6th International Conference on Advances in Experimental Structural Engineering
11th International Workshop on Advanced Smart Materials and Smart Structures Technology
August 1-2, 2015, University of Illinois, Urbana-Champaign, United States

Development and Evaluation of a Geographically Distributed Real-Time Hybrid Simulation Platform

A.I. Ozdagli[1], S.J. Dyke[2]

1 *Research Fellow, Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, United States.*
E-mail: aliirmak@gmail.com
2 *Professor, School of Mechanical Engineering, Purdue University,West Lafayette, IN, United States.*
E-mail: sdyke@purdue.edu

ABSTRACT

Real-time hybrid simulation (RTHS) has become a recognized methodology for isolating and evaluating performance of critical structural components under potentially catastrophic events such as earthquakes. While RTHS is efficient in its utilization of equipment and space compared to traditional testing methodologies, the laboratory resources may not always be available in a single testing facility to employ large scale experiments. Consequently, use of geographically distributed RTHS (dRTHS) platform, where multiple RTHS setups located at numerous test sites are connected to each other through the Internet, becomes essential. dRTHS is delay-sensitive by nature. Therefore, a fixed transmission rate with minimum jitter and latency in the network traffic should be maintained during a dRTHS experiment. A Smith predictor can compensate network delays, but requires use of a known dead time for its optimal operation. In this study, a novel dRTHS platform running on xPC/MATLAB framework is proposed with the following features: (i) Use of User Datagram Protocol (UDP) to satisfy information exchange over the Internet between test sites with minimum time delay and jittering; (ii) Smith predictor to compensate network delay; (iii) delay estimator to determine the network time delay on-the-fly for the optimal operation of Smith predictor; and (iv) a dejitterizer buffer to minimize network jittering. To demonstrate the effectiveness of the platform, global responses of a three story structure tested on the shake table are compared to dRTHS results. It is shown that dRTHS can be considered as a viable option where facility resources are not available to conduct shake table tests or single-site RTHS.

KEYWORDS: *distributed real-time hybrid simulation, Smith predictor, delay compensation, User Datagram Protocol*

1. INTRODUCTION

In the last decades, hybrid simulation (HS) has gained a lot of interest from the earthquake engineering community. Essentially, HS is used as an efficient testing method to evaluate performance of the structures subjected to seismic excitation. In this method, the structure is split into physical substructure, which is usually challenging to model, and numerical substructure, which is money- and space-wise ineffective to construct but comparatively easy to model. With the help of advancements in hardware and software, hard real-time computing extended capabilities of HS towards real-time hybrid simulation (RTHS) to explore more realistic simulation cases of rate-dependent structural systems. Although HS scheme is efficient in utilization of equipment and space, laboratory resources may not always be optimized to employ more complex testing plans involving multiple apparatus and large-scale systems. As a result, distributed systems, capable of connecting multiple HS setups located at numerous geographically dispersed facilities, become essential.

The research on distributed HS (dHS) has begun with Multi-Site Online Simulation Testbed (MOST) where two physical substructures located at University of Illinois at Urbana-Champaign (UIUC) and University of Colorado, Boulder (CU) were linked to a numerical model simulated by National Center for Supercomputing Applications (NCSA), also located at UIUC through a message passing protocol, NEESGrid Teleoperation Control Protocol (NTCP) (NEESgrid, 2003; Spencer et al., 2004). Other notable dHS frameworks are UI-SIMCOR, a universal middleware that establishes a coordinator between multiple sites and numerical simulation developed by Kwon et al. (2005) and OpenFresco, an interface to connect OpenSees FEM to a physical substructure developed by Takahashi et al. (2005) and Schellenberg et al. (2009). Generally, for all dHS frameworks, real-time exchange of the command and feedback signals between the facilities running numerical and physical substructures is not an essential requirement. Efforts are more focused on reliable transmission.

