Towards Configurable Real-Time Hybrid Structural Testing: A Cyber-Physical Systems Approach

Terry Tidwell[1], Xiuyu Gao[2], Huang-Ming Huang[1],
Chenyang Lu[1], Shirley Dyke[2], Christopher Gill[1]
[1]*Department of Computer Science and Engineering*
{ttidwell, hh1, lu, cdgill}@cse.wustl.edu
[2]*Department of Mechanical, Aerospace and Structural Engineering*
xg2@cec.wustl.edu, sdyke@wustl.edu
Washington University, St. Louis, MO, USA

Abstract

Real-time hybrid testing of civil structures represents a grand challenge in the emerging area of cyber-physical systems. Hybrid testing improves significantly on either purely numerical or purely empirical approaches by integrating physical structural components and computational models. Actuator dynamics, complex interactions among computers and physical components, and computation and communication delays all hamper the ability to conduct accurate tests. To address these challenges, this paper presents initial work towards a Cyber-physical Instrument for Real-time hybrid Structural Testing (CIRST). *CIRST aims to provide two salient features: a highly configurable architecture for integrating computers and physical components; and system support for real-time operations in distributed hybrid testing. This paper presents the motivation of the CIRST architecture and preliminary test results from a proof-of-concept implementation that integrates a simple structural element and simulation model. CIRST will have broad impacts on the fields of both civil engineering and real-time computing. It will enable high-fidelity real-time hybrid testing of a wide range of civil infrastructures, and will also provide a high-impact cyber-physical application for the study and evaluation of real-time middleware.*

1. Introduction

Sensing and actuation devices that also can compute and communicate are increasingly crucial to the evaluation and optimized operation of modern structural systems. Using computing and communication (*cyber*) elements in conjunction with mechanical and structural (*physical*) elements to manage, monitor and control the behavior of such systems represents a fundamental change in how to interact with these structures. While high-fidelity validation is critical to the acceptance of structural monitoring and control systems, in many applications testing numerous possible scenarios of a new structural control device or a new system is not feasible due to the *cost* and *time* required for such a comprehensive test. The *scale* of civil structures also makes it infeasible to test them fully through empirical techniques alone, due to the tremendous sizes of the structures and requirements regarding loading. For instance, the performance of a new vibration suppression system (e.g., for earthquake or hurricane mitigation) usually cannot be validated at full scale prior to its implementation on a large bridge.

The cost associated with empirical testing prohibits performing more than a few representative tests at small scales relative to the massive sizes of the structures. However, reduced scale testing cannot always capture important behaviors of the full scale structure, even if scaling effects have been carefully considered. Numerical simulation is therefore an equally important technique in modern structural analysis, and has benefited significantly by leveraging hardware innovations that offer improved computational capabilities. However, experimental validation is still essential to examine the underlying assumptions made by the numerical models, especially considering the existence of highly nonlinear elements under extreme dynamic loading.

Hybrid testing, which integrates both physical components of the structure of interest and computational models of other known structural components, thus improves significantly on either purely numerical or purely empirical approaches. Due to the lack of real-time support for hybrid testing, however, hybrid testing at a slow (a.k.a. *pseudodynamic*) time scale is the state of the art [1], [2], [3], [4]. Unfortunately, testing at such time scales may not reveal critical dynamic system features, necessitating a real-time approach. The leap from slow time scales to real-time raises significant research challenges such as real-time coordination, fault tolerance and control stability. These issues in turn require the development of new kinds of cyber-physical instruments for real-time hybrid testing. In this paper we present our vision and initial work towards such an instrument, a *Cyber-physical Instrument for Real-time hybrid Structural Testing* (CIRST).

The contributions of this paper include: (1) the identification of key limitations of existing approaches to hybrid structural testing; (2) a vision for how CIRST can extend the state of the art to overcome those limitations; and (3) a case study that validates our approach for a simple hybrid structural testing example, and illustrates the importance of real-time application and platform design for studying real-time scheduling, resource management, and other crucial research topics in this domain.

In Section 2 we discuss the state of the art in both the numerical techniques that drive the computational models during a hybrid test, and the architectures of the platforms that run them. In Section 3 we describe how common themes among those current approaches suggest methods for formalizing reusable solutions in hybrid test development, deployment and execution. In Section 4 we present a case study comparing our proof of concept system with a leading current platform. Finally, in Section 5 we summarize the work and its expected impacts.

