-time simulation because they have lower delay variation than TCP [54], [88]. In addition, UDP is better for real-time simulation than TCP, as TCP increases latency and jitter by retransmitting missed packets and acknowledging received packets [49].

4) REAL-TIME TRANSPORT PROTOCOL (RTP) AND RTP CONTROL PROTOCOL (RTCP)

Real-time Transport Protocol (RTP) operates over UDP to transmit video and audio across internet networks; thus, it is suitable for entertainment and a communication system. In addition, RTP is utilized alongside the RTP Control Protocol (RTCP), where RTCP observes transmission statistics and quality of service (QoS) and assists in synchronizing numerous streams. Furthermore, RTP and RTCP are sister protocols employed concurrently for data exchange between gateway nodes [55], where RTCP is utilized in exchanging QoS reports that measure delay, packet loss, and jitter, whereas RTP handles data transfer [55].

Synchronization, a timed coordination of distributed system events, is required in GDRTCS; however, the synchronization data is signaled over the same network as transferred data, which is influenced by latency. Therefore, this latency must be accurately predicted and compensated to ensure synchrony in participating real-time simulators. Network time protocols such as Inter Range Instrumentation Group Timecode B (IRIG-B), Precision Time Protocol (PTP), and Network Time Protocol (NTP) mitigate this challenge by applying techniques like Christian's or Best Master Clock (BMC) [9].

The IRIG-B is a standard timecode format commonly used in various industries, including aerospace, defense, and telecommunications [89]. IRIG-B is designed to provide accurate and synchronized time information across multiple systems and locations. IRIG-B timecode consists of a binary-coded decimal (BCD) representation of time, including hours, minutes, seconds, and sometimes milliseconds. The timecode is transmitted using Amplitude Modulation (AM) or other suitable methods [90].

The PTP is used to synchronize clocks in computer-based systems with high precision. PTP is designed to achieve sub-microsecond level synchronization accuracy, making it suitable for applications that require extremely precise time synchronization [91]. PTP operates by exchanging timing messages between devices on a network, allowing them to establish a common time reference. It uses the IEEE 1588 standard for clock synchronization, which defines the PTP protocol and message format [92].

The NTP is used for time synchronization in computer-based systems. NTP enables devices on a network to obtain accurate and synchronized time information from reference time sources, such as atomic clocks or other highly accurate time servers [93].

In the setup shown in Figure 23, VILLASnode collects communication connection statistics and protocol translation between UDP used for connecting to RTDS' GTNET card and RTP deployed between the laboratories; an example is implemented in Reference [55].

C. TRANSMISSION RATE

The transmission rate is the interface signals transferring rate between geographically dispersed real-time simulators. Depending on the interface card capabilities of the geographically dispersed real-time simulators, the rate of transmission could be the same with time step or below it in simulation [60]. A fixed transmission rate uses the same sending and receiving rates. However, due to packet loss and no retransmission, UDP's receiving rate is lower than its sending rate.

The interfaces transfer time-sensitive simulation data in geographically dispersed real-time co-simulation between laboratories, necessitating reliable communication connectivity. However, the predictable behavior of connectivity over a shared communication network is not easy to guarantee since a dedicated communication infrastructure is not always available. Interface drops duplicated and reordered packets, buffers packets in order to reduce delay variance, adjust data rate to and rate from simulators and collects communication link information. It also includes a timestamp in data packets and allows random data processing. Regardless of whether remote laboratory packets are received, each interface sends packets to the remote interface at a predetermined rate. This eliminates waiting and prevents communication issues from affecting the other side. Interfaces use stateless protocols like UDP [49].

One of the factors influencing simulation results is the frequency with which simulators exchange their interface signals. The maximum rate is limited by the communication links' available bandwidth. Because other users share the link, the obtainable bandwidth varies over time, potentially causing congestion. Congestion must be prevented because it causes packet loss and a reduction in simulation fidelity. Study [55] implements congestion prevention based on an RTP that dynamically modifies the sending rate to improve this situation.

V. FRAMEWORKS

The co-simulation framework for the virtual connection of geographically separated real-time simulators allows concurrent simulation of subsystems of a complex model using available software and hardware. Geographically dispersed real-time research infrastructures utilize a co-simulation framework to connect participating real-time simulators. Frameworks such as VTB-RT, VILLAS (Virtually Interconnected Laboratories for LArge systems Simulation/emulation) Framework [54], [74], [75], [94], [95], [96], and HELICS (Hierarchical Engine for Large-scale Infrastructure Co-Simulation) have provided platforms for integrating geographically dispersed laboratories worldwide. For example, VTB-RT is proposed in reference [66] as a modular experiment platform for virtual test beds and real-time simulation.

1) VILLAS FRAMEWORK

VILLAS connects test beds and real-time simulations in remote laboratories. RWTH Aachen University developed VILLAS for regionally dispersed electrical networks RT simulation, and it can still be deployed in other domains of simulation and multi-physical model simulation [97]. HIL simulations require real-time electromagnetic transient models. This requires customized real-time simulation targets to accommodate time steps from 1 to 50 s. The computational restrictions of a single target prevent scaling such simulations to thousands of nodes. VILLAS clusters have proven the possibility of connecting several distributed real-time simulators over the Internet to bypass this constraint [12].

The framework's independent components can be integrated depending on requirements and functions. Several components form the architecture of VILLASnode for arbitrarily large

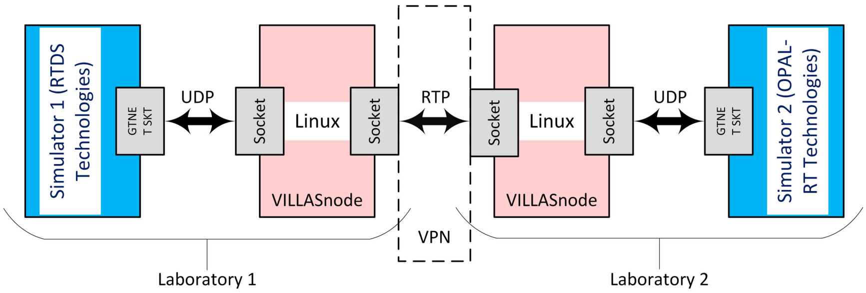

FIGURE 23. VILLASnode customized for the RT simulation data exchange in different formats and protocols.

co-simulation installations. The framework architecture is shown in Figure 24.

VILLASnode connects the components and receives simulation and measurement data. In addition, it provides interfaces to RT simulators or targets and protocols adaptors that enable real-time data exchange. VILLAS framework, unlike HELICS, is a modular and decentralized framework that speeds up simulations without a central scheduler and broker. The VILLASnode gateways connect simulators to other simulators and gateways. Complex co-simulation without a central Internet can be developed [12]. VILLASweb allows remote simulation monitoring through application software for accessing websites and has a graphical control panel for measurement, results, and gauges, which can be configured to suit researchers using real-time WebSockets. The simulated experiment can be interacted with via this web interface. The VILLAScontroller manages the included laboratory infrastructure's setup, control, and inventory by exchanging the configurations, current status, and control commands between the virtual control center and the participating laboratories' real-time simulators and hardware devices. A schematic of the VILLASframework co-simulation platform is given in Figure 23.

Study [49] deployed the VILLAS framework, allowing flexible and modular integration of real-time simulators in Europe. The platform also provides web browser access for third parties. This study uses dynamic phasors to represent interface quantities in a co-simulation interface technique. Phase shift compensates for the time delay between real-time digital simulators to simulate slow transients accurately. The architecture integrates laboratories at German universities, an Italian polytechnic, and EU Joint Research Centres in Netherlands and Italy. In addition, this study co-simulates transmission and distribution networks using a virtual laboratory framework.

In study [49], a pan-European RT co-simulation is initiated with the interconnection of four universities' laboratories, to examine voltage and frequency control schemes with renewable energy integration. RWTH Aachen University, University Politehnica of Bucharest (UPB), Politecnico di Torino (POLITO), and University College Dublin (UCD) laboratories use the networked real-time simulation infrastructure to complete the task.

2) HELICS

The Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) design concepts is discussed in [98]. HELICS is also a scalable open-source cyber-physical framework for real-time co-simulation that integrates transmission and distribution dynamic simulators to simulate local, national, and interconnection-scale electric power and energy systems at a given speed [98]. It simulates electric power system characteristics by integrating transmission, distribution, and communication simulators. In addition, it uses a broad data exchange interface to incorporate several energy and power real-time simulators [99]. A typical HELICS co-simulation architecture is in [100]. Each core of HELICS has one integrated executable federate on the same machine and is connected to the same broker. Small federates and co-simulations under development use this design.

Federates exchange physical and control signals while simulating subsystems. Brokers keep several federates synchronized and facilitate message interchange. Federates, also known as simulators, include transmission ANDES and distribution OpenDSS simulators. Federates exchange co-simulation messages. Subscriptions or federate-to-federate endpoint connections enable message interchange. The co-simulation can replicate cyber-physical interactions by including the filter determining communication delay or packet loss in endpoint communications. HELICS can co-simulate many federates using existing power and communication system technologies. This research [99] proposes a HELICS-based framework for the co-simulating transmission and distribution dynamics for renewable energy-based distributed generator frequency response.

3) APPLICATIONS OF FRAMEWORK

The applications of VILLAS and HELICS frameworks in some selected relevant publications are summarized in Table 6. In summary, VILLASnode is used to co-simulate power system components such as generators, transformers, transmission lines, and loads. It enables the analysis of system behavior, stability, and grid integration of renewable energy sources [101]. VILLAS enables the evaluation and testing of advanced control strategies for power system components. This includes testing coordinated control schemes, voltage

TABLE 6. Application and implementation of VILLAS and HELICS framework.