Compared to dHS, there is limited research on distributed RTHS (dRTHS). The first attempt was made by Kim et al. (2012) by distributing physical and numerical structures between University of Connecticut (UCONN) and UIUC using QUARC Real-time Control Software. Transmission Control Protocol (TCP) is used as the message passing method. In addition, to compensate network delays, a Smith predictor is utilized in this scheme. Another framework, Interdependent Channel - Distributed Hybrid Testing (IC-DHT) developed by Ojaghi et al. (2014) was employed as a middleware between Oxford and Bristol Universities operating numerical simulations and physical substructure testing. IC-DHT operates in soft real-time over a Data Handling Protocol (DHP) which is a higher level protocol encapsulating User Datagram Protocol (UDP). The network delays were treated by compensating the command input with polynomial extrapolation before sending to the actuator site.

The available dRTHS middleware is either proprietary or closed-source. Furthermore, complex control mechanisms to compensate network delays implemented within the middleware prohibit dRTHS from running at high rates. This paper demonstrates a method for researchers to design their own platforms based on a new dRTHS platform proposed here. The data transmission logic of the platform is built on UDP which is the only essential control element required for packet exchange. Moreover, to estimate and compensate nondeterministic network time delay on-the-fly, a Smith predictor-based time delay compensation scheme is introduced. The performance of the proposed dRTHS platform is validated by comparing results to shake table responses of a three story test structure equipped with MR damper. It has been shown that successful dRTHS can be achieved at 5000 Hz.

2. IMPLEMENTATION OF DRTHS PLATFORM

In a common HS setting, physical and numerical substructures representing the actual structure are interfaced via a transfer device, usually an actuator connected to the physical substructure. In a dRTHS, each of these substructures are distributed among two or more laboratories, each operated at real-time. In addition to the transfer interface, the Internet Protocol (IP) is utilized as the standard medium for data exchange. An example overview of the architecture is shown in Fig 2.1. In this scheme, physical substructure is located at Intelligent Infrastructural Systems Laboratory (IISL) at Purdue University, whereas numerical substructure is simulated at Smart Structures Technology Laboratory (SSTL) at University of Illinois, Urbana-Champaign (UIUC). Both simulator and actuator controller run on real-time target systems (xPC).

The RT target at IISL (xPC1), responsible for compensation of actuator dynamics to realize the received desired command coming from SSTL target, collects the measured force from the physical substructure and sends to remote target over the Internet. The RT target located at SSTL (xPC2) simulates the numerical model, generates the desired displacement and sends to IISL and receives the feedback force from IISL. In addition, the network time delays are also compensated within this site.