2. Limitations of Current Approaches

Traditional approaches for studying structures include numerical simulations and physical testing, though as we noted in Section 1 relying on either approach alone has significant limitations. To overcome the limitations of both approaches, hybrid testing has gained significant attention as a promising means to integrate the benefits of both simulation and physical testing. In this section we examine the state of the art in hybrid testing and note limitations of current approaches which motivate our vision for CIRST.

2.1. Pseudo-dynamic Testing

The concept of pseudo-dynamic (PSD) testing was developed and implemented in the 1980's to enable meaningful testing of large structures [3], [5]. The approach is to divide the test structure into two portions: a portion involving computation and communication (we term this the *cyber* portion of the system), and a *physical* portion involving sensors, actuators, and the test specimen. While most of the structure is numerically modeled and tested on a computational platform, critical components (or those with complex behaviors to be studied) are tested physically to develop a better understanding of their behavior. The common procedure is: within each time step the responses of the numerical substructure are calculated and applied to the physical substructure with hydraulic actuators, and then measurements of the responses from the physical specimen are fed back to calculate the responses of the computational elements at the next time step. Due to the challenges of achieving end-to-end real-time performance and coordination of physical and cyber components, PSD testing is typically non-real-time, in that the test is often significantly slower than the dynamics of the original structure. This approach is shown in the block diagram in Figure 1.

PSD testing has been implemented and proven to be effective for certain classes of structures and behaviors.

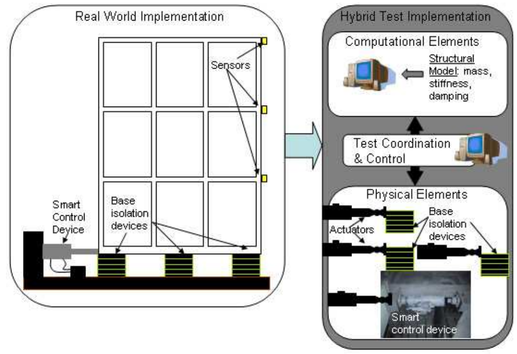

Figure 1. From real world implementation to hybrid test

However, *real-time* guarantees are necessary when rate dependent physical elements are present in the test, or when coordination and synchronization of cyber elements is necessary for validation. *Real-time Hybrid Testing (RTHT)* requires the use of computational, electrical, hydraulic and structural components, each with its own associated dynamics that need to be understood and properly compensated, if necessary. Currently the ability to perform real-time hybrid testing is often hindered by the dynamics of actuators and sensors, by the complex interactions between these elements and the physical specimens, and by lags between when the latest approaches to real-time communication and computation become available and when they are integrated into these testing procedures. Furthermore, although purely computational simulations are not temporally constrained, hybrid design, modeling, analysis, and testing involving both physical and cyber components imposes timing constraints on all system elements. Real-time coordination of elements in turn requires precise temporal synchronization, timely interactions and predictive modeling techniques.

Summary: Managing the complexities induced by RTHT over and above PSD testing requires increasingly sophisticated techniques and tools. CIRST will extend the state of the art by easing the design and deployment of hybrid testing experiments and of platforms for running these experiments.

2.2. Compensation for Dynamics

Due to the complex dynamics of the test apparatus and the interactions between it and the tested structure, time delays are inevitably present in each component (e.g. numerical computation, data communication, sensor measurement and actuation devices) of a RTHT system. Without proper consideration of these dynamics, they can introduce errors, e.g., preventing a desired displacement from being applied properly to the test structure. Among all the contributions to this form of experimental error, the hydraulic actuator dynamics and their interaction with the test structure [6] appear to be the most dominant concern [7]. Such error can be significant and may cause instabilities in the experimental components without proper compensation.

Appropriate compensation for these complex dynamics is essential and has been investigated by several researchers. One widely accepted compensation scheme [8] is to predict the displacement after the delay and input it as the command signal to the actuator. In this approach an n^{th} order polynomial function is adopted to extrapolate the predicted value based on the current information and a few earlier displacement values. An intrinsic assumption in this approach is that the actuator has a pure time delay that is constant for all frequencies and will not alter the amplitude of the command signal. Such a compensation approach represents a major step forward, but this assumption only corresponds to a very simple model of the actuator. Darby et al. [9] also proposed a method to estimate the delay online within each time step during RTHT, which would be better for capturing the variation in the delay with respect to the stiffness change of the physical substructure and thus may enable a better compensation scheme. Here the error in the delay estimation is assumed to be proportional to the relative position error by an adaptive gain which is the hyperbolic tangent of the velocity. Zhao et al. [10] also derived a nonlinear mathematical model for a servo-hydraulic actuation system. Experimentally it was demonstrated that both an amplitude reduction and a phase lag were introduced in the testing system. These effects were compensated using a first-order phase lead filter. Using more complex methods, an adaptive minimum control synthesis (MCS) [11] strategy was developed to offer the advantage that the dynamics of experimental substructure are not required when designing the controller. Furthermore, a model based strategy [7] that combines a unity gain low-pass filter as the feed-forward compensator with a feedback proportional gain controller has been developed. This strategy is adopted for the case study presented in Section 4 of this paper.

