The 2015 World Congress on
Advances in Structural Engineering and Mechanics (ASEM15)
Incheon, Korea, August 25-29, 2015

Actuator controller interface program for pseudo-dynamic hybrid simulation

Hu Zhan1) and *Oh-Sung Kwon2)

1), 2) Department of Civil Engineering, University of Toronto, Canada
2) os.kwon@utoronto.ca

ABSTRACT

One of the essential tasks in hybrid (analysis-experiment) simulation is to establish communication between an actuator controller and a numerical integration module. In addition, if multiple actuators are used to control coupled degrees of freedoms (DOF) of a specimen (such as a column under axial and lateral forces and moment), the displacement commands in the numerical model's Cartesian coordinate system need to be transformed to actuators' strokes and feedback displacements, and forces need to be converted back to the model's Cartesian coordinate system. This conversion requires iteration due to the geometric nonlinearity of the testing setup. Establishing the communication and enabling the coordinate transformation for hybrid simulation are not trivial tasks for a testing facility that is new to the simulation method. To facilitate the adoption of a hybrid simulation method in a conventional structural testing facility, a generalized controller interface program is developed based on LabView and National Instrument's hardware. The interface program, called the Network Interface for Controllers (NICON), receives commands from the network based on a standardized data exchange format, converts coordinate systems, generates analog voltage commands to actuator controllers, and returns measured responses. The program is generalized so that it can be used for various configurations of testing setups such as single DOF, three coupled DOFs, six coupled DOFs, and ten uncoupled DOFs. Validation tests were carried out using multi-axial testing apparatus at the University of Toronto. Considering the generality of the program and the low initial cost of the hardware, the development is expected to greatly facilitate the application of pseudo-dynamic hybrid simulation in research projects.

1. INTRODUCTION

Hybrid simulation allows for the integration of physical specimens with a numerical structural model. The simulation method can accurately predict the structural performance when there is a lack of understanding on the hysteretic behaviour of physically represented components, and when the hysteretic behaviour of the

---

1) Former graduate student
2) Associate professor

components influences the response of the structural system. When the behavior of the components is well understood and a numerical model of the component is available, then a hybrid simulation with a physical specimen is not warranted because one can analyze the entire structural system numerically. Similarly, if the physical components' hysteretic behaviour does not influence the global structural response (i.e., there is no significant interaction between the tested component and the structural system), hybrid simulation is not needed. When there are many similar elements, which collectively contribute to the lateral load resisting system, testing only one specimen does not necessarily increase the accuracy of the hybrid simulation. In such a case, one can use the model updating method (Kwon and Kammula 2013) to maximize the use of experimental data, or needs to test multiple specimens using a specialized experimental apparatus as shown in Mojiri, et al. (2015).

Theoretical bases for hybrid simulation have been extensively investigated in the past decades. A group of research papers focuses on the development of hybrid simulation frameworks which can integrate numerical models with physical specimens (Karavasilis et al. 2008; Kwon et al. 2008; Saouma et al. 2012; Schellenberg et al. 2009). Other groups of study focus on actuator delay compensation in real-time hybrid simulations (Carrion and Spencer, Jr. 2007; Darby et al. 2002; Horiuchi et al. 1999; Mercan and Ricles 2009; Phillips and Spencer 2012), and the development of robust time integration schemes (Chang et al. 2011; Chen and Ricles 2008; Hung and El-Tawil 2009). Shao and Griffith (2013) also summarized recent applications of hybrid simulations.