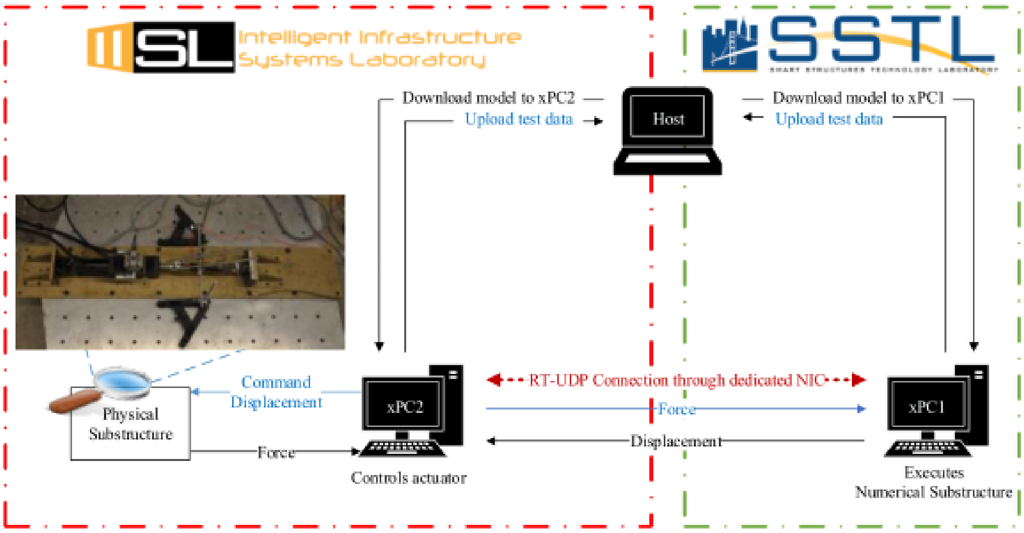

Figure 2.1 dRTHS system architecture

Finally, a host computer is tasked to compile simulation and download compiled application to related target and download simulation results.

In the following sections, important components of the proposed platform, i.e. data transmission and network delay estimator are discussed in detail.

2.1. Target-to-Target Data Transmission

There are two prime protocols, called Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) to route the traffic between network devices. Essentially, TCP is a connection-oriented protocol that handles traffic flow, connection reliability and congestion control, by default. Although TCP is considered as de facto protocol for applications requiring reliable connection, it should be noted that this protocol does not guarantee low network delays to large overhead. Alternative to TCP, UDP, as a connectionless protocol, does not have a connection quality control and has less overhead. Thus, UDP is more suitable for real-time applications requiring high transmission rates such as VoIP.

By nature, a typical dRTHS application inherits all challenges present in RTHS. Namely, like any RTHS applications, dRTHS is also sensitive to time delay. On the other hand, the Internet is not optimized to sustain high-rate real-time communication. Therefore, a lightweight protocol promising less network delays should be preferred for the sake of the test performance. Consequently, UDP is selected for the proposed platform.

There are many real-time target platforms that might be used for performing RTHS. While the architecture described here can be applied to many of those, xPC is selected for this study, since MATLAB provides Real-time UDP (RT-UDP), readily available UDP module for real-time distributed testing (see Fig. 2.2). RT-UDP block uses a dedicated Network Interface Card (NIC) for target-target communication to utilize a reliable connection that does not share bandwidth with the host computer.

Figure 2.2 xPC RT-UDP blocks

When a data package is sent over the Internet, it is likely that each packet will follow a different path along the web to reach its destination. During a real-time transmission, it is expected that not each packet will arrive to the remote computer on time due to differences in data flow rate. Fortunately, RT-UDP block support a buffering feature to eliminate jitter in the transmission. Basically, buffer is a routine that compensates differences in data flow rate, by queuing incoming packets in first in-first out (FIFO) mechanism into a temporary medium, as illustrated in Fig. 2.3. When a buffered package is requested, it is removed from the queue. It should be noted that use of any buffer will also add delay to the dRTHS.

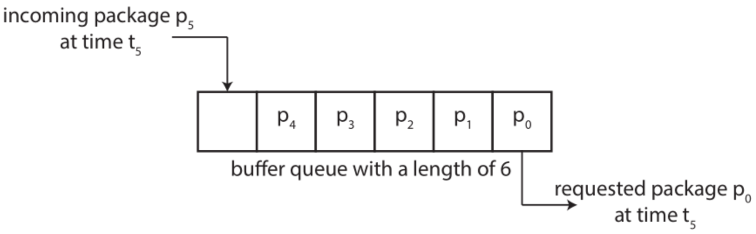

Figure 2.3 xPC buffering application

2.2. Network Delay Compensation

One of the challenges for real-time network applications is the presence of network time delay. Mainly, network delay, also known as latency, is the length of dead time it takes from a data package leaving the local target until being received by the remote computer. The network delay is usually in the order of tens of milliseconds.

The network time delays in the dRTHS platform can be idealized as given in Fig. 2.4. Here, τ_1 and τ_2 are the transmission delays due to Internet for the inbound and outbound packets between SSTL and IISL sites, respectively. These transmission delays can be regarded as dead time, hence, can be treated with Smith predictor.