To solve the equations of motion (EOM) for the structural components in a test, a number of discretization schemes have been adopted and modified for hybrid testing. With nonlinear numerical substructures with a high number of degrees-of-freedom (DOF), the response analysis (numerical integration) takes much longer than signal generation. Nakashima et al. [12] proposed an algorithm to conduct two tasks in parallel but at different rates (running at 10 ms/sample and 1 ms/sample, respectively). This enables refined analysis of complex models while also providing continuous loading to the physical components. Here polynomial extrapolation was first used to generate the command displacement signal, and then interpolation was triggered once the target displacement became available. Darby et al. [13] used the integral form of the EOM in modal coordinates and discretized it by applying a first-order-hold (FOH) equivalence approach. Additional work was presented by Blakeborough et al. [14] who developed an approach to solve the nonlinear numerical problem with less computational effort. This approximation method combines linear mode shape vectors with a reduced basis of additional deformed shape vectors that encapsulate inelastic behavior. An unconditional stable implicit method was chosen as the numerical integration scheme by Shing et al. [15]. Because the number of iterations for each time step to reach convergence is not known in advance, a special iterative method that has a fixed number of iterations was proposed. After the trial displacement was evaluated at the end of iteration, the desired displacement was computed with a quadratic interpolation and imposed on the experimental substructure. To minimize the error and enforce equilibrium at the end of each time step, a correction was introduced for updating displacement and force in the last iteration. Chen and Ricles [16] developed an explicit algorithm (CR) which has the same accuracy as the well established Newmark method. By properly assigning stable poles to the discrete transfer function, this algorithm can be unconditionally stable.

Summary: As existing compensation techniques are suitable for many real-time hybrid testing situations, CIRST aims first to incorporate those solutions in a reusable manner. If needed to improve particular performance levels, however, we will also consider designing and implementing new compensation algorithms as needed.

2.3. System Architectures

System architectures for hybrid testing consist primarily of a *target system* with the following components: hydraulic actuators and inner loop controllers, sensors, analog-to-digital (A/D) and digital-to-analog (D/A) converters for sensor data input and actuator command output respectively, and target computer hardware for system simulation and control. The target system performs real-time computationally intensive tasks such as solving the equations of motion for test structures, analyzing the analytical substructure and generating actuator commands.

In addition to the target system, a separate host computer is often used for system design, and for monitoring and visualization of test results. Host systems do not require real-time capabilities and are generally realized by standard desktop computers. However, since the target system is the center of a RTHT system and its design has the greatest impact on the accuracy of the entire system and thus on the fidelity of the experiments run with it, in this section we will focus primarily on prior research into the hardware and software aspects of target systems.

There are two major kinds of target systems: proprietary systems with DSP processors, or generic ones based on x86 processors. The system from UIUC [7], for example, uses a dSPACE [17] parallel processing DSP board to provide real-time control. Proprietary hardware usually provides a higher degree of software/hardware integration, which can improve system performance and predictability, but may limit flexibility in extending the system.

In software, a controlled system can be specified using high level Simulink models and then compiled into object code for target systems. The object code is then linked with a light weight real-time kernel which provides basic interrupts and I/O services to generate executables. dSPACE's TargetLink [17] is an integrated toolset which includes the Simulink model compiler and a real-time kernel to produce executables in their hardware platforms. The xPC target [18] from Mathworks is a similar toolset that targets generic x86 hardware instead. The major benefit of both solutions is that the tools provide streamlined environments from model definition and evaluation to target deployment that requires little programming for basic systems.