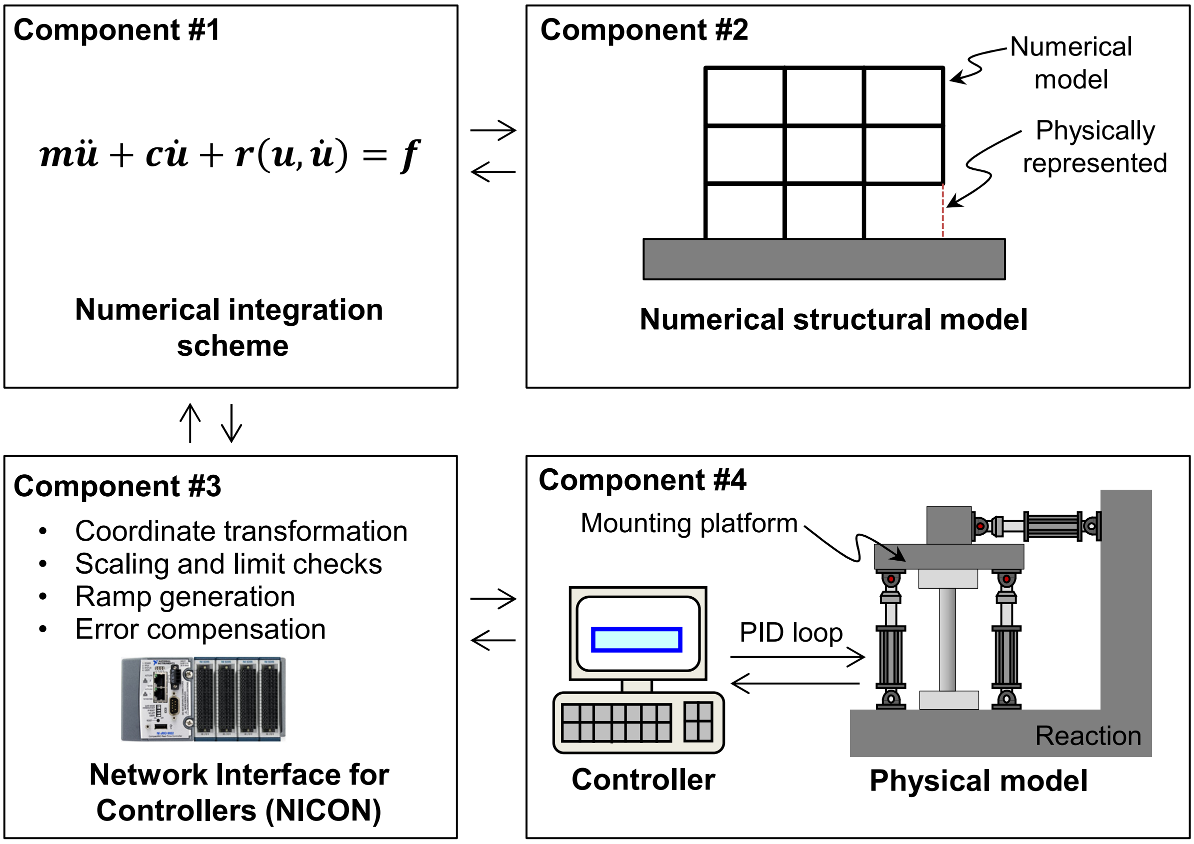
Even though there are a large number of publications which push the boundaries of state-of-the-art hybrid simulation methods. Implementation of hybrid simulation in a structural testing facility, which is new to the simulation method, is not a trivial task due to several practical reasons. For example, a numerical model and time integration scheme should be able to send predicted displacements, actuator controllers should be able to impose the displacements to a physical specimen, and measured forces and displacements need to be returned back to the main time integration scheme. In addition to the communication related issue, the displacement commands in the Cartesian coordinate system should be properly transformed to actuators' strokes. For a specimen with a single degree of freedom (DOF), this transformation is trivial. If multiple degrees of freedoms are controlled with multiple actuators, the coordinate transformation is not a trivial task. Furthermore, in the pseudo-dynamic hybrid simulation, the accurate control of the specimens' deformation is always a challenging issue due to the elastic deformation of the supporting systems as well as backlash or slippage of the connections.

The objective of this paper is to provide a solution for the above-mentioned practical issues such that the pseudo-dynamic hybrid simulation method can be easily adopted in any structural testing facility. The objective is achieved by developing a generalized network interface program for actuator controllers (NICON). The program can be used for various configuration of actuator setups and provide functionality of communication, coordinate transformation, error compensation, ramp generation, and limit checks. In this paper, the functionalities required for a PSD hybrid simulation, which are implemented in NICON, are briefly summarized. The results from validation tests with a multi-DOF testing apparatus are presented.