References	Framework	Application and Implementation
Reference [101]	VILLAS	Developed an architecture for a federated cyber-physical testbed consisting of an OPAL-RT and RTDS using the VILLAS framework for end-to-end testing.
Reference [111]	VILLAS	A distributed co-simulation environment illustrates the potential and limitations of cooperative real-time modeling of local energy communities (LECs) over great distances utilizing the VILLASnode interface and a broad range of supported protocols.
Reference [99]	HELICS	This research introduces an effective open-source transmission and distribution dynamic co-simulation framework based on the HELICS and commercially available transmission and distribution simulators for DER frequency response.
Reference [112]	HELICS	This research also presents the transmission and distribution dynamic co-simulation framework on HELICS for exploring the effect of electric vehicles on frequency response using open-source power system simulators (ANDES and OpenDSS).
Reference [113]	HELICS	This research employs a relatively small timestep (1 ms) for the transmission system and a significantly big timestep (5 ms) for the distribution system, and HELICS data exchange in multi-time steps co-simulation timesteps are not required to be identical. The simulation of these different systems starts simultaneously but runs separately on different CPU cores. HELICS synchronizes the time and guarantees that distribution feeder models precisely follow the transmission model with
Reference [104]	HELICS	The proposed scalable multi-timescale transmission and distribution co-simulation system leverages HELICS software in conjunction with Python as the interface that regulates timing and variable exchange between the T and D solvers. Through a Python instance, the HELICS interface can be accessed.

TABLE 7. Architectural layers of real-time co-simulation.

Layers	Details
Digital Real-Time Simulator	The infrastructure's core layer permits a bidirectional point-to-point link between two real-time simulators. Simulators are required to meet the following conditions: 1) support Aurora 8B/10B protocol with SFP interfaces for data exchange between real-time simulators, and 2) supplying a synchronization board equipped with an RJ45 interface to implement the IEEE 1588 Precision Time Protocol (PTP) stack. This configuration allows for the synchronization of internal reference clocks with the underlying layer [33] [114].
GPS Synchronization	The GPS synchronization layer employs a GPS clock to synchronize its internal reference clock with GPS atomic clock technology to create the IEEE1588 PTP master node. The infrastructure can access atomic time standards without a local atomic clock, ensuring worldwide synchronization. Since DRTS racks evolve their simulation with an independent time-regulating schema, the time-regulation schema can be ignored. The source section of the proposed source-load circuit will start the load simulation. Since the IEEE1588 PTP stack ensures the right time synchronization schema, DRTSrack results are aligned using the internal clock time [33] [114]. DRTS requires synchronization to inject signals at the right time and phase. Remote subsystem synchronization using GPS signals is frequent. GPS signals ensure subsystems start their simulations simultaneously [41].
Power system co-simulation	This layer uses the ITM or any advanced IA to separate a power system model into two subsystems. This IA, used in PHIL applications, uses a controlled voltage generator in the second subsystem to recreate the voltage measured in the first subsystem and a current generator in the first subsystem to reproduce the current measured in the second subsystem. It also imposes a latency proportional to the transferred variable's latency from the first DRTS to the second subsystem circuit and vice versa [33] [114].

control strategies, or demand response algorithms in a realistic simulation environment [102]. VILLAS facilitates the integration and co-simulation of DERs like solar photovoltaics, wind turbines, energy storage systems, and electric vehicles. It allows for studying the impact of DERs on system stability, grid resilience, and optimal operation [103].

Furthermore, HELICS is used to co-simulate wide-area monitoring and control systems in power systems. It enables the exchange of real-time measurements and control signals between geographically distributed simulation models, allowing for coordinated monitoring and control of the power grid [104]. HELICS enables the integration of cyber and physical components of power systems [105]. This includes co-simulation of power system dynamics, communication networks, and control systems, enabling the study of cyber-physical interactions and their impact on system behavior and resilience [106], [107]. In similarity to VILLAS [102], HELICS facilitates the co-simulation of transactive energy

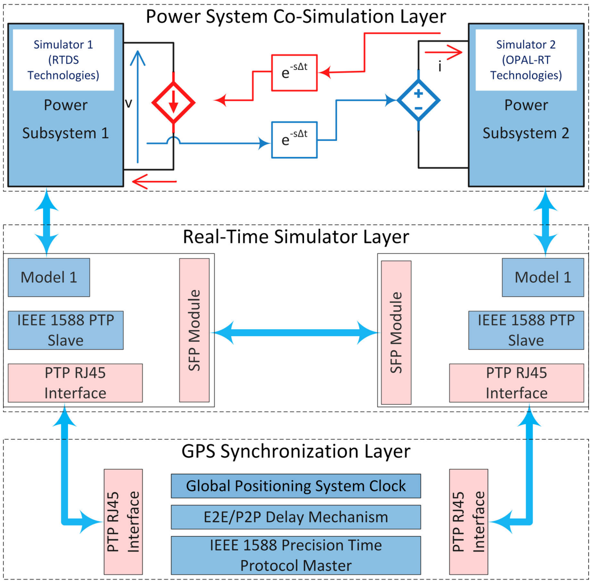

FIGURE 24. GDRTCS three main architectural layers.

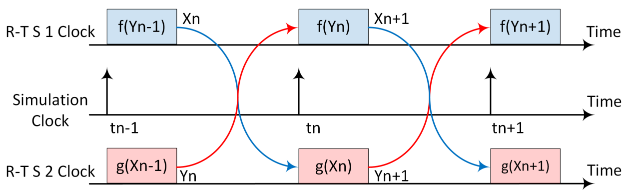

FIGURE 25. Simulation times of two real-time simulators with zero communication latency between them.

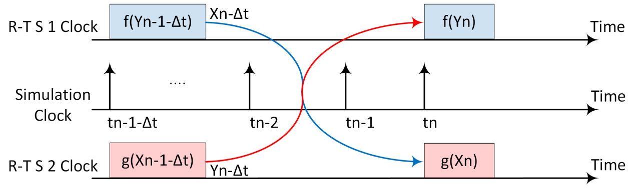

FIGURE 26. Simulation times of two real-time simulators with communication latency between them.

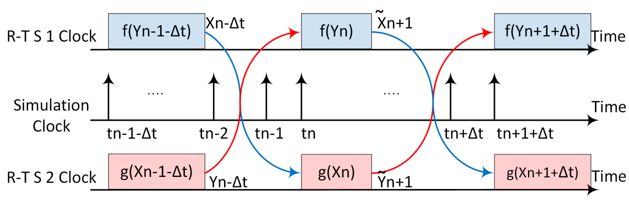

FIGURE 27. Simulation times of two real-time simulators with communication latency between them and its effect.

systems, where energy producers and consumers interact in real-time markets. It enables the evaluation of market-based mechanisms, demand response strategies, and optimization algorithms for energy exchange and grid management [108], [109], [110].

VI. LATENCY AND DELAY COMPENSATION

A. COMMUNICATION LATENCY

For feasibility, communication happens via the Internet, with time delays depending on the obtainable network infrastructure between locations, as opposed to monolithic systems, which exhibit non-deterministic time delays [44]. In the context of co-simulation, latency refers to the time delay between the sending process of data collected from a real-time simulator and the subsequent receiving process, which passes the received data to the solver of other distributed real-time simulators for completing their numerical calculations. This critical latency can lead to instability and inaccuracies within a GDRTCS environment. Therefore, minimizing latency by implementing specific solutions to ensure the smooth and reliable operation of distributed real-time simulators is imperative. Addressing latency-related challenges is crucial for achieving precise and robust results in GDRTCS setups [30].

Communication latency is important in internet-based GDRTCS as it affects simulation fidelity and stability. According to [5], Internet communication delay has four forms: Signal propagation delay, Transmission delay, Network processing delay, and Queuing delay. The signal propagation delay arises due to the nature of the medium and the distance between nodes. For example, the signal propagates at around 70% speed of light (299,792,458 meters per second) in a vacuum. Network equipment processing operation and technology determine network processing delay. Transmission delay is a packet's time to reach the physical link layer. Therefore, transmission delay depends on link bandwidth and packet size. Queuing delay is a packet's time in a queue. Several packets from a single input port sent through to a single output port cause queuing delays. A queue holds the remaining packets while one is transmitted. All four delays make up communication latency. Communication delay typically varies from some milli-seconds to several 100 milli-seconds. The communication delay is crucial for geographically distributed real-time simulations, especially for synchronized cases like an internet-based virtual integration of real-time simulators for power system simulation. In addition, geographically dispersed power system transient events and WAN delay will influence real-time simulations. [5].

Figures 25, 26, and 27 show the timeline of geographically distributed real-time co-simulation with two simulators to show the influence of latency. These figures show the timing diagram between two real-time simulators in a co-simulating; thus, indicating the delay impact on the time in reference [49]. The simulators simulate subsystems $f(Y)$ and $g(X)$ of a model. Subsystem 2's input is X which is Subsystem 1's output. The two RT simulators' overall communication delay is Δt. Figure 25 shows the ideal situation with no communication lag between the two real-time simulators; the two subsystems complete their respective simulations for each time step and

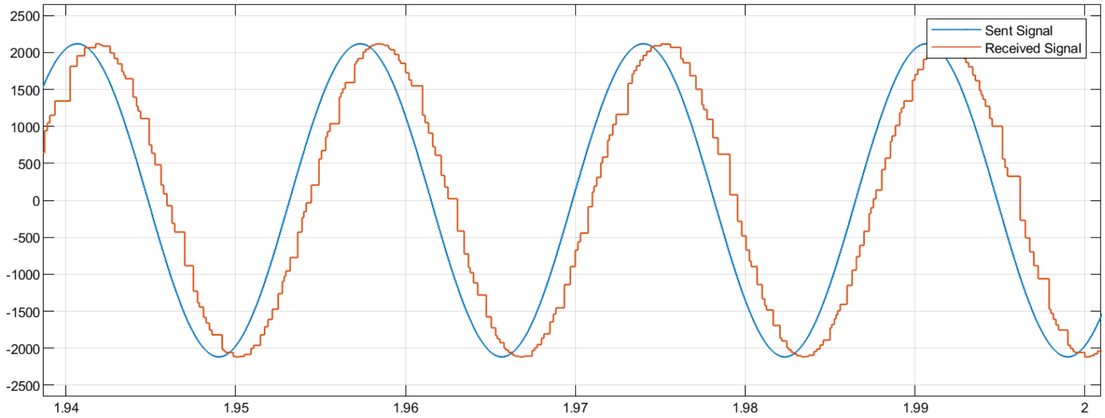

FIGURE 28. Sent and received sine waves during co-simulation.

communicate the simulated subsystem's status to the next one as input. Figure 26 indicates that when two real-time simulators have communication latency Δt, the subsystem inputs at t_n are altered from X_n and Y_n to $X_n - \Delta t$ and $Y_n - \Delta t$ as given in [5].