However, for more complicated hybrid testing systems with hundreds of degrees of freedom, more complex nonlinear material and structural models are needed, which are core elements of the OpenSees [19] open source structural analysis framework adopted by NEES [15]. OpenSees also uses object oriented programming to provide tools for reusably specifying numerical models for simulations. OpenSees is purely for computation of the response of the numerical model, and provides no built-in support for real-time operation. NEES runs it on Phar Lap ETS (an real-time OS which provides a subset of Win32 APIs to minimize the effort for porting desktop application to embedded systems) in order to achieve real-time performance.

Summary: Existing solutions provide real-time support, portability of solutions and reusable experiment specifications. However, no single existing platform incorporates all these advantages. CIRST aims to extend the state of the art by incorporating the strengths of existing systems into a single reconfigurable instrument with reusable components.

3. Motivation for CIRST

Any architecture designed to support real-time hybrid testing must support the following capabilities: (1) specifying and importing different experiment designs; (2) mapping execution of the simulation model onto available hardware resources; (3) performing I/O to and from digital and/or analog devices; and (4) meeting strict temporal deadlines for execution and I/O.

A key goal of our CIRST project is to develop configurable and reusable components to tackle recurring problems in the design and deployment of real-time hybrid structural testing. Similar design criteria in other domains have led to the design of reusable system software for managing programming complexity in order to create robust systems without sacrificing efficiency and flexibility. In order to realize such an approach, which is essential to our vision for CIRST, we must first consider the specific challenges facing platforms designed for real-time hybrid structural testing.

Flexible timing mechanisms: Target platforms for real-time hybrid simulation often adopt a time-triggered architecture, where the system runs at some preset frequency. During a single period the system must first read sensor data, and use that data to both calculate and issue commands to the actuator. This induces a requirement that a platform designed to run real-time hybrid test experiments must first have accurate time triggering. This may be accomplished using either hardware or software mechanisms, as long as the desired timing semantics can be achieved. This suggest that providing the ability to plug in different time-triggering mechanisms by providing common upcalls during which specified code is executed, can increase the portability and applicability of CIRST.

In addition CIRST must be able to provide for interactive control of experiments. This ability must not compromise the real-time execution of simulated components or communication amongst virtual and physical components.

Porting and deployment: The simulation run at each time step is the most critical of the real-time operations performed during an experiment. The process of porting this code onto the target platform and deploying it to appropriate computing resources is thus a critical feature of a system architecture for running real-time hybrid tests.

There are several strategies for porting and deployment. The strategy employed by dSPACE, as described previously, is to use a toolset to create executable code from Simulink to be run on the DSP board. This approach has several advantages. It provides dedicated computation via use of parallel processing available on a DSP board. It also allows for experiment code to be specified using high level Simulink models - a familiar programming environment for domain specialists. However, because this approach targets specific hardware the feasible scale of the experiments is limited both by the underlying hardware and by the efficiency of the code generated during this translation step. It does however allow rapid turn-around from experiment design to deployment, and that the approach can be used with minimal complexity by domain experts. Conversely, more general programming languages like C++ might prove more efficient - but at the cost of longer turn around times due to less language level support for needed mathematical operations and greater required end user sophistication.

An ideal balance would be a architecture that allows porting from multiple sources - quick if potentially less efficient deployments created in a environment familiar to the domain expert; or potentially more efficient experiments that in return require longer deployment times and more programming skills from end users, but allow more flexibility in targeting different platforms (for example the ability to spread computation among a cluster of computers). To do this requires an architecture that can both flexibly deploy computations and provide flexibility in how the experiment can be specified, allowing for both high performance and rapid deployment and testing. This architecture will be in sharp contrast to existing systems that tend to be vertically “stove piped” in their design, restricting both flexibility in experiment design and specification as well as deployment options.

Communication, input, and output: Another common requirement for an architecture for running real-time hybrid testing is to allow input from disparate sources, both analog and digital. For each of these types of input different considerations must be observed - for analog signals this includes caution with the range of acceptable inputs and outputs of physical devices, associated gains for translating voltage into physical units and issues related to frequency and sampling rate. However, it is natural to want to use these inputs interchangeably, and providing a common abstraction for CIRST would greatly simplify development and deployment of experiments.

Communication between architecture components also must be specified in order to reduce development complexity and the potential for error. Analog voltage signals from sensors, for instance, have to be converted into physical units for simulation. This translation can be a potential source of error during experiment design, and can itself be a non-trivial operation, if for instance, compensation for sensor dynamics must occur before the sensor reading can be interpreted. Additionally, signal and command voltage ranges may need to be limited in order to protect experiment equipment or for safety concerns. A system that allows verification of these communication channels - in order to assure that data is being sent converted into proper units - is essential to correct deployment of experiments.