2. REQUIRED FUNCTIONALITIES FOR PSD SIMULATION

2.1. Components in PSD Hybrid Simulation

Essential components that are required to carry out a pseudo-dynamic hybrid simulation are #1) a non-iterative numerical time integration scheme, #2) a numerical model of a structural system with substructuring capability, #3) a middleware between the integration scheme and the actuator controller, and #4) an actuator controller which can impose displacement commands to a physical specimen. The overview of the components of pseudo-dynamic hybrid simulations is illustrated in Fig. 1.



Notes:

- Components #1 and #2 are typically integrated with software such as OpenSess or Abaqus. In UI-SimCor, these two components are separated and communicate through a TCP/IP network.
- Components #1 and #3 generally communicate through a TCP/IP network (UI-SimCor and OpenFresco). In some frameworks targeted for real-time simulation, the components #1 through #3 are integrated in a single program.
- Components #3 and #4 typically communicate through analog voltage signals or network shared memory.

Fig. 1 Components in hybrid simulation

Depending on research groups, the above components are implemented in various ways. For example, in the hybrid simulation framework, UI-SimCor (Kwon et al. 2008), the numerical time integration scheme (component #1) is independent from the structural analysis module (component #2) or physical testing module (component #4). UI-SimCor only runs a numerical time integration scheme, and all restoring forces are obtained from structural analysis modules or a physical testing module. UI-SimCor requires middleware (component #3), which can receive commands through a network and transfer it to an actuator controller (component #4). In the OpenSees-OpenFresco framework (Schellenberg et al. 2009), OpenSees is used as a main module with a numerical time integration scheme and numerical structural elements (components #1 and #2), and OpenFresco is used as an interface between OpenSees and an actuator controller (component #3). In other frameworks, such as Mercury (Saouma et al. 2012) or HybridFEM (Karavasilis et al. 2008), computationally efficient nonlinear elements are implemented in a real-time platform such that hard real-time hybrid simulation with nonlinear elements can be carried out. The focus on this paper is on the generalized middleware for component #3, which is termed Network Interface for Controllers (NICON). The following sections present an overview of the communication mechanism implemented in NICON and various functionalities.

2.2. Communication

The method that is used for communication between the main hybrid simulation framework and an actuator controller varies. One way to establish the communication is using a TCP/IP network in which a main integration scheme sends data packets to an interface program (e.g. OpenFresco or NICON). Then, the interface program, which is also referred to as middleware, communicates with an actuator controller either through network shared memory (e.g. SCRAMNet) or voltage input/output. The shared network memory approach requires a proprietary network card and an actuator controller, which can accept the commands through the network shared memory. Because data is transferred through the network in a digital format, the accuracy of the commands and measurements does not decrease due to digital-to-analog (D-A) and analog-to-digital (A-D) conversion processes. The second approach using an analog voltage signal requires relatively low-cost A-D or D-A modules, and most commercial actuator controllers (e.g. controllers from Instron, MTS, and Shore Western) can take the analog voltage signal as a command source. Thus, the second approach can be readily implemented in most structural testing facilities. The disadvantage of the second approach is potential loss of accuracy in the A-D or D-A conversion process and noise coming from wiring. Using high-resolution converters, shielded cables, and/or band-pass filters, however, can minimize the loss of accuracy.

The communication mechanism in NICON is based on the second approach; all communication with the main integration module (component #1 in Fig.1) is carried out through TCP/IP network and all communication with an actuator controller is carried out through analog voltage signals as illustrated in Fig. 2. The network communication is based on the standardized data exchange format presented in Mojiri et al. (2015). Hardware equipment from the National Instruments (NI) is adopted in this development. The main processing unit is the CompactRIO (cRIO) from NI, which has on-board real-time processor and network cards. The chassis of the cRIO can accept voltage input/output modules. NICON is compiled and loaded on cRIO, which runs autonomously when the unit is turned on. A user can access the control of NICON through a computer that is connected to NICON. The computer, however, is just an interface between a user and NICON rather than a main computing unit. While NICON is written with LabView 2009 and compiled for cRIO in the current version, it is possible to customize the program for different types of hardware equipment from NI, such as PXI- or USB-based systems.