Figure 28 shows how communication latencies affect the perfect sine wave. The reference signal sent, and the delayed signal received differ at any given time. Similarly, in reference [115], the sent and received sinusoidal 0.1 Hz wave between RTDS and OPAL-RT real-time simulators revealed that there is 0.2 seconds communication latency shift in the sent and received signals.

Time-step t_n are altered and not strictly adhered to due to subsystem input errors, resulting in less accurate results coming from the simulation. In Figure 27, the outputs of both subsystems at tn diverge from the standard, thereby adding error to the successive time-step and affecting all subsequent simulations. These errors can be ignored if the communication delay is smaller than the RT simulator time-step. Nevertheless, the average time-step for real-time experiments of power system dynamics and transients ranges from a few milli-seconds to only some microseconds, whereas the delay is only some milliseconds to only some hundred milliseconds. Consequently, a GDRTCS utilizing dynamic simulation and real-time transient analysis is susceptible to resulting errors due to numerical instability [5].

1) INTERFACE ENERGY CONSERVATION

Power variables—current and voltage—are transferred between real-time simulators in co-simulation; thus, the co-simulation set-up accuracy depends on interface energy conservation. Simulated subsystem interfaces cannot generate or store energy. Therefore, simulation stability and fidelity are compromised if energy conservation is breached, thereby invalidating simulation results. Simulation fidelity measures how closely geographically distributed real-time simulation results match monolithic model simulations. Therefore, the model splitting point selected for distributed simulation of the monolithic model is crucial. According to empirical investigations, substation transformers and HVDC lines are suited for model partitioning. However, Internet connectivity between real-time simulators introduces communication latency, jitter, packet loss, and packet reordering [12]. Thus, interface energy conservation has the potential to be considerably breached. Thus, co-simulation interface algorithms, as discussed in Section II, are needed to compensate for communication network effects on simulation stability and fidelity. The benefit of the co-simulation interface algorithm based on dynamic phasors is that their representation inherently permits time delay compensation via phase shift [49].

In GDRTCS, the exchange of interface values between distributed real-time simulators typically involves variables specifying power, voltage, and current at the interfaces. However, transferring power variables over a shared communication network can compromise energy conservation, a crucial requirement for GDRTCS. As shown in Figure 29, co-simulation connections introduce virtual power due to time delays and other communication characteristics between subsystems. This violation of energy conservation in GDRTCS can lead to issues with simulation stability and accuracy. To address this challenge, the Co-simulation IA aims to mitigate and compensate the impact of subsystem communication characteristics [53]. The literature defines this issue as creating residual energy, which gets added to the energy flow in the co-simulation process. This energy discrepancy can distort system dynamics, compromise simulation accuracy, and even potentially lead to instability [8].

Digital RT co-simulation power system splitting is suggested in [41], where Figure 30 shows a six-step flowchart. Splitting a power system begins with identifying splitting points. Depending on system complexity and series component connections, splitting points can be simple, moderate, or complicated. The grid's basic elements, usually connected by a few series components, are called "easy" levels (e.g., single transmission lines). Moderately complex places have multiple

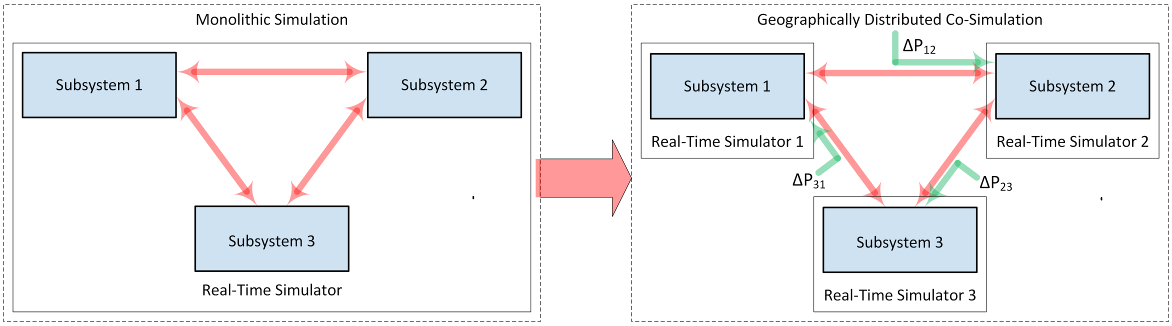

FIGURE 29. Energy conservation violation in GDRTCS.

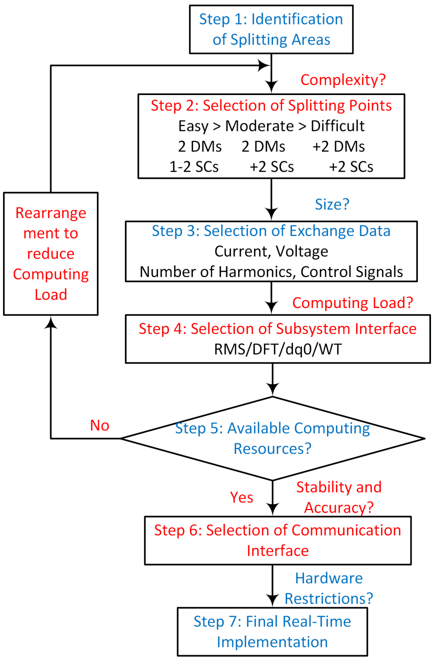

FIGURE 30. Power systems splitting methodology for GDRTCS.

series components (e.g., two double transmission lines linking three substations). Complicated splitting sites, such as densely meshed areas that connect numerous nodes, should be avoided. Choosing easy or moderate splitting points can separate a monolithic model into two distributed models (DMs). In areas with low complexity, one or two series components are coupled at the splitting point. In moderately complex areas, more than two series components are coupled. However, in highly complex areas, partitioning a monolithic model results in more than two distributed models and multiple series components, necessitating the adoption of more than one digital real-time simulation framework. To implement GDRTCS for the IEEE Australian benchmark model, the network partitioning and splitting are carried out in alignment with the methodology proposed by Reference [41]. The approach considers the complexity of different areas and adapts the coupling of series components accordingly.

2) DISTORTION

The internet's delay, jitter, and loss make it a challenging and complicated channel for geographically dispersed real-time simulation [86]. Distortion measures how far an Internet-based geographically dispersed real-time simulation fails fidelity [61]. Transport-level protocols like TCP can reduce loss but increase the delay. The coupling point, or port where the subsystem models are integrated via the internet, impacts distortion and delay. These Internet properties affect system stability, robustness, and transparency [86].

3) CLOSED LOOP NATURE OF CO-SIMULATION SYSTEM

The virtual integration and coupling of geographically scattered real-time simulators across communication networks are done in a closed loop [76]. A networked closed-loop system is a system that integrates scattered components into a closed loop via a communication channel. Examples of applications include networked PHIL, CHIL, control systems, and teleoperation [116]. The delays caused by the closed loop between geographically separated experiments across communication networks must be overcome to maximize the potential of geographically distributed real-time simulators. Delays can severely impact system performance and possibly destabilize networked closed-loop systems [76], [116].

B. DELAY COMPENSATION FRAMEWORKS

Delay compensation frameworks attempt to improve system performance and preserve system stability. These frameworks can either be model-based or model-free. Researchers have developed different delay compensation frameworks for networked systems with delayed communications that attempt to improve system performance and maintain system stability. These frameworks may utilize models or not.

Figure 31 provides a visualization of the timeline involved in integrating distributed real-time simulation with a prediction function denoted as $P(X_n)$. At each time-step t_n, X_n and Y_n represent the predicted values. However, due to communication latency and incomplete data exchange, the predicted value X_n cannot precisely match the actual value X_n. The primary objective of prediction is to minimize the error between the predicted and actual values of variables transmitted over the WAN.

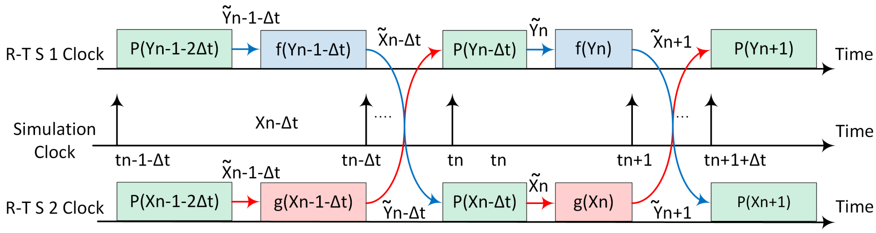

FIGURE 31. Communication latency and prediction between the two RT simulators.

The error is calculated as the absolute difference between X_n and $X_n - \Delta t$, depicted as $X_n - X_n - \Delta t$. As depicted in Figure 31, the prediction is carried out before each simulation's time step. This approach ensures that the total time required for prediction and simulation remains lesser than the time-step duration of the real-time simulation.

1) DELAY COMPENSATION FRAMEWORK FOR TIME-VARYING SIGNALS

The estimated time delay can be converted into an equivalent phase, thereby compensating delay in reconstructing the sent signals at the receiving end.

Delay Compensation in Dynamic Phasor: The remote simulator gets estimated DPs with a phase shift-compensated time delay t in [49]. After receiving a data packet, the phase shift needs accurate time delay information. The packets transmitted between real-time simulators are time-stamped, allowing the receiver to calculate the time delay [49]. The following defines phase shift-based delay compensation signal reconstruction:

$$x_p\left(n+d_n\right) = \sum_{k \in K} X_k(n) e^{+jk(2\pi/N)n} e^{+jk(2\pi/N)d_n} \tag{28}$$

In Equation 28, the range for the variable K is defined as follows: $K = [-(N/2), (N/2) - 1]$ when N is an even number, and $K = [-(N - 1/2), (N - 1/2)]$ when N is an odd number, as specified in Section III. The time delay of received DP relative to the time instant $t = nTs$ when the DP values were transmitted is denoted as d_n. This parameter varies with time and is nondeterministic. However, the procedure for reconstructing a sample of the time domain waveform is deterministic. This deterministic reconstruction is beneficial not only for compensating for time-varying delays but also for addressing packet losses. In the event of a packet loss, the previous DP estimates will be utilized, with the phase shift time delay extended by Ts. The study conducted by [49] presents a timing diagram that illustrates the timing chart of GDRTCS using the IA in a co-simulation framework between two real-time simulators.