Holistic reconfiguration: These different communication and deployment options mean that a system like CIRST must also handle configuration changes in a rigorous manner. For instance, there is a high potential for error as a system is moved from one configuration to another. For instance, different communication paths are necessary when running a simulation only regression test versus a closed-loop real-time hybrid simulation. This configuration changes introduce cross-cutting dependencies. First, in order to replace input from a live sensor with input from a trace file in order to script experiment execution requires several interdependent changes to be made. For instance, gain functions associated with the sensor might need to be bypassed. Secondly, simulations of physical components might need to be included. Instead of sending displacement commands to a physical actuator, commands are instead sent through a actuator simulation and fed back into the system as input.

Another configuration property is ensuring proper shutdown and startup of an experiment. While a single actuator experiment might be very simple to start and stop - the actuator should be moved to some zero displacement at the beginning and end of every test run - this may be a non-trivial operation in experimental setups with multiple actuators or dynamic elements.

Finally, it is desirable that such an architecture be portable, and deployable using a configuration common off the shelf components, greatly reducing the cost of these systems and the need to expensively over-provision research labs for future experiments. In contrast, the specialized hardware and software required for systems like dSPACE can easily run into the tens of thousands of dollars.

4. Case Study

In order to test the feasibility of creating such a platform we have designed and implemented a prototype software layer that integrates a commercially available I/O device and C++ and Matlab simulation specifications on a standard Linux machine. To evaluate our configurable approach to real-time hybrid testing, we designed an experiment with the following specific goals: (1) to verify the ability of our prototype platform to run a hybrid test; and (2) to compare the performance and accuracy to a commercially available platform, specifically dSPACE.

We have designed an experiment with identical numerical integration and control schemes on both platforms. We conducted the experiment on a single story, one bay portal frame structure (as in [7]). The structure can be idealized as a single degree-of-freedom system, assuming the beam is rigid with a concentrated mass. A schematic diagram of this specimen is shown in Figure 2. Mass, damping and the left hand column stiffness are all modeled computationally, leaving only the right hand column to be physically tested.

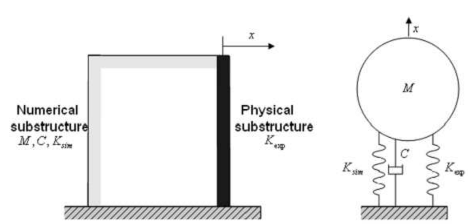

Figure 2. Test Setup: (a) Computational and Physical Substructures, and (b) Mechanical Model

To verify the developed real-time control system, a small scale steel compression spring is used to represent an actual cantilevered column. The linear elastic spring has a nominal stiffness of 37.6 kN/m (215 lb/in) and the maximum allowable deflection is 0.07 m (2.77 in). The computational substructure is assumed to have the same stiffness. Mass is taken to be 1910 kg so that system natural frequency is 1 Hz and a damping ratio of 5% is assumed.

A hydraulic actuator is typically used in hybrid structural testing to apply the necessary displacements to the physical specimen to provide the large forces needed for structural testing. Verification experiments were conducted in the Washington University Structural Control and Earthquake Engineering Laboratory. The lab houses a hydraulic pump that can be operated at 3,000 psi with maximum flow rate of 43 GPM. A Shore-Western 910D double-ended hydraulic actuator is employed as the loading device to drive the test specimen. The actuator has a maximum stroke of 6 inches, with a built-in concentric linear variable differential transformer (LVDT) for ready integration into a position feedback control system. A Schenck-Pegasus 162M servo-valve rated for 15 GPM at 1,000 psi pressure drop is used to control the actuator. The servo-valve has a nominal operational frequency range of 0-60 Hz and is driven by a Schenck-Pegasus 5910 digital controller. An Omega load cell with a range of 1 kip is included in series with the test specimen to measure the restoring force. The experimental setup is shown in Figure 3.