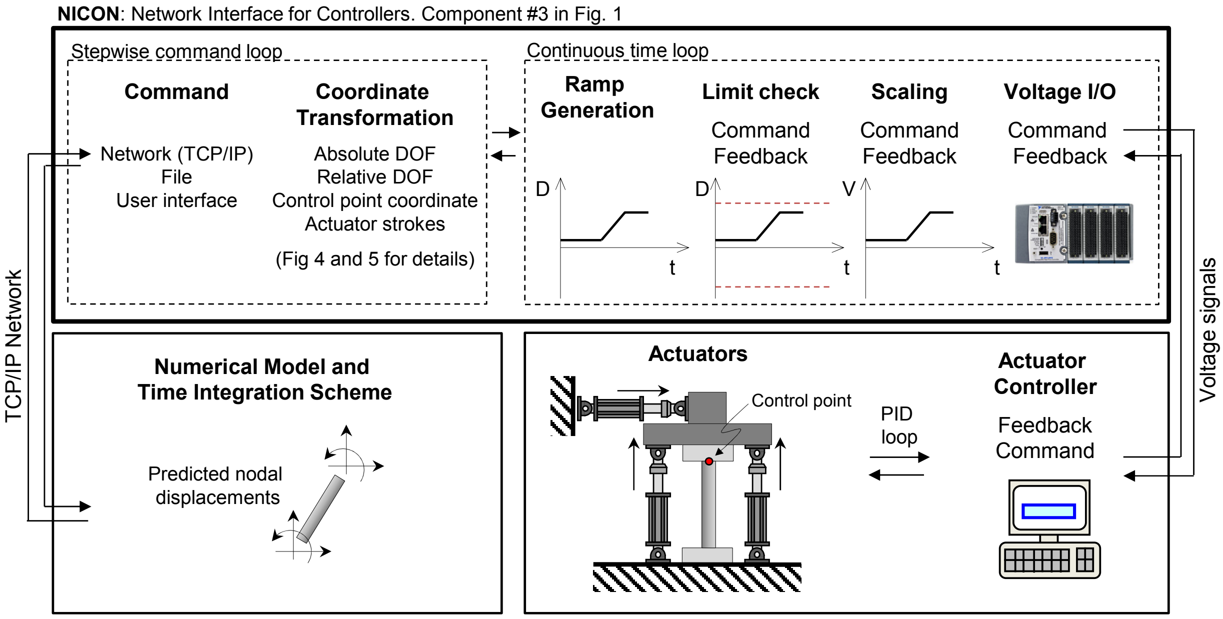


Fig. 2 Overview of NICON and cRIO (component #3 in Fig.1)

2.3. Conversion of coordinate systems

One of the major functionalities of the interface program, NICON, is a coordinate transformation. In hybrid simulations, a control point of a physically tested component is attached to a specimen-mounting block (i.e., a rigid transfer element), which is attached to actuator(s). Each specimen has one control point, which can have up to six DOFs. Depending on the number of DOFs of the control point, the configuration of the specimen, a mounting block, and actuators can vary as illustrated in Fig. 3. It is also possible to use multiple control points in a specimen, such as a beam-column connection which was tested by Mahmoud et al. (2013).