Figure 32 shows the implementation results of communication latency compensation in GDRTCS and IA. The reference signal is transmitted across a communication medium with a constant and variable delay over time. The remote subsystem's time shift unaffected the waveform sent via the proposed

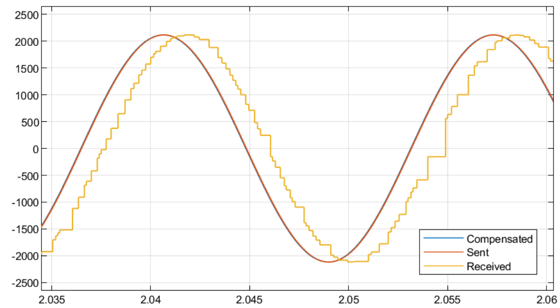

FIGURE 32. Delay compensation.

IA with DPs. In the absence of a DPs-based interface, the time-domain waveform is time-shifted relative to the reference signal due to a time delay. In the event of time-varying delay, considerable enhancements can also be noticed. When DPs-based IA is used, a time-varying delay will not severely degrade the received signal. In the absence of a DP-based link, delay variations introduce nonexistent in the reference signal dynamics [49].

Delay Compensation in Synchronous Reference Frame: At each subsystem end, the time delay is computed upon reception of the UDP packet by subtracting the UDP packet's time stamp from the current GPS time. Thus, the elapsed time for a UDP packet to traverse from the first subsystem to the second subsystem is referred to as the feed-forward time delay, denoted as d_{ff}. Conversely, the time taken for a UDP packet to travel from the second subsystem back to the first subsystem is termed the feedback time delay, designated as d_{fb}. The feed-forward and feedback time delays advance the reconstruction of voltage and current at the second and first subsystems, respectively. These are relative to each (other) remote subsystem by a phase equal to the measured time delay computed as ϕ_{ff} and ϕ_{fb} in Equations 29 and 30.

$$\varphi_{ff} = 2\pi f_h \tau_{dff} \tag{29}$$

$$\varphi_{fb} = 2\pi f_h \tau_{dfb} \tag{30}$$

where f_h is the nominal frequency of each compensable harmonic component. Shifting the recorded phase signal, as shown in Figure 33, mitigates or compensates for the delay's influence during signal or waveform reconstructions in respective subsystems [3]; the reconstructed received signals match the sent signal without delay. Therefore, the

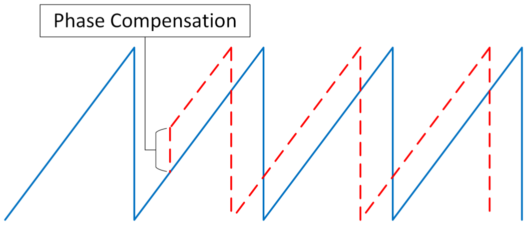

FIGURE 33. Instantaneous phase signal's phase compensation.

phase compensation terms (ϕ_{ff} and ϕ_{fb}) must be frequency-adaptive (measuring the instantaneous frequency of each harmonic component instead of a constant nominal frequency) for precise compensation. The nominal frequency use in the signal estimating corresponding phase delay leads to steady-state inaccuracies under transients. These are shown in Equations 31 and 32:

$$\varphi_{ff} = 2\pi f_{hm}\tau_{dff} \tag{31}$$

$$\varphi_{fb} = 2\pi f_{hm}\tau_{dfb} \tag{32}$$

The interface waveforms are rebuilt using the inverse Park's transformation as described by Equations 33 and 34.

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \cos(wt+\phi_{Td}) & -\sin(wt+\phi_{Td}) & 1 \\ \cos\left(wt-\frac{2\pi}{3}+\phi_{Td}\right) & -\sin\left(wt-\frac{2\pi}{3}+\phi_{Td}\right) & 1 \\ \cos\left(wt+\frac{2\pi}{3}+\phi_{Td}\right) & -\sin\left(wt+\frac{2\pi}{3}+\phi_{Td}\right) & 1 \end{bmatrix} \begin{bmatrix} u_d \\ u_q \\ u_0 \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \frac{2}{3}\begin{bmatrix} \cos(wt+\phi_{Td}) & -\sin(wt+\phi_{Td}) & 1 \\ \cos\left(wt-\frac{2\pi}{3}+\phi_{Td}\right) & -\sin\left(wt-\frac{2\pi}{3}+\phi_{Td}\right) & 1 \\ \cos\left(wt+\frac{2\pi}{3}+\phi_{Td}\right) & -\sin\left(wt+\frac{2\pi}{3}+\phi_{Td}\right) & 1 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} \tag{34}$$

2) MODEL-BASED DELAY COMPENSATION FRAMEWORK

Model-based control compensation approaches utilize a dynamic plant model of the system components to predict the system's future states and consequently calculate the required control command to apply to the system, considering the delays. Model-Based Delay Compensation Framework compensates for modeling errors and disturbances. This framework handles delay compensation and tracking well.. This method improves prediction accuracy when models are high-fidelity. The drawbacks include: When models are inaccurate or unavailable, their applications may be hindered [76], and When dynamic models are unavailable, they may not be robust and applicable [116]

Examples of model-based implementation include the Predictive control scheme implemented in [117] and [118], the active resilient control scheme implemented in [119], the Model-based event-triggered predictive control (MB-ETPC) scheme implemented in [120], the Adaptive unscented Kalman filter framework implemented in [121], Variable-horizon model predictive control scheme implemented in [121]

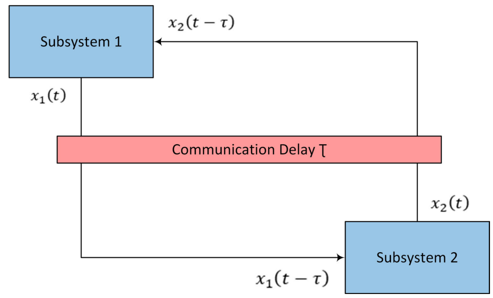

FIGURE 34. Subsystems with constant communication delays in between.

3) MODEL-FREE DELAY COMPENSATION FRAMEWORK

This delay compensation framework can estimate without a system dynamic model. The merits of this technique are that they compensate for unknown and changing delays in the control system. These can be used in many networked systems despite model faults or unavailability. However, according to reference [66], current model-free approaches may trade off performance, be suitable for a specific application, need repeated tests, or require correct information or estimation of the statistical properties of the noise that drives prediction.

Examples include the butterfly-shaped inner feedback loop implemented in [122], the Kalman-filter-based delay compensation framework implemented in [123], wave transformations (WTs) implemented in [124], neuro-fuzzy (NF) predictive controller implemented in [124], hybrid-driven control framework implemented in [125], model-free method of signal prediction implemented in [126] and iterative learning control approach implemented in [127].

4) DELAY COMPENSATION FRAMEWORK IMPLEMENTATIONS

Reference [76] adopted the framework proposed by [116] as a foundation and expanded its design by introducing an additional tuning parameter. This enhancement allowed for increased design flexibility, going from one to two degrees of freedom, resulting in improved overall performance. Importantly, the extended framework retained the original input-output structure and model-free characteristics of the predictor. The finding and future works recommended in references [76], [86], and [116] are summarized in Table 8.

Two subsystems with and without predictors added for constant communication delays in-between are shown in Figures 34 and 35. In addition, one parameter predictor with a restricted design degree of freedom is proposed in reference [76]. According to [86], to measure delay time τ, synchronization clocks are required on both ends of the network and subtract the send and receipt timestamps.

TABLE 8. Summary of delay compensation frameworks in [76], [86], and [116].

References	Framework/Concept	Findings	Further work
Reference [86]	Model-Based Integrate two separated HIL simulations for an automotive application. Engine-in-the-loop simulation.	Even with a four-fold nominal latency, a model-based approach transparently integrated hardware across the internet. System signals vary in transparency.	Identifying HIL model based solution's fundamental constraints. Exploring how to push those constraints or limitations. Adding new communication methods.
Reference [76]	Model-Free Framework. To enhance performance application-agnostically without repeating experiments. Remote components interact via local predictors to forecast their current output signals based on measured delays of remote set-up. One design parameter tunes each predictor.	Extended predictor structure compensates for delays. Network delays as a two-time-scale Markov chain ensure stability with changing delays. A good but not flexible range of the predictor parameters exists to ensure stability despite delays. The predictor-based methodology greatly reduces the detrimental delay impacts on networked integration fidelity in case studies (constant and changing delay). Predictors can stabilize a delayed closed-loop system.	Studying predictor performance with different delays and robustness to communication network issues such as packet losses and reordering. Application to nonlinear systems. Predictor testing using physical subsystems.
Reference [116]	Model-Free Framework. Added a second tuning parameter to boost performance by doubling the design degree of freedom. Evaluated open-loop stability and frequency-domain performance of this framework. Framework compares to its one-parameter predecessor in simulated and real networked testbeds.	Design parameters and delay define the predictor's open-loop stability boundary. Then, the predictor's frequency-domain performance is assessed with respect to these parameters and constant latency. Improves prediction accuracy by tuning parameters. A connected testbed simulates and experiments on a commercial vehicle. Improved coupling fidelity of the system by adding one additional tuning parameter to the predictor design to increase the design degree of freedom.	Investigating additional tuning parameters (design degree of freedom) and closed-loop stability study. The predictor can be tuned adaptively to simplify practical applications and eliminate human adjustment. The scalability of this framework is to be tested with multiple remotely Connected and accessible testbeds.

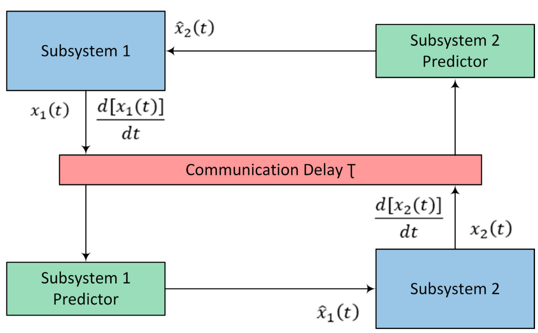

FIGURE 35. Subsystems with predictors added for constant communication delays in-between.