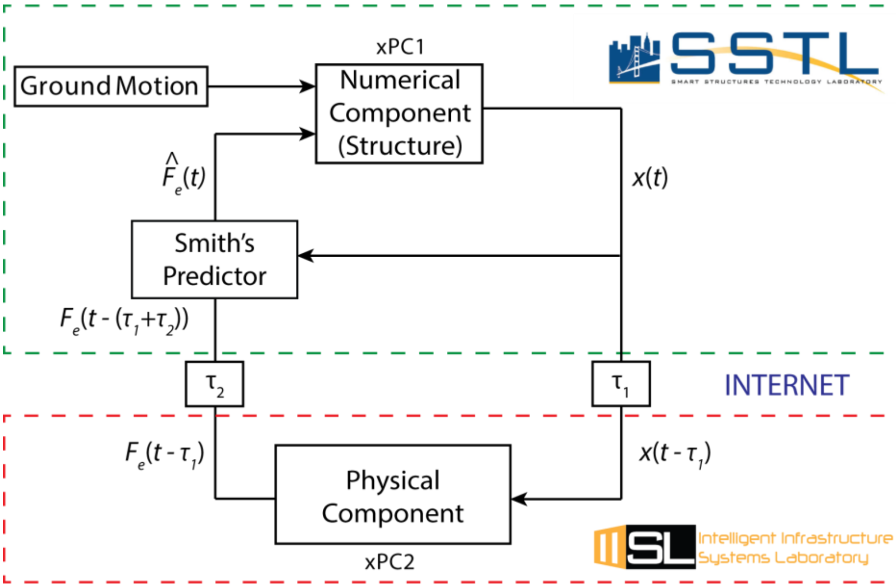

Figure 2.4 dRTHS architecture including network time delays

A Smith predictor control structure proposed by Kim et al. (2012) is illustrated in Fig. 2.5. The Smith predictor uses an internal model of the physical substructure to predict the delay-free and delayed model responses, $F_a(t)$ and $F_a(t-(\hat{\tau}_1+\hat{\tau}_2))$, respectively. $\hat{\tau}_1+\hat{\tau}_2$ is the estimated round-trip delay prior to the testing, $F_e(t)$ is physical substructure plant force and $\hat{F}_e(t)$ is the delay-compensated plant force.

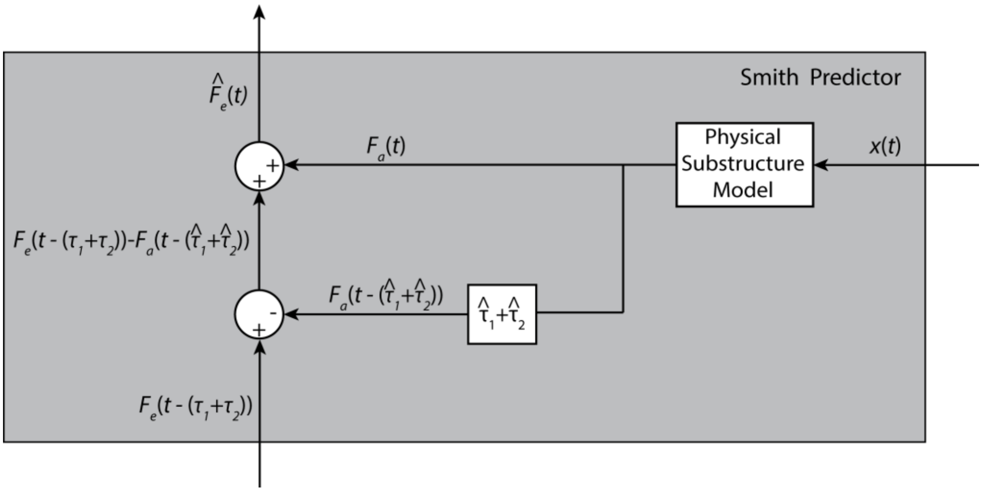

Figure 2.5 Implementation of Smith predictor

2.3. Time Delay Estimation

The Smith predictor requires the dead time to be known prior to testing for the optimal operation. However, it may not always be possible to estimate the network time delay accurately since network conditions change continuously. The performance of the Smith predictor could be enhanced if network time delay could be determined beforehand. The estimator proposed herein has the ability to calculate the time delay between the targets by simply observing the time difference between transmitting and receiving signals.