Fig. 3 presents a comprehensive but not exhaustive list of test configurations of structural elements with a single control point. In the configuration in Fig. 3 (a), for example, the axial deformation of a specimen is controlled with a single actuator, which does not require coordinate transformation. In configuration (b), the horizontal displacement of a specimen is controlled with an actuator attached in the same direction. In this configuration, the stroke of the actuator is not identical to the horizontal displacement of the control point because of the geometric nonlinearity of the specimen. When multiple DOFs are controlled, multiple actuators are attached to a mounting block as shown in Fig. 3 (c) to (g). To control n number of DOFs, then at least the same number of actuators is required. For example, two DOFs are controlled using two actuators in configurations (c) and (d). The number of actuators can be larger than the number of the DOFs, which generally introduce complexity in the control because the actuator-mounting block system becomes statically indeterminate.

When multiple DOFs are used as shown in test configurations Fig. 3 (c) through Fig. 3 (g), it is necessary to convert the displacement commands from the Cartesian coordinate system to the individual actuator's stroke. If it is assumed that a frame element is used for hybrid simulation, the numerical element in structural analysis software has total twelve DOFs as shown in Fig. 4 (a), which includes six DOFs of rigid body mode and six DOFs of deformation mode. The twelve DOFs are defined in the structural coordinate system. If the element is represented with a physical model, the six DOF deformation modes need to be abstracted from the twelve DOFs, and be imposed to the physical model as shown in Fig. 4. This process requires conversion of the displacement commands from global to local coordinate systems (x-y-z in Fig. 4.a to x'-y'-z' in Fig. 4.b) and the abstraction of deformation mode (Fig. 4.b to Fig. 4.c). In addition, the testing apparatus' Cartesian coordinate system (x''-y''-z'' in Fig. 4.d) may not be aligned with the specimen's local Cartesian coordinate system (x'-y'-z' in Fig. 4.c). Thus, it needs coordinate transformation from Fig. 4.c to Fig.4.d.

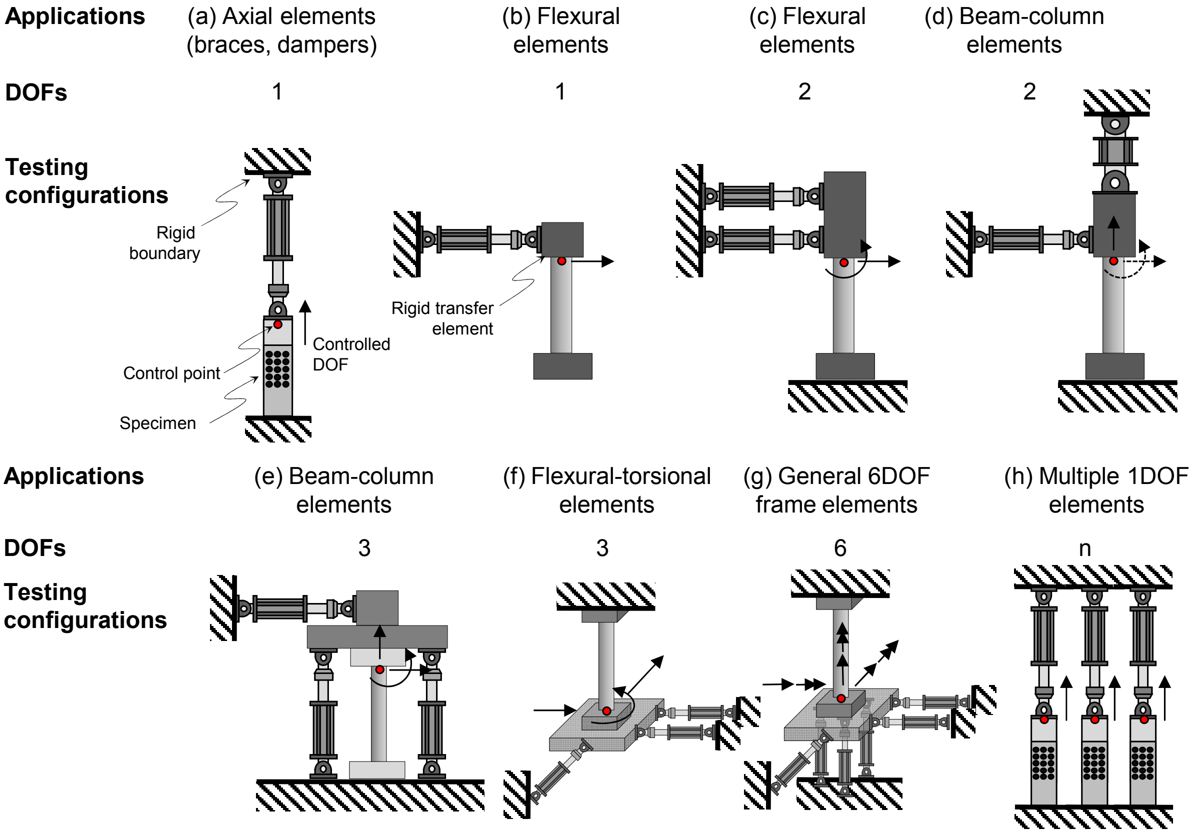
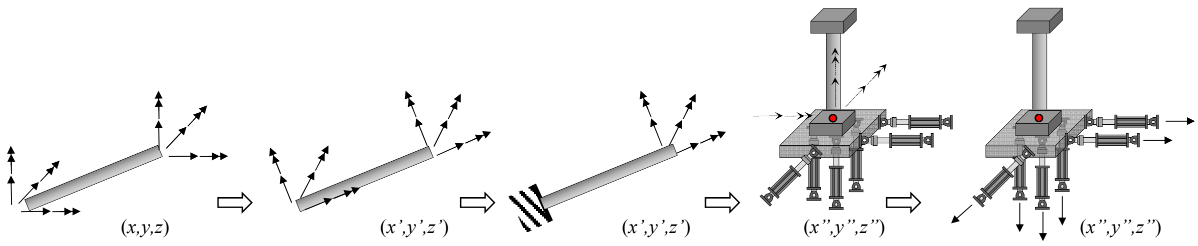