VII. CONCLUSION AND FUTURE WORKS

The real-time simulators are geographically dispersed, and the capabilities of the local real-time simulator are insufficient for the specific simulation scenario due to modeling and simulation restrictions. GDRTCS is required when the complexity of the system models exceeds the capability of a single simulation platform or when system components are being developed in different locations. Bringing them together in a single facility would be impractical, costly, or violate intellectual property restrictions. Thus, the simulated model can be partitioned for a distributed simulation while contending with energy conservation at the interface. In addition, utilizing geographically dispersed simulation via effective communication mediums like the internet is cost-effective. Furthermore, these real-time simulators can collaborate to execute widespread co-simulation for resolving common challenges. This method allows specialized computational resource sharing and facilitates teamwork. However, in addition to the difficulty of splitting a complex model and building interfaces for parallel execution, the latency in power system EMT simulations might be even longer than the simulation time step.

This overview has identified areas of future work regarding the conduct of real-time co-simulation experiments that virtually integrate geographically distributed research facilities with appropriate interface algorithms and delay compensation algorithms to improve the co- simulation's stability and resiliency:

1) Until now, instances of synchronously coupled configurations documented in the literature have primarily been confined to small-scale electrical circuits and experimental arrangements. These setups involve the

portioning of power systems across a stiff and higher capacity grid (transmission) and a weak and lower capacity grid (distribution). However, the primary objective of geographically dispersed real-time co-simulation is to enable the capability to undertake high-fidelity large system validation studies. In addition, applications have primarily been restricted to the study of slower dynamics. The interface strategies utilized for co-simulations are comparable to those utilized for monolithic CHIL and PHIL simulations; however, due to the geographic separation, the co-simulation procedures are slower and less stable. When compared to monolithic PHIL simulations with hardware attached to a real-time simulator, co-simulations present a greater challenge when it comes to partitioning. As a result, further research is required to determine the best method for splitting a system so that it can be subjected to co-simulation.

2) Investigating the operational range of different IAs to determine an advanced ITM best suitable for closed-loop virtual integration of two research facilities. The operational range of interface algorithms in real-time co-simulation of power systems depends on various factors, such as the complexity of the system, the level of detail required, the accuracy and speed of the numerical solvers, and the computational resources available. Interface algorithms should be designed to accurately represent the simulated components' behavior while minimizing computational complexity and maintaining real-time performance. It is necessary to ensure that the co-simulation is robust under transient situations and to prevent the propagation of errors within the simulation. The choice of interface algorithm will depend on the specific application and the simulation requirements. Also, IA must effectively manage the residual energy and ensure energy conservation in GDRTCS setups to guarantee fidelity and reliability.
3) Mitigating the degradation of simulation fidelity using the appropriate combination of signal decomposition and signal reconstruction interface to solve the difficulties in reproducing rapid dynamics and transients; There is a need to ensure robustness under fault conditions and prevent the propagation of errors in the simulation. Similarly, there is a need for scalability, as the size of power systems grows, the complexity of the interface signal transformation process also increases; thus, research is needed to develop interface signal transformation that can handle large-scale power systems and ensure the scalability of the simulation.
4) Compensating for communication delays using more than two-parameter model-free delay prediction and compensation framework for the networked systems to improve performance and stability. These tuning parameters must be designed, and the dynamic relationship between tunning parameters and the stability of the entire predictor must be established and developed. Furthermore, the designed predictor's frequency-domain analysis must be investigated to gain insight into the connection between the predictor's performance in steady-state, delays and the proposed additional design parameters.
5) Developing a delay-adaptive parameter model-free prediction and compensation framework for time-varying communication delay. The delay prediction and compensation frameworks must be extended to a varying time delay typical of a communication network and real-time experiments in a power system. This will prove validity for real-time co-simulation.

REFERENCES

[1] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.

[2] J. Ihrens, S. Möws, L. Wilkening, T. A. Kern, and C. Becker, "The impact of time delays for power hardware-in-the-loop investigations," *Energies*, vol. 14, no. 11, p. 3154, May 2021.

[3] M. Bokal, I. Papic, and B. Blažic, "Stabilization of hardware-in-the-loop ideal transformer model interfacing algorithm by using spectrum assignment," *IEEE Trans. Power Del.*, vol. 34, no. 5, pp. 1865–1873, Oct. 2019.

[4] M. Mirz, A. Estebsari, F. Arrigo, E. Bompard, and A. Monti, "Dynamic phasors to enable distributed real-time simulation," in *Proc. 6th Int. Conf. Clean Electr. Power (ICCEP)*, Jun. 2017, pp. 139–144.

[5] R. Liu, M. Mohanpurkar, M. Panwar, R. Hovsapian, A. Srivastava, and S. Suryanarayanan, "Geographically distributed real-time digital simulations using linear prediction," *Int. J. Electr. Power Energy Syst.*, vol. 84, pp. 308–317, Jan. 2017.

[6] J. M. B. Smith, K. Balasubramaniam, and R. Hadidi, "Assessing the need and state of power system and transportation system co-simulation," *IEEE Access*, vol. 11, pp. 44941–44951, 2023.

[7] R. A. Tharinda and J. K. Rupasinghe, "Advanced methods, models, and algorithms for multi-rate co-simulation of power system transients," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Manitoba, Winnipeg, MB, Canada, 2021.

[8] W. Fusto, "Interface algorithms for a multi-site platform for real-time co-simulation of power systems," Ph.D. dissertation, Collegio di Ingegneria Elettrica, Politecnico di Tori, Turin, Italy, 2017.

[9] S. Vogel, "Development of a modular and fully-digital PCIe-based interface to real-time digital simulator," M.S. thesis, Inst. Automat. Complex Power Syst., RWTH Aachen Univ., Aachen, Germany, 2017.

[10] P. Palensky, A. A. Van Der Meer, C. D. Lopez, A. Joseph, and K. Pan, "Cosimulation of intelligent power systems: Fundamentals, software architecture, numerics, and coupling," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 34–50, Mar. 2017.

[11] P. Palensky, A. van der Meer, C. Lopez, A. Joseph, and K. Pan, "Applied cosimulation of intelligent power systems: Implementing hybrid simulators for complex power systems," *IEEE Ind. Electron. Mag.*, vol. 11, no. 2, pp. 6–21, Jun. 2017.

[12] A. Monti, M. Stevic, S. Vogel, R. W. De Doncker, E. Bompard, A. Estebsari, F. Profumo, R. Hovsapian, M. Mohanpurkar, J. D. Flicker, V. Gevorgian, S. Suryanarayanan, A. K. Srivastava, and A. Benigni, "A global real-time superlab: Enabling high penetration of power electronics in the electric grid," *IEEE Power Electron. Mag.*, vol. 5, no. 3, pp. 35–44, Sep. 2018.

[13] M. D. O. Faruque, T. Strasser, G. Lauss, V. Jalili-Marandi, P. Forsyth, C. Dufour, V. Dinavahi, A. Monti, P. Kotsampopoulos, J. A. Martinez, K. Strunz, M. Saeedifard, X. Wang, D. Shearer, and M. Paolone, "Real-time simulation technologies for power systems design, testing, and analysis," *IEEE Power Energy Technol. Syst. J.*, vol. 2, no. 2, pp. 63–73, Jun. 2015.

[14] *OP101: RT-LAB—Real-Time Simulation Systems Fundamentals*. Accessed: Mar. 31, 2023. [Online]. Available: https://www.opal-rt.com/opal_tutorial/startup-rtlab/

[15] M. O. Faruque, V. Dinavahi, M. Sloderbeck, and M. Steurer, "Geographically distributed thermo-electric co-simulation of all-electric ship," in *Proc. IEEE Electric Ship Technol. Symp.*, Apr. 2009, pp. 36–43.

[16] P. H. Hoang, G. Ozkan, P. R. Badr, L. Timilsina, B. Papari, and C. S. Edrington, "Integrating degradation forecasting into distribution grids' advanced distribution management systems," *Int. J. Electr. Power Energy Syst.*, vol. 150, Aug. 2023, Art. no. 109071.

[17] F. Loku, L. Osterkamp, P. Düllmann, C. Klein, M. Maimer, T. Bergwinkl, M. Kufner, M. Stevic, K. S. A. Kumar, and R. Venugopal, "Utilization of the impedance-based stability criterion for stability assessment of PHiL interface algorithms," in *Proc. IEEE 13th Int. Symp. Power Electron. Distrib. Gener. Syst. (PEDG)*, Jun. 2022, pp. 1–7.

[18] M. Stevic, M. Panwar, M. Mohanpurkar, R. Hovsapian, and A. Monti, "Empirical study of simulation fidelity in geographically distributed real-time simulations," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2017, pp. 1–6.

[19] M. H. Syed, E. Guillo-Sansano, S. M. Blair, A. Avras, and G. M. Burt, "Synchronous reference frame interface for geographically distributed real-time simulations," *IET Gener., Transmiss. Distrib.*, vol. 14, no. 23, pp. 5428–5438, Dec. 2020.

[20] W. Ren, M. Sloderbeck, M. Steurer, V. Dinavahi, T. Noda, S. Filizadeh, A. R. Chevrefils, M. Matar, R. Iravani, C. Dufour, J. Belanger, M. O. Faruque, K. Strunz, and J. A. Martinez, "Interfacing issues in real-time digital simulators," *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 1221–1230, Apr. 2011.

[21] R. Brandl, "Operational range of several interface algorithms for different power hardware-in-the-loop setups," *Energies*, vol. 10, no. 12, p. 1946, Nov. 2017.

[22] W. Ren, M. Steurer, and T. L. Baldwin, "Improve the stability and the accuracy of power hardware-in-the-loop simulation by selecting appropriate interface algorithms," *IEEE Trans. Ind. Appl.*, vol. 44, no. 4, pp. 1286–1294, Aug. 2008.

[23] G. Lauss and K. Strunz, "Multirate partitioning interface for enhanced stability of power hardware-in-the-loop real-time simulation," *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 595–605, Jan. 2019.

[24] Z. Zhang, L. Fickert, and Y. Zhang, "Power hardware-in-the-loop test for cyber physical renewable energy infeed: Retroactive effects and an optimized power hardware-in-the-loop interface algorithm," in *Proc. 17th Int. Sci. Conf. Electr. Power Eng. (EPE)*, May 2016, pp. 1–6.