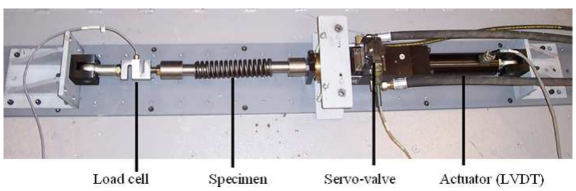

Figure 3. Experimental Setup for Hybrid Structural Testing

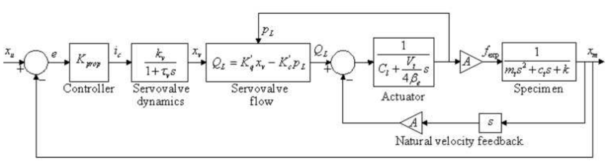

Figure 4. Block Diagram of the Physical Components

Figure 4 illustrates the control relationships in our experimental setup [6], [7]. To provide inner loop control to the hydraulic actuator (for stabilization and tracking) a digital controller is used to measure the displacement of the actuator and compute the difference between the command and the measured feedback signal, and then generates an electrical signal to drive the servo-valve spool. Hydraulic flow due to spool displacement causes a pressure difference inside the actuator chambers resulting in a force on the test specimen.

Due to the dynamics involved in this component of the experiment, there is a noticeable lag in the response to commands which requires compensation. To provide compensation, a model is developed for the test equipment. Known models for each component of the hydraulic actuator are employed [20], [6], [7] and the physical substructure can be simplified as a fourth-order linear dynamic system. Parameters of this model may be identified before experiments are conducted using a nonlinear least-squares optimization routine to fit the model to the frequency domain experiment transfer function shown in equation 1:

$$G_{mu}(s) = \frac{K_p \frac{K_q A}{K_c}}{H(s)} \tag{1}$$

where $H(s) =$

$$(\frac{V_t}{4\beta_e K_c} m_t \tau_v) s^4 + (\frac{V_t}{4\beta_e K_c} m_t + m_t \tau_v + \frac{V_t}{4\beta_e K_c} c_t \tau_v) s^3 +$$

$$(m_t + \frac{V_t}{4\beta_e K_c} c_t + \frac{A^2}{K_c} \tau_v + c_t \tau_v + \frac{V_t}{4\beta_e K_c} k \tau_v) s^2 +$$

$$(c_t + \frac{V_t}{4\beta_e K_c} k + \frac{A^2}{K_c} + k\tau_v) s + k + K_p \frac{K_q A}{K_c}.$$

A combined feed-forward and feedback control scheme is used to compensate for the actuator dynamics [7]. The primary goal of the compensator is to let the experimental substructure measurement (x_m) track the desired displacement (x_d) calculated from the numerical substructure. The transfer function of the experimental substructure described in equation 1 can be expressed in general form in equation 2:

$$G_{mu}(s) = \frac{K}{\prod_{i=1}^{n} (s - p_{mu,i})} \tag{2}$$

where K is the gain and $p_{mu,i}$ are the poles of the system. A unity gain low-pass filter is used as the feed-forward compensator which can approximate the inverse of G_{mu} while also maintaining system stability, as is shown in equation 3:

$$G_{FF}(s) = \alpha^n \frac{\prod_{i=1}^{n} (s - p_{mu,i})}{\prod_{i=1}^{n} (s - \alpha \bullet p_{mu,i})} \tag{3}$$

Here the zeros of the filter are used to cancel the poles of the test structure, and α takes a value greater than one so that the filter has larger poles which will not interfere with the low frequency dynamics of interest. A feedback proportional controller $G_{FB}(s) = K_{FB}$ is adopted to address the test structure modeling inaccuracy. Note that the feedback gain needs to be small enough to maintain system stability.

α needs to be carefully selected for the feed-forward compensator design. As indicated in equation 3, a value of $\alpha = 1$ is equivalent to non-compensation. On the other hand, when α approaches infinity, the compensator will behave as the pure inverse dynamics of the test structure. Ideally it can enable x_m to provide exactly the commanded displacement x_d, but this non-causal system is not acceptable in reality. Too large an α value can potentially fail in two aspects. Since the hydraulic actuator is inherently a nonlinear device, modeling error due to linear approximation can be largely magnified even cause unbounded high frequency

K_p	3	mA/in	controller proportional gain
τ_v	4.52e-3	s	servo-valve time constant
K_q	38.966	$in^3/s/mA$	valve flow gain
$K_c^{'}$	2.53e-6	$in^3/s/psi$	valve flow pressure gain
A	0.858	in^2	piston area
C_l	1e-6	$in^3/s/psi$	piston leakage coefficient
V_t	32.326	in^3	volume of fluid
β_e	95387	psi	effective bulk modulus
m_t	0.06	$lb-s^2/in$	mass of test specimen
c_t	17.45	$lb-s/in$	viscous damping coefficient
k	200.32	lb/in	stiffness of specimen

Table 1. Identified actuator parameters

response. In terms of digital implementation, relatively high compensator natural frequencies are in the same order of magnitude as the Nyquist frequency, and this will cause significant computational error or numerical instability.