Fig. 3 Typical test configurations for structural components with a single control point



(a) Total displacements in the global coordinate
(b) Total displacements in the element coordinate
(c) Relative displacements in the element coordinate
(d) Displacements in the control point's coordinate
(e) Actuator strokes

Fig. 4 Conversion of coordinate systems

The transformation steps from Fig. 4.a. to Fig. 4.d are all linear (i.e., transformation matrices can be multiplied in each step). Once the relative deformation of the element is defined in the testing apparatus's Cartesian coordinate system (Fig. 4.d), then the commands need to be converted to each actuator's stroke. As an example, Fig. 5 illustrates a three DOF testing apparatus in which two translational DOFs and one rotational DOF are controlled with three actuators, which is similar to the configurations in Fig. 3(e) and (f). The specimen-mounting block (i.e., rigid transfer element) is assumed to be infinitely rigid. A specimen is mounted at an arbitrary control point in the mounting block. Once the control point moves from the original configuration, $\boldsymbol{v}_0$, to a deformed configuration, $\boldsymbol{v}$, then each actuator's stroke changes as below.

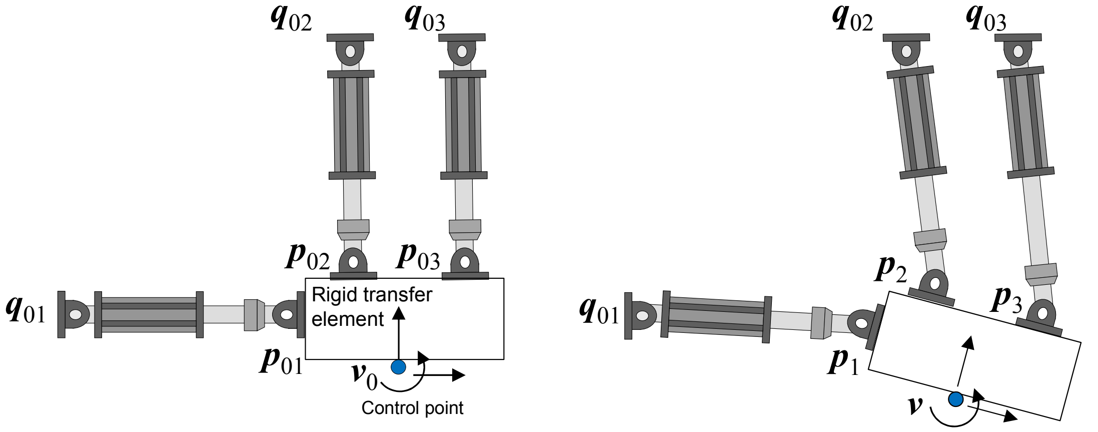
Deformation of the control point, $\boldsymbol{u} = \boldsymbol{v} - \boldsymbol{v}_0$ (1)