[25] R. Schwendemann, L. Stefanski, and M. Hiller, "Comparison of different interface algorithms for a highly dynamic grid emulator based on a series hybrid converter," in *Proc. IEEE 23rd Workshop Control Modeling Power Electron. (COMPEL)*, Jun. 2022, pp. 1–8.

[26] S. Ruhe, M. Fechner, S. Nicolai, and P. Bretschneider, "Simulation of coupled components within Power-Hardware-in-the-Loop (PHiL) test bench," in *Proc. 55th Int. Universities Power Eng. Conf. (UPEC)*, Sep. 2020, pp. 1–6.

[27] M. G. Kashani, H. Pulakhandam, S. Bhattacharya, F. Katiraei, and D. Kaiser, "Design considerations and test setup assessment for power hardware in the loop testing," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting*, Oct. 2017, pp. 1–8.

[28] R. Schwendemann, D. Schulz, L. Stefanski, and M. Hiller, "A 60 kW power hardware-in-the-loop test bench for grid emulation based on a series hybrid cascaded H-bridge converter," in *Proc. 23rd Eur. Conf. Power Electron. Appl. (EPE ECCE Europe)*, Sep. 2021, pp. 1–10.

[29] G. F. Lauss, M. O. Faruque, K. Schoder, C. Dufour, A. Viehweider, and J. Langston, "Characteristics and design of power hardware-in-the-loop simulations for electrical power systems," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 406–417, Jan. 2016.

[30] L. Barbierato, E. Pons, A. Mazza, E. F. Bompard, V. S. Rajkumar, P. Palensky, E. Macii, L. Bottaccioli, and E. Patti, "Stability and accuracy analysis of a distributed digital real-time cosimulation infrastructure," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, pp. 3193–3204, May 2022.

[31] S. Resch, J. Friedrich, T. Wagner, G. Mehlmann, and M. Luther, "Stability analysis of power hardware-in-the-loop simulations for grid applications," *Electronics*, vol. 11, no. 1, p. 7, Dec. 2021.

[32] M. Steurer, C. S. Edrington, M. Sloderbeck, W. Ren, and J. Langston, "A megawatt-scale power hardware-in-the-loop simulation setup for motor drives," *IEEE Trans. Ind. Electron.*, vol. 57, no. 4, pp. 1254–1260, Apr. 2010.

[33] X. Wu, S. Lentijo, A. Deshmuk, A. Monti, and F. Ponci, "Design and implementation of a power-hardware-in-the-loop interface: A nonlinear load case study," in *Proc. 20th Annu. IEEE Appl. Power Electron. Conf. Expo. (APEC)*, vol. 1, Mar. 2005, pp. 1332–1338.

[34] X. Wu and A. Monti, "Methods for partitioning the system and performance evaluation in power-hardware-in-the-loop simulations. Part I," in *Proc. 31st Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2005, p. 6.

[35] V. B. Dmitriev-Zdorov and B. Klaassen, "An improved relaxation approach for mixed system analysis with several simulation tools," in *Proc. EURO-DAC. Eur. Design Autom. Conf.*, Sep. 1995, pp. 274–279.

[36] V. B. Dmitriev-Zdorov, "Generalized coupling as a way to improve the convergence in relaxation-based solvers," in *Proc. EURO-DAC. Eur. Design Autom. Conf. EURO-VHDL Exhib.*, Sep. 1996, pp. 15–20.

[37] A. Aguirre, M. Davila, P. Zuniga, F. Uribe, and E. Barocio, "Improvement of damping impedance method for power hardware in the loop simulations," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Nov. 2016, pp. 1–6.

[38] S. Paran and C. S. Edrington, "Improved power hardware in the loop interface methods via impedance matching," in *Proc. IEEE Electr. Ship Technol. Symp. (ESTS)*, Apr. 2013, pp. 342–346.

[39] S. Paran, T. V. Vu, F. Diaz Franco, and C. S. Edrington, "Evaluation of the interface accuracy for power hardware-in-the-loop experiments," *Electr. Power Compon. Syst.*, vol. 45, no. 7, pp. 763–773, Apr. 2017.

[40] A. Castro, P. Zuniga, F. A. Uribe, and E. Barocio, "Synchronous generator emulator prototype as a testbed for electric equipment," *IEEE Trans. Energy Convers.*, vol. 30, no. 1, pp. 404–406, Mar. 2015.

[41] Z. Shen, F. Arraño-Vargas, H. R. Wickramasinghe, and G. Konstantinou, "Development of power system models for distributed real-time simulations," *IEEE Access*, vol. 10, pp. 119706–119721, 2022.

[42] M. Pokharel and C. N. M. Ho, "Development of interface model and design of compensator to overcome delay response in a PHIL setup for evaluating a grid-connected power electronic DUT," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, pp. 4109–4121, May 2022.

[43] A. Summers, J. Hernandez-Alvidrez, R. Darbali-Zamora, M. J. Reno, J. Johnson, and N. S. Gurule, "Comparison of ideal transformer method and damping impedance method for PV power-hardware-in-the-loop experiments," in *Proc. IEEE 46th Photovoltaic Specialists Conf. (PVSC)*, pp. 2989–2996, Jun. 2019.

[44] J. Montoya et al., "Advanced laboratory testing methods using real-time simulation and hardware-in-the-loop techniques: A survey of smart grid international research facility network activities," *Energies*, vol. 13, no. 12, p. 3267, Jun. 2020.

[45] J. Rupasinghe, S. Filizadeh, and K. Strunz, "Assessment of dynamic phasor extraction methods for power system co-simulation applications," *Electr. Power Syst. Res.*, vol. 197, Aug. 2021, Art. no. 107319.

[46] S. Vogel, M. Stevic, R. Kadavil, M. Mohanpurkar, P. Koralewicz, V. Gevorgian, R. Hovsapian, and A. Monti, "Distributed real-time simulation and its applications to wind energy research," in *Proc. IEEE Int. Conf. Probabilistic Methods Appl. Power Syst. (PMAPS)*, 2018, pp. 1–6, doi: 10.1109/PMAPS.2018.8440410.

[47] S. Vogel, H. T. Nguyen, M. Stevic, T. V. Jensen, K. Heussen, V. S. Rajkumar, and A. Monti, "Distributed power hardware-in-the-loop testing using a grid-forming converter as power interface," *Energies*, vol. 13, no. 15, p. 3770, Jul. 2020.

[48] T. Demiray, G. Andersson, and L. Busarello, "Evaluation study for the simulation of power system transients using dynamic phasor models," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo., Latin Amer.*, Aug. 2008, pp. 1–6.

[49] M. Stevic, A. Estebsari, S. Vogel, E. Pons, E. Bompard, M. Masera, and A. Monti, "Multi-site European framework for real-time co-simulation of power systems," *IET Gener., Transmiss. Distrib.*, vol. 11, no. 17, pp. 4126–4135, Nov. 2017.

[50] K. Mudunkotuwa and S. Filizadeh, "Co-simulation of electrical networks by interfacing EMT and dynamic-phasor simulators," *Electr. Power Syst. Res.*, vol. 163, pp. 423–429, Oct. 2018.

[51] T. Demiray and G. Andersson, "Optimization of numerical integration methods for the simulation of dynamic phasor models in power systems," *Int. J. Electr. Power Energy Syst.*, vol. 31, no. 9, pp. 512–521, Oct. 2009.

[52] T. Yang, S. Bozhko, J.-M. Le-Peuvedic, G. Asher, and C. I. Hill, "Dynamic phasor modeling of multi-generator variable frequency electrical power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 563–571, Jan. 2016.

[53] M. Stevic and A. Monti, "A bilateral teleoperation approach for interface algorithms in distributed real-time simulations," in *Proc. IEEE Workshop Complex. Eng. (COMPENG)*, Oct. 2018, pp. 1–5.

[54] M. Stevic, S. Vogel, A. Monti, and S. D'Arco, "Feasibility of geographically distributed real-time simulation of HVDC system interconnected with AC networks," in *Proc. IEEE Eindhoven PowerTech*, Jun. 2015, pp. 1–5.

[55] S. Vogel, V. S. Rajkumar, H. T. Nguyen, M. Stevic, R. Bhandia, K. Heussen, P. Palensky, and A. Monti, "Improvements to the co-simulation interface for geographically distributed real-time simulation," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, vol. 1, Oct. 2019, pp. 6655–6662.

[56] E. G. Sansano, "Novel methods for enhancing accuracy and stability of power hardware-in-the-loop simulations," Ph.D. dissertation, Dept. Electron. Elect. Eng., Univ. Strathclyde, Glasgow, U.K., 2018.

[57] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, Dec. 2006.

[58] X. Gao, J. Silvério, S. Calinon, M. Li, and X. Xiao, "Bilateral teleoperation with object-adaptive mapping," *Complex Intell. Syst.*, vol. 8, pp. 2983–2990, Oct. 2021.

[59] M. Stevic and A. Monti, "Wave transformation based interface algorithm for distributed simulation of HVDC systems," in *Proc. 44th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2018, pp. 4019–4023.

[60] M. H. Syed, E. Guillo-Sansano, Y. Wang, S. Vogel, P. Palensky, G. M. Burt, Y. Xu, A. Monti, and R. Hovsapian, "Real-time coupling of geographically distributed research infrastructures: Taxonomy, overview, and real-world smart grid applications," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1747–1760, Mar. 2021.

[61] T. Ersal, R. Brent Gillespie, M. J. Brudnak, J. L. Stein, and H. K. Fathy, "Effect of coupling point selection on distortion in Internet-distributed hardware-in-the-loop simulation," *Int. J. Vehicle Design*, vol. 61, nos. 1–4, pp. 67–85, 2013.

[62] D. Liu, Z. Wu, Q. Guo, and Y. Shi, "Resilience and its thresholds of scientific collaboration network," *IEEE Access*, vol. 7, pp. 69339–69350, 2019.

[63] S. Ahmadi, P. Poure, S. Saadate, and D. A. Khaburi, "A real-time fault diagnosis for neutral-point-clamped inverters based on failure-mode algorithm," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1100–1110, Feb. 2021.