RT-UDP Receive block provides two output ports for processing incoming data. Using the outputs from these ports, arrival of packet can be checked during simulation time. Use of this feature provides the basis of the delay estimator. An algorithmic flowchart of the estimator is given in Fig. 2.6. Fundamentally, in this algorithm, the local target sends a reference signal to the remote target. Meanwhile a time step counter is initiated. As soon as the reference signal is delivered, the remote target loops the signal back to local target. Eventually, the time step difference between the real data and looped-back data is determined by counting the counter tick. This time step difference is in fact the dead time required by the Smith predictor. This algorithm is only performed one time at the beginning of the test to estimate the initial network latency.

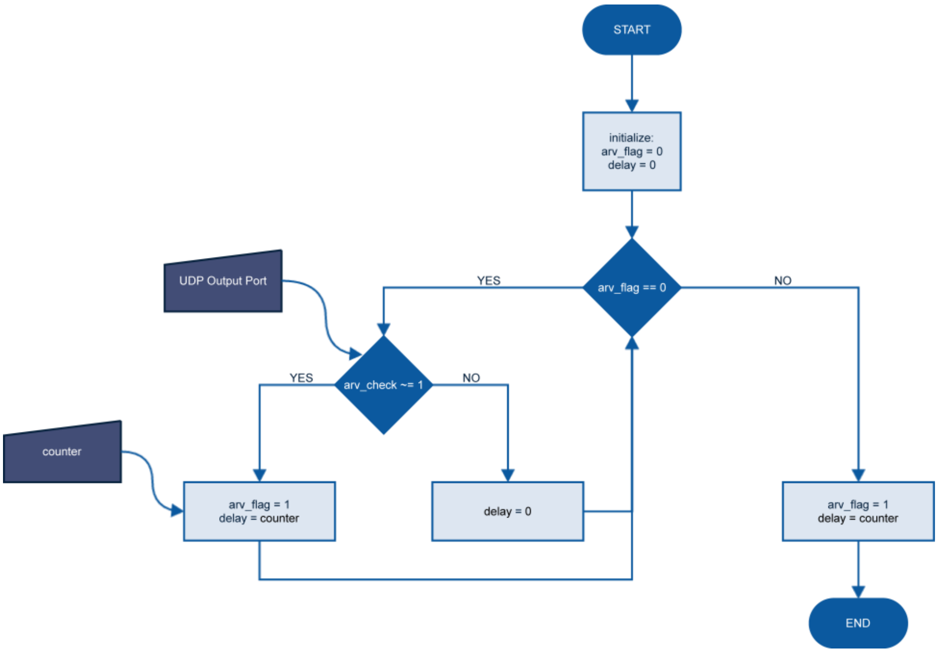

Figure 2.6 Implementation of time delay estimator

Procedurally, the algorithm first initialize an arrival flag (***arv_flag***) and time delay (***delay***) variables. For the first time step, the algorithm will check if the size of the packet is zero (**arv_check**) until data arrives. When first packet is received (***arv_check = 1***), ***arv_flag*** will be raised and delay will be fixed to the number of the current time step produced by a counter. Once ***arv_flag*** is raised, delay will persistently contain the already estimated time step number until the end of simulation. This algorithm is applied at numerical simulation site to the inbound feedback force coming from physical substructure, before compensated by Smith predictor control structure block. The output of this algorithm, which is the estimated time delay (in fact, it is the true time delay of the network), is fed to the Smith predictor.