Initial length of actuator i, $l_{oi} = |\boldsymbol{p}_{0i} - \boldsymbol{q}_{0i}|$ (2)

Final length of actuator i, $l_i = |\boldsymbol{p}_i - \boldsymbol{q}_{0i}|$ (3)

Stroke of actuator i, $\Delta l_i = l_i - l_{io}$ (4)

where the symbols, $\boldsymbol{v}$, $\boldsymbol{p}$, and $\boldsymbol{q}$ are position vectors of each point with any reference point in the space. Because the specimen-mounting block is assumed to be rigid, defining each actuator's stroke using the above process does not require iteration, even though it is a non-linear process. Detailed steps for the conversion process are available in Nakata et al. (2010). The difficulty is in the backward conversion process. Because the actuators' PID control loop does not perfectly impose the command stroke, the feedback from a linear displacement transducer to an actuator does not always match the commanded displacement to the actuator. Thus, to evaluate the actual displacement of the control point, $\boldsymbol{u}$ in Eq. (1), it is necessary to back calculate the position vector $\boldsymbol{v}$ from the measured stroke, Δl_m, from each actuator. The backward conversion is an iterative process and the algorithm for the process is presented in Nakata et al. (2007). If the load cells attached to the actuators are used to measure forces, the actuators' forces can be converted to the force vector in the global Cartesian coordinate system using force equilibrium.



(a) Original configuration (b) Deformed configuration

Fig. 5 Conversion from Cartesian to the actuator coordinate system

In NICON, the forward and backward coordinate conversion process in Nakata et al. (2007) is implemented for general experimental configurations presented in Fig. 3. The user can specify the number of controlled DOFs, the coordinates of the actuator pins ($\boldsymbol{p}_0$ and $\boldsymbol{q}_0$), the coordinate of the control point ($\boldsymbol{v}_0$), and other information required for the calibration of linear displacement transducers and force transducers. As an example, Table 1 summarizes the input data related to coordinate transformation for the configuration (f) in Fig. 3, which was used for validation tests presented in Section 3.

Table 1. Example input data for a three-DOF testing configuration

Variable	**Value**	**Variable**	**Value**
Number of DOFs	3	Number of actuators	3
Controlled DOFs	Dx, Dy, Rz	Actuator setup	2D
Base points, $\boldsymbol{q}_0$	$\boldsymbol{q}_{01}$ = (1550.9, 76.2) $\boldsymbol{q}_{02}$ = (1550.9, 685.8) $\boldsymbol{q}_{03}$ = (381, 1550.9)	Unit vectors for the global Cartesian coordinate system (x, y, z)	$\boldsymbol{u}_x$ = (1,0,0) $\boldsymbol{u}_y$ = (0,1,0) $\boldsymbol{u}_z$ = (0,0,1)
Attachment points ($\boldsymbol{p}_0$)	$\boldsymbol{p}_{01}$ = (762,76.2) $\boldsymbol{p}_{02}$ = (762,685.8) $\boldsymbol{p}_{03}$ = (381,762)	Unit vectors for the element's local coordinate system (x', y')	$\boldsymbol{u}_{x'}$ = (0,0,1) $\boldsymbol{u}_{y'}$ = (-1,0,0)
Control point ($\boldsymbol{v}_0$)	$\boldsymbol{v}_0$ = (381,381)	Unit vectors for the apparatus' Cartesian coordinate system (x'', y'')	$\boldsymbol{u}_{x''}$ = (0,1,0) $\boldsymbol{u}_{y''}$ = (0,0,1)

2.4. Other features

Controlling the actuators through external signal requires several other functionalities to accurately impose the target displacements, to protect the specimen and equipment, and to debug the system in case of unexpected termination of the simulation. Here are a few other features that were implemented or are currently being implemented in NICON.