[64] L. Pellegrino, C. Sandroni, E. Bionda, D. Pala, D. T. Lagos, N. Hatziargyriou, and N. Akroud, "Remote laboratory testing demonstration," *Energies*, vol. 13, no. 9, p. 2283, May 2020.

[65] J. Xie and M. Swaminathan, "Electrical-thermal co-simulation with Joule heating and convection effects for 3D systems," U.S. Patent 9 037 446, May 19, 2015.

[66] H. Figueroa, J. Bastos, A. Monti, and R. Dougal, "A modular real-time simulation platform based on the virtual test bed," in *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 2, Jul. 2006, pp. 1537–1541.

[67] G. Cao, W. Gu, C. Gu, W. Sheng, J. Pan, R. Li, and L. Sun, "Real-time cyber–physical system co-simulation testbed for microgrids control," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 4, no. 1, pp. 38–45, Mar. 2019.

[68] V. Venkataramanan, A. Srivastava, and A. Hahn, "Real-time co-simulation testbed for microgrid cyber-physical analysis," in *Proc. Workshop Model. Simul. Cyber-Phys. Energy Syst. (MSCPES)*, Apr. 2016, pp. 1–6.

[69] R. Mao, H. Li, Y. Xu, and H. Li, "Wireless communication for controlling microgrids: Co-simulation and performance evaluation," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2013, pp. 1–5.

[70] T. I. Strasser, E. C. de Jong, and M. Sosnina, *European Guide to Power System Testing: The ERIGrid Holistic Approach for Evaluating Complex Smart Grid Configurations*. Cham, Switzerland: Springer, 2020.

[71] B. Palmintier, B. Lundstrom, S. Chakraborty, T. Williams, K. Schneider, and D. Chassin, "A power hardware-in-the-loop platform with remote distribution circuit cosimulation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2236–2245, Apr. 2015.

[72] B. Lundstrom, B. Palmintier, D. Rowe, J. Ward, and T. Moore, "Trans-oceanic remote power hardware-in-the-loop: Multi-site hardware, integrated controller, and electric network co-simulation," *IET Gener., Transmiss. Distrib.*, vol. 11, no. 18, pp. 4688–4701, Dec. 2017.

[73] M. Syed, T. T. Hoang, A. Kontou, A. Paspatis, G. Burt, Q. T. Tran, E. Guillo-Sansano, S. Vogel, H. T. Nguyen, and N. Hatziargyriou, "Applicability of geographically distributed simulations," *IEEE Trans. Power Syst.*, vol. 38, no. 4, pp. 3107–3122, Jul. 2023.

[74] M. Stevic, A. Monti, and A. Benigni, "Development of a simulator-to-simulator interface for geographically distributed simulation of power systems in real time," in *Proc. 41st Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2015, pp. 5020–5025.

[75] M. Stevic, S. Vogel, and A. Monti, "From monolithic to geographically distributed simulation of HVdc systems," in *Proc. IEEE 19th Workshop Control Modeling Power Electron. (COMPEL)*, Jun. 2018, pp. 1–5.

[76] S. Guo, Y. Liu, Y. Zheng, and T. Ersal, "A delay compensation framework for connected testbeds," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 7, pp. 4163–4176, Jul. 2022.

[77] Energy Sciences Network. (2022). *About ESnet: A Platform for Science Discovery*. [Online]. Available: https://www.es.net/about/

[78] *ESnet Networks*. (2022). [Online]. Available: https://www.es.net/assets/About-ESnet/15-CS-1035-ESnet-EuropeUS-MapEEXv4.pdf

[79] ESnet. *Historical Network Maps*. Accessed: Mar. 31, 2023. [Online]. Available: https://www.es.net/engineering-services/the-network/network-maps/historical-network-maps/

[80] J. Bashor. (2017). *ESnet, Internet2 and NYSERNet Extend Pact to Connect U.S. Research Community to Major International Hub*. [Online]. Available: https://www.es.net/news-and-publications/esnet-news/2017/esnet-internet2-and-nysernet-extend-pact-to-connect-u-s-research-community-to-major-international-hub/

[81] R. E. Hilpisch, R. Duchscher, M. Seel, and P. S. K. Hansen. (2009). *Wireless Communication Protocol*. [Online]. Available: https://worldwide.espacenet.com/patent/search/family/034965382/publication/US7529565B2?q=pn%3DUS7529565

[82] P. Wang, H. Wang, M. Yi, and Y. Zhang, "Conformance test method and implementation of EPA protocol state machine," *Comput. Integr. Manuf. Syst.*, vol. 13, no. 3, pp. 621–624, 2007.

[83] H. Fan, H. Wang, S. Xia, X. Li, P. Xu, and Y. Gao, "Review of modeling and simulation methods for cyber physical power system," *Frontiers Energy Res.*, vol. 9, May 2021, Art. no. 642997.

[84] Q. Wu, Y. Sun, M. Spiryagin, and C. Cole, "Parallel co-simulation method for railway vehicle-track dynamics," *J. Comput. Nonlinear Dyn.*, vol. 13, no. 4, Apr. 2018, Art. no. 041004.

[85] K. Jim and D. R. Keith, *Computer Networking: A Top-Down Approach*. London, U.K.: Pearson Education, 2009.

[86] T. Ersal, M. Brudnak, A. Salvi, J. L. Stein, Z. Filipi, and H. K. Fathy, "Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform," *Mechatronics*, vol. 21, no. 1, pp. 22–29, Feb. 2011.

[87] Y. Fan, Q. Wang, H. Peng, S. Lin, K. Fan, and Y. Chen, "GOOSE over UDP transmission mechanism for real-time data fast transmission in distribution network," in *Proc. 7th Int. Green Sustain. Comput. Conf. (IGSC)*, Nov. 2016, pp. 1–5.

[88] S. Heinig, K. Ilves, S. Norrga, and H.-P. Nee, "On energy storage requirements in alternate arm converters and modular multilevel converters," in *Proc. 18th Eur. Conf. Power Electron. Appl. (EPE ECCE Europe)*, Sep. 2016, pp. 1–10.

[89] D. Macii and S. Rinaldi, "Time synchronization for smart grids applications: Requirements and uncertainty issues," *IEEE Instrum. Meas. Mag.*, vol. 25, no. 6, pp. 11–18, Sep. 2022.

[90] M. Farraj, R. Salehi, A. Momeni, A. Zamani, and R. Kadem, "Validation of the power plant controller for frequency support applications using hardware-in-the-loop testing," in *Proc. IEEE Electr. Energy Storage Appl. Technol. Conf. (EESAT)*, Nov. 2022, pp. 1–5.

[91] L. Jacobs, C. DeCusatis, P. Wojciak, C. Kaiser, and S. Guendert, "Covert message channels and attack vectors for IEEE precision time protocol," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun. (ISPCS)*, Oct. 2022, pp. 1–5.

[92] W. Alghamdi and M. Schukat, "A security enhancement of the precision time protocol using a trusted supervisor node," *Sensors*, vol. 22, no. 10, p. 3671, May 2022.

[93] D. Chefrour, "Evolution of network time synchronization towards nanoseconds accuracy: A survey," *Comput. Commun.*, vol. 191, pp. 26–35, Jul. 2022.

[94] S. Vogel, M. Mirz, L. Razik, and A. Monti, "An open solution for next-generation real-time power system simulation," in *Proc. IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2017, pp. 1–6.

[95] E. Bompard, A. Monti, A. Tenconi, A. Estebsari, T. Huang, E. Pons, M. Stevic, S. Vaschetto, and S. Vogel, "A multi-site real-time co-simulation platform for the testing of control strategies of distributed storage and V2G in distribution networks," in *Proc. 18th Eur. Conf. Power Electron. Appl. (EPE ECCE Europe)*, Sep. 2016, pp. 1–9.

[96] M. Stevic, S. Vogel, M. Grigull, A. Monti, A. Estebsari, E. Pons, T. Huang, and E. Bompard, "Virtual integration of laboratories over long distance for real-time co-simulation of power systems," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 6717–6721.

[97] FEIN. (2022). *VILLASframework—FEIN Aachen*. [Online]. Available: https://fein-aachen.org/en/projects/villas-framework/#publications

[98] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily, and J. Fuller, "Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework," in *Proc. Workshop Modeling Simulation Cyber-Phys. Energy Syst. (MSCPES)*, Apr. 2017, pp. 1–6.

[99] W. Wang, X. Fang, H. Cui, F. Li, Y. Liu, and T. J. Overbye, "Transmission-and-distribution dynamic co-simulation framework for distributed energy resource frequency response," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 482–495, Jan. 2022.

[100] A. K. Bharati, V. Ajjarapu, W. Du, and Y. Liu, "Role of distributed inverter-based-resources in bulk grid primary frequency response through HELICS based SMTD co-simulation," *IEEE Syst. J.*, vol. 17, no. 1, pp. 1071–1082, Mar. 2023.

[101] V. Venkataramanan, P. S. Sarker, K. S. Sajan, A. Srivastava, and A. Hahn, "Real-time federated cyber-transmission-distribution testbed architecture for the resiliency analysis," *IEEE Trans. Ind. Appl.*, vol. 56, no. 6, pp. 7121–7131, Nov. 2020.

[102] A. Boumaiza, M. Z. C. Wanik, and A. Sanfilippo, "Modeling a blockchain-enabled transactive energy system for community microgrids," in *Proc. IEEE 16th Int. Conf. Compat., Power Electron., Power Eng. (CPE-POWERENG)*, Jun. 2022, pp. 1–6.

[103] A. Meghwani, S. C. Srivastava, and A. Srivastava, "Development of real-time distribution system testbed using co-simulation," in *Proc. 21st Nat. Power Syst. Conf. (NPSC)*, Dec. 2020, pp. 1–6.

[104] A. K. Bharati and V. Ajjarapu, "SMTD co-simulation framework with HELICS for future-grid analysis and synthetic measurement-data generation," *IEEE Trans. Ind. Appl.*, vol. 58, no. 1, pp. 131–141, Jan. 2022.

[105] W. Wang, M. Cai, X. Fang, and C. Irwin, "Impact of open communication networks on load frequency control with plug-in electric vehicles by cyber-physical dynamic co-simulation," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Jan. 2023, pp. 1–5.