3. VALIDATION OF DRTHS ARCHITECTURE

To demonstrate effectiveness of the proposed dRTHS platform, responses of a three story structure, with an MR damper at its first floor, tested on the shake table at Harbin Institute of Technology, China are compared to dRTHS results (see Fig. 3.1). The test structure has a base plan with dimensions 1.84m by 2.04m and is 3.6m tall. The identified natural frequencies of the structure are calculated 2.88Hz, 8.10Hz and 12.34Hz.

The test structure is parted into (i) MR damper serving as physical substructure and (ii) MCK model of the bare structure representing the numerical part. The physical substructure is located at IISL whereas numerical substructure is simulated at SSTL. The sampling and transmission rates are chosen 5000Hz for both sites. To minimize the jittering and sustain a lossless transmission, a buffer of 300 and 10 time steps is used at IISL and SSTL, respectively. To compensate the network delay, Smith predictor with time delay estimation block is implemented on SSTL target. The Smith predictor uses the Bouc-Wen model of the MR damper as the plant model. The round-trip delay is estimated by network time delay estimator as ~ 400 time steps or ~80msec.

Figure 3.1 Test structure on the shake table

To drive the MR damper, the first floor displacement generated by the numerical model at SSTL is sent to IISL as the command signal over the Internet. Robust Integrated Actuator Control to compensate actuator dynamics and realize the command displacement with high fidelity is implemented at IISL (Ou et al. 2014). The reaction force produced by MR damper is looped back to SSTL. Additionally, to control the MR damper in semi-active control fashion, a control voltage command is sent from SSTL to IISL. Clipped-optimal control is selected for vibration control.

The semi-active control case is considered as the basis of the comparison. The ground motion to the structure is chosen El Centro earthquake. In Fig. 3.2, the acceleration and displacement responses of shake table (ST) and dRTHS cases are compared in time domain. In addition, in Table 3.1, a quantitative analysis of the comparisons is provided.

Table 3.1 Evaluation criteria for ST–dRTHS comparison

Error Case	Peak Displacement Error [%]	Peak Acceleration Error [%]	RMS Displacement Error [%]	RMS Acceleration Error [%]
First Floor	13.58	21.68	5.18	5.77
Second Floor	6.93	15.57	4.33	3.63
Third Floor	9.17	10.86	4.41	3.14