Our prototype system is a C++ implementation with software time-triggering using Linux OS time API calls and busy waiting. It runs on a Dual Pentium 4 Xeon 2.40 GHz processor machine with one gigabyte of RAM. For communicating with the analog sensors, we use a National Instruments Data Acquisition Board (BNC-2120 DAB), and interface with it using the vendor provided C API. Experiments can be specified using either Matlab and the Matlab C Compiler (MCC), or natively in C++. At every time trigger event we use a basic version of a generic upcall to execute the appropriate simulation step. Our results reflect a native C++ implementation of the experimental simulation.

For comparative purposes the real-time control system is also implemented using a commercial product. A computer equipped with a dSPACE DS1003 DSP real-time control system is used to perform numerical integration as well as actuator dynamics compensation. The algorithms are implemented in MathWorks SIMULINK and then downloaded to the dSPACE processor using Real-time Workshop. A DS2003 A/D board with 32 channels and a DS 2102 D/A board with 6 channels are used to convert between digital and analog signals at a resolution of 16 bits.

Results: An experimental transfer function for the overall physical component was obtained under a band-limited white noise excitation signal with a bandwidth of 50 Hz and root mean square magnitude of 0.0028 inch. Actuator parameters were identified by fitting a curve to the experimental data and the values are shown in Table 1. As can be seen in Figure 5, the curve-fitted model represents the actuator dynamics well so it is used for the controller design to compensate the actuator dynamics. Figure 6 compares the floor displacements under a 2 Hz sinusoidal acceleration input with 7.7 m/s^2 amplitude. To prevent the maximum displacement from exceeding the physical actuator stroke, the input was scaled down to 1/40 of the original magnitude. The results for the developed system match those for the dSPACE

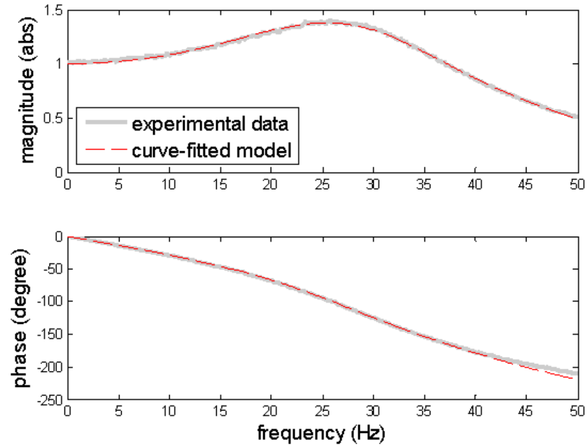

Figure 5. Experimental transfer function vs. model

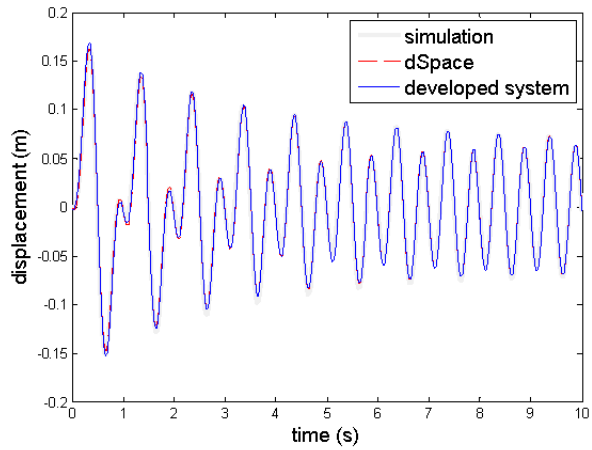

Figure 6. Comparison of floor displacement

system, and both sets of results follow the simulation results closely as well.

5. Conclusion

We identify real-time hybrid structural testing as a grand challenge for cyber-physical systems. Actuator dynamics, complex interactions among cyber and physical components, and computation and communication delays all hamper the ability to conduct high-fidelity tests of structures in real-time. We present the motivation and vision of Cyber-physical Instrument for Real-time hybrid Structural Testing (CIRST), which aims to provide a highly configurable and reusable middleware framework for real-time hybrid testing. Preliminary results on a simple hybrid test demonstrate the feasibility of our approach. We believe CIRST will have broad impacts on the fields of both civil engineering and real-time systems. It will enable high-fidelity real-time testing of a wide range of civil infrastructures, and provide a high-impact cyber-physical application for the study and evaluation of real-time middleware.