[106] J. Zhang, J. Daily, R. A. Mast, B. Palmintier, D. Krishnamurthy, T. Elgindy, A. Florita, and B.-M. Hodge, "Development of HELICS-based high-performance cyber-physical co-simulation framework for distributed energy resources applications," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Nov. 2020, pp. 1–5.

[107] W. Wang, X. Fang, and A. Florita, "Impact of DER communication delay in AGC: Cyber-physical dynamic co-simulation," in *Proc. IEEE 48th Photovoltaic Specialists Conf. (PVSC)*, Jun. 2021, pp. 2616–2620.

[108] D. Cutler, T. Kwasnik, S. Balamurugan, T. Elgindy, S. Swaminathan, J. Maguire, and D. Christensen, "Co-simulation of transactive energy markets: A framework for market testing and evaluation," *Int. J. Electr. Power Energy Syst.*, vol. 128, Jun. 2021, Art. no. 106664.

[109] B. A. Bhatti, "A game theoretic-based transactive energy framework for distributed energy resources," Ph.D. dissertation, Bradley Dept. Elect. Comput. Eng., Virginia Tech, Blacksburg, VA, USA, 2021.

[110] N. Li, W. Hou, and S. E. Ghoreyshipour, "A secured transactive energy management framework for home AC/DC microgrids," *Sustain. Cities Soc.*, vol. 74, Nov. 2021, Art. no. 103165.

[111] D. Carta, M. Weber, P. Glücker, T. Pesch, V. Hagenmeyer, A. Benigni, and U. Kühnapfel, "VILLASnode-based co-simulation of local energy communities," in *Proc. Open Source Modeling Simulation Energy Syst. (OSMSES)*, Apr. 2022, pp. 1–6.

[112] Y. Liu, T. J. Overbye, W. Wang, X. Fang, J. Wang, H. Cui, and F. Li, "Transmission-distribution dynamic co-simulation of electric vehicles providing grid frequency response," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Jul. 2022, pp. 1–5.

[113] Y. Liu, R. Huang, W. Du, A. Singhal, and Z. Huang, "High-performance transmission and distribution co-simulation with 10,000+ inverter-based resources," in *Proc. IEEE Ind. Appl. Soc. Annu. Meeting (IAS)*, Oct. 2022, pp. 1–5.

[114] L. Barbierato, E. Pons, E. F. Bompard, V. S. Rajkumar, P. Palensky, L. Bottaccioli, and E. Patti, "Exploring stability and accuracy limits of distributed real-time power system simulations via system-of-systems cosimulation," *IEEE Syst. J.*, vol. 17, no. 2, pp. 3354–3365, Jun. 2023.

[115] M. O. Faruque, M. Sloderbeck, M. Steurer, and V. Dinavahi, "Thermo-electric co-simulation on geographically distributed real-time simulators," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2009, pp. 1–7.

[116] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "A predictor-based framework for delay compensation in networked closed-loop systems," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2482–2493, Oct. 2018.

[117] K. Kobayashi and Y. Uchimura, "Model based predictive control for a system with long time delay," in *Proc. IEEE 15th Int. Workshop Adv. Motion Control (AMC)*, Mar. 2018, pp. 581–586.

[118] K. Kobayashi, A. Minohara, and Y. Uchimura, "Network-based control with state prediction for time-delayed system," in *Proc. IEEE 27th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2018, pp. 982–987.

[119] R. Sakthivel, S. Santra, and B. Kaviarasan, "Resilient sampled-data control design for singular networked systems with random missing data," *J. Franklin Inst.*, vol. 355, no. 3, pp. 1040–1072, Feb. 2018.

[120] Y. Deng, X. Zhang, N. Im, G. Zhang, and Q. Zhang, "Model-based event-triggered tracking control of underactuated surface vessels with minimum learning parameters," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4001–4014, Oct. 2020.

[121] P. Loon Tang and C. W. de Silva, "Compensation for transmission delays in an Ethernet-based control network using variable-horizon predictive control," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 4, pp. 707–718, Jul. 2006.

[122] C.-L. Lai and P.-L. Hsu, "The butterfly-shaped feedback loop in networked control systems for the unknown delay compensation," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1746–1754, Aug. 2014.

[123] C. Wang, C.-Y. Lin, and M. Tomizuka, "Statistical learning algorithms to compensate slow visual feedback for industrial robots," *J. Dyn. Syst., Meas., Control*, vol. 137, no. 3, Mar. 2015, Art. no. 031011.

[124] W. Li, L. Ding, H. Gao, and M. Tavakoli, "Haptic tele-driving of wheeled mobile robots under nonideal wheel rolling, kinematic control and communication time delay," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 336–347, Jan. 2020.

[125] T. Q. Dinh, J. Marco, D. Greenwood, K. K. Ahn, and J. I. Yoon, "Data-based predictive hybrid driven control for a class of imperfect networked systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 5187–5199, Nov. 2018.

[126] M. Saków and K. Marchelek, "Model-free and time-constant prediction for closed-loop systems with time delay," *Control Eng. Pract.*, vol. 81, pp. 1–8, Dec. 2018.

[127] T. Ersal, M. Brudnak, A. Salvi, Y. Kim, J. B. Siegel, and J. L. Stein, "An iterative learning control approach to improving fidelity in Internet-distributed hardware-in-the-loop simulation," *J. Dyn. Syst., Meas., Control*, vol. 136, no. 6, Nov. 2014, Art. no. 061012.

ELUTUNJI BURAIMOH (Member, IEEE) received the B.Tech. (Hons.) degree in electronic and electrical engineering from the Ladoke Akintola University of Technology, Nigeria, in 2012, the M.Sc. degree in electronic and electrical engineering from Obafemi Awolowo University, Nigeria, in 2016, and the D.Eng. degree from the Durban University of Technology, South Africa, in 2021. He is currently pursuing the Ph.D. degree with the Holcombe Department of Electrical and Computer Engineering, Clemson University. He is a Research Assistant with the Virtual Prototyping of Autonomy-Enabled Ground Systems (VIPR-GS) Centre, Clemson University. His research interests include geographically distributed real-time co-simulation of power systems, military-based vehicle-to-grid and vehicle-to-vehicle microgrids, advanced power management, reinforcement learning, and energy management. He is a registered Engineer of the COREN, a Corporate Member of the NSE, a Candidate for ECSA, and a member of SAE.

GOKHAN OZKAN (Senior Member, IEEE) received the B.S. degree from the Department of Electrical Education, Marmara University, in 2006, the B.Sc. (as a valedictorian) and M.S. degrees from the Department of Energy System Engineering, Erciyes University, in 2014 and 2016, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Florida State University, in 2019. In 2020, he joined Clemson University as a Postdoctoral fellow, where he has been a Research Assistant Professor with the Holcombe Department of Electrical and Computer Engineering, since 2021. He is currently the Research Lead with the Dominion Energy Innovation Center, North Charleston, SC, USA, and the Manager of the Real-Time Control and Optimization Laboratory (RT-COOL). His main research interests include power systems and DER integration, model predictive control in power converters and drives, active thermal management and fault-tolerant control, distributed controls, power and EM, real-time power distribution system modeling, simulation, co-simulations, and hardware-in-the-loop instantiation.

LAXMAN TIMILSINA (Graduate Student Member, IEEE) received the B.E. degree in electrical engineering and the M.Sc. degree in distributed generation in electrical engineering (with a focus on the improvement of the power qualities of an isolated microgrid) from the Institute of Engineering, Tribhuvan University, Nepal, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Clemson University, under the supervision of Dr. Christopher S. Edrington. He is a Graduate Research Assistant with the Real-Time Control and Optimization Laboratory (RT-COOL), Holcombe Department of Electrical and Computer Engineering (ECE), Clemson University. His research interests include electric vehicles, the integration of renewable energy, and smart grids.

PHANI KUMAR CHAMARTHI (Member, IEEE) received the joint M.Tech. and Ph.D. degrees with a specialization in power electronics and power systems from the Department of Electrical and Electronics Engineering, Indian Institute of Technology Bombay, Mumbai, India, in 2018. He is currently a Postdoctoral Fellow with Clemson University, USA. His research interests include new single-stage transformerless inverter topologies, multilevel converter topologies, space vector control of multilevel converters, and new modulation and control strategies of multilevel converters with a focus on renewable energy applications.

BEHNAZ PAPARI (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Bushehr, Iran, in 2008, the M.Sc. degree in electrical engineering from the K. N. Toosi University of Technology, Iran, in 2011, and the Ph.D. degree in electrical and computer engineering from Florida State University, USA, in 2018. Prior to joining the Department of Automotive Engineering and Electrical and Computer Engineering, Clemson University, in Spring 2022, she was an Assistant Professor with the Department of Marine Engineering Technology, Texas A&M University, and an Assistant Professor of practice with the Energy Production and Infrastructure Center (EPIC), UNC Charlotte, for three years, where she taught at undergraduate and graduate levels. Her background is in power systems controls with an emphasis on modeling, analysis, planning, and AI-based optimization. Her current research interests include distributed controls and secure control frameworks under uncertainty, applications on energy system modeling, and stochastic optimization, electric vehicles, including electric and hybrid-electric, and ESSs.

CHRISTOPHER S. EDRINGTON (Senior Member, IEEE) received the B.S. degree in engineering from Arkansas State University, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from the Missouri University of Science and Technology, in 2001 and 2004, respectively. He was a GAANN Fellow and IGERT Fellow with the Missouri University of Science and Technology. From 2004 to 2007, he was an Assistant Professor of electrical engineering with the College of Engineering, Arkansas State University. From 2007 to 2019, he held the ranks of an Assistant Professor, an Associate Professor, and a Full Professor of electrical and computer engineering with the FAMU-FSU College of Engineering and was the lead for the Energy Conversion and Integration Trust, Center for Advanced Power Systems, Florida State University. In 2019, he joined Clemson University as the Warren H. Owen Distinguished Professor of Electrical and Computer Engineering where he leads the Real-Time Control and Optimization Laboratory (RT-COOL). In addition, he is the Associate Director for Industry Relations with the Virtual Prototyping of Ground Systems (VIPR-GS) Center, Clemson University. His research interests include modeling, simulation, optimization, and control of electromechanical drive systems; applied power electronics; integration of renewable energy, storage, and pulse power loads; and HIL (hardware in the loop) VV&A experimentation.