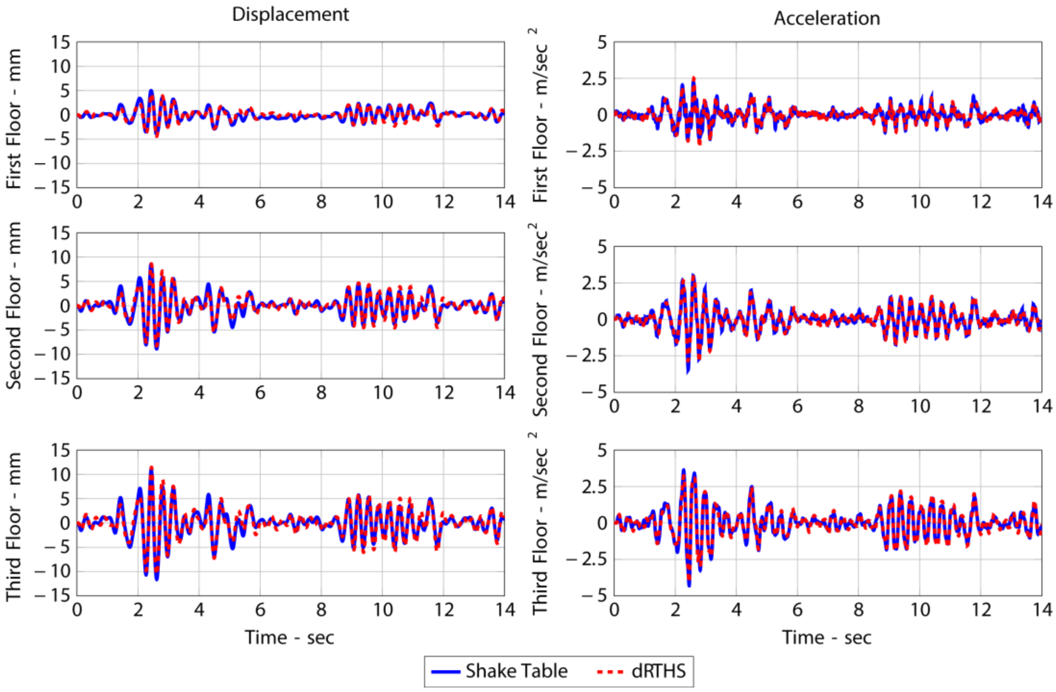

Figure 3.1 El Centro earthquake response comparison in time domain for ST–dRTHS case

5. CONCLUSION

In this paper, a new dRTHS platform is proposed that utilizes low overhead UDP as the main data exchange protocol. The network delay occurring due to the nature of the Internet is compensated with a Smith predictor. Since this predictor requires a known dead time for its optimal operation, a delay estimator algorithm, which is capable of calculating the network delay prior to the test, is introduced. Finally, performance of the proposed platform is evaluated by comparing global responses of a three story structure tested on the shake to dRTHS results. Comparisons showed that dRTHS operating at high sampling and transmission rates can be considered as a valid testing environment for geographically distributed laboratories having limited experimental resources, alternative to shake table testing.

ACKNOWLEDGEMENT

The financial support of this research is provided in part by the U.S. National Science Foundation under Grant CMMI-1011534 (NEESR), ACI-1148255 and Purdue International Programs under the Sohmen Fund.

REFERENCES

1. NEESgrid (2003). *The MOST Experiment - Whitepaper 1.0*. http://www.neesgrid.org/.
2. Spencer, B. F., Finholt, T., Foster, I., Kesselman, C., Beldica, C., Futrelle, J., Gullapalli, S., Hubbard, P., Liming, L., Marcusiu, D., Pearlman, L., Severance, C., and Yang, G. (2004). NEESgrid: A Distributed Collaboratory for Advanced Earthquake Engineering Experiment and Simulation. *13th World Conference on Earthquake Engineering*, Vancouver, Canada.
3. Kwon, O. S., Nakata, N., Elnashai, A., and Spencer, B. F. (2005). A Framework for Multi-Site Distributed Simulation and Application to Complex Structural Systems. *Journal of Earthquake Engineering*. **09:05**, 741-753.
4. Takahashi, Y., and Fenves, G. L. (2005). Software framework for distributed experimental–computational simulation of structural systems, *Earthquake Engineering & Structural Dynamics*, **35:3**, 267—291.
5. Schellenberg, A. H., Mahin, S. A., and Fenves, G. L. (2009). *Advanced Implementation of Hybrid Simulation - PEER 2009/104*, PEER, University of California, Berkeley. http://www.neesgrid.org/mini-most/Mini_MOST_requirements_revised3.pdf.
6. Kim, S. J., Christenson, R., Phillips, B., and Spencer, B. F. (2012). Geographically Distributed Real-Time Hybrid Simulation of MR Dampers for Seismic Hazard Mitigation. 20th Analysis and Computation Specialty Conference. Chicago, Illinois, United States.
7. Ojaghi, M., Williams, M. S., Dietz, M. S., Blakeborough, A. and Lamata M. I. (2014). Real-time distributed hybrid testing: coupling geographically distributed scientific equipment across the Internet to extend seismic testing capabilities. Earthquake Engineering & Structural Dynamics. **43**, 1023–1043.
8. Ou G., Ozdagli Ai I., Dyke S. J., and Wu B. (2015). Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation. Earthquake Engineering & Structural Dynamics, **44**, 441–460.