Acknowledgment: The research presented in this paper is supported in part by NSF grants CNS-0821713 (MRI), CNS-0448554 (CAREER), and CCF-0448562 (CAREER).

References

[1] M. Ahmadizadeh, G. Mosqueda, and A. Reinhorn, "Compensation of actuator delay and dynamics for real-time hybrid structural simulation," in *Earthquake Engineering and Structural Dynamics*, vol. 37, pp. 21–42, 2008.

[2] J. Carrion and S. B.F., "A model based delay compensation approach for real time hybrid testing," in 4^{th} *International Conference on Earthquake Engineering*, (Taipei,Taiwan), 2006.

[3] S. A. Mahin and P. B. Shing, "Pseudodynamic method for seismic testing," in *Journal of Structural Engineering*, vol. 111, pp. 1482–1503, 1985.

[4] S. Mahin, P. Shing, C. Thewalt, and R. Hanson, "Pseudodynamic test method. Current status and future directions," in *Journal of Structural Engineering*, vol. 115, pp. 2113–2128, ASCE, 1989.

[5] K. Takanashi and M. Nakashima, "Japanese activities on on-line testing," in *Journal of Engineering Mechanics*, vol. 113, pp. 1014–1032, ASCE, 1987.

[6] S. J. Dyke, B. F. Spencer, P. Quast, and M. K. Sain, "Role of control-structure interaction in protective system design," in *Journal of Engineering Mechanics*, vol. 121, pp. 322–338, ACSE, 1995.

[7] J. Carrion and B. F. Spencer, "Model-based strategies for real-time hybrid testing," Tech. Rep. NSEL-006, University of Illinois at Urbana-Champaign, 2007.

[8] T. Horiuchi, M. Inoue, T. Konno, and N. Y., "Real-time hybrid experimental system with actuator delay compensation and its application to a piping system with energy absorber," in *Earthquake Engineering and Structural Dynamics*, vol. 28, pp. 1121–1141, 1999.

[9] A. P. Darby, M. S. Williams, and A. Blakeborough, "Stability and delay compensation for real-time substructure testing," in *Journal of Engineering Mechanics*, vol. 128, pp. 1276–1284, ACSE, 2002.

[10] J. Zhao, C. French, C. Shield, and T. Posbergh, "Considerations for the development of real-time dynamic testing using servo-hydraulic actuation," in *Earthquake Engineering and Structural Dynamics*, vol. 32, pp. 1773–1794, 2003.

[11] S. A. Neild, D. P. Stoten, D. Drury, and D. J. Wagg, "Control issues relating to real-time substructuring experiments using a shake table," in *Earthquake Engineering and Structural Dynamics*, vol. 34, pp. 1171–1192, 2005.

[12] M. Nakashima and N. Masaoka, "Real time on-line test for mdof systems," in *Earthquake Engineering and Structural Dynamics*, vol. 28, pp. 393–420, 1999.

[13] A. P. Darby, A. Blakeborough, and M. S. Williams, "Improved control algorithm for real-time substructure testing," in *Earthquake Engineering and Structural Dynamics*, vol. 30, pp. 431–448, 2001.

[14] A. Blakeborough, M. S. Williams, A. P. Darby, and D. M. Williams, "The development of real-time substructure testing," in *Philosophical Transactions: Mathematical, Physical and Engineering Sciences: Dynamic Testing of Structures*, vol. 359, pp. 1869–1891, The Royal Society, 2001.

[15] P. B. Shing, Z. Wei, R. Y. Jung, and E. Stauffer, "NEES fast hybrid test system at the University of Colorado," in 13^{th} *World Conference on Earthquake Engineering*, (Vancouver, Canada), 2004.

[16] C. Chen and J. M. Ricles, "Development of direct integration algorithms for structural dynamics using discrete control theory," in *Journal of Engineering Mechanics*, vol. 134, pp. 676–683, 2008.

[17] "http://www.dspace.de/ww/en/inc/home.cfm."

[18] "http://www.mathworks.com/products/xpctarget/."

[19] "http://opensees.berkeley.edu/."

[20] C. W. DeSilva, *Control Sensors and Actuators*. Prentice Hall, Inc., 1989.