Error compensation: In pseudo dynamic hybrid simulations, the target displacement should be accurately imposed on the testing specimen. If there is a tracking error at each time step of hybrid simulation, the error accumulates and propagates throughout the simulation. It is impossible to perfectly or accurately impose the target displacements, but the potential source of systematic errors needs to be considered. From the past hybrid simulations by the authors (Elnashai et al. 2008; Kammula et al. 2014; Mahmoud et al. 2013; Sextos et al. 2014), the following were found to be the sources of the error between the target displacements and specimens' actual deformation:

- Elastic deformation of the testing system: Reaction walls or slabs, or specimen-mounting blocks develop elastic deformation especially when a stiff specimen, such as a concrete column under an axial load, is used in hybrid simulation.
- Backlash and slip in the connection: When friction-type connection is used to connect a specimen to a mounting block, the connection may develop slippage until fasteners develop bearing against holes. When the pins are used to connect an actuator to a specimen, the pin connection may develop backlash depending on the tolerance of the hole sizes.
- Slip in the friction-type specimen: For a specimen which develops resistance based on frictional force, imposing accurate displacement is challenging because of the stick-slip nature of the frictional contact surfaces. If the specimen is very rigid except the slipping surface, the control is not difficult. However, if the specimen has some flexibility which can store elastic energy until the surface starts to slip, then the actual deformation of the slip surface tends to overshoot the target displacement.
- Actuator tracking error: Depending on how the PID gain values are calibrated, the actuators may not execute the displacement commands accurately.

Unless the experimental apparatus including the reaction system has much larger stiffness than the specimen, the error coming from the elastic deformation is unavoidable. The best practice to overcome the error is to directly measure the deformation of the specimen rather than deducing the specimen's deformation using the actuator's displacement transducers. Once the actual deformation of the specimen is measured, the command from NICON to an actuator controller can be updated to compensate for the error. This procedure is analogous to the integrator control where the servo error is added to servo command in each iteration. As a rule of thumb, the displacement of a specimen could be accurately controlled up to 0.05 mm (or 0.002 inches), which was sufficiently accurate for the purpose of pseudo-dynamic hybrid simulation.

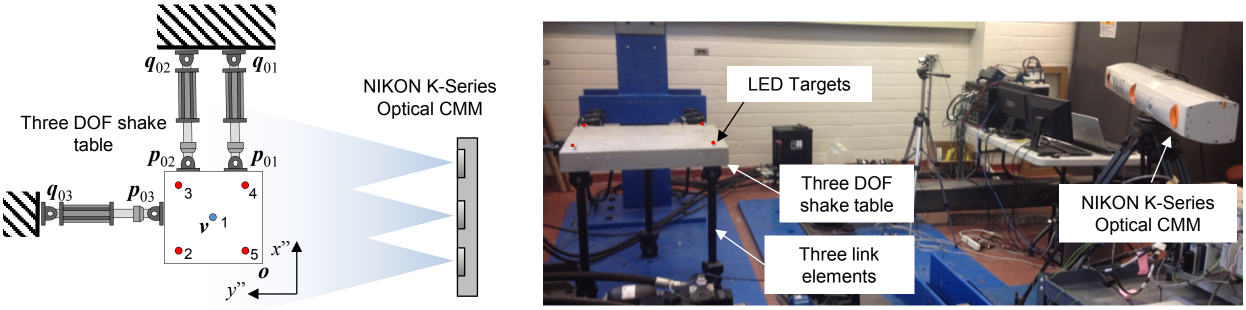
Limit checks: The user can limit the displacement commands sent to the actuator controller. In all actuator controllers, there exists a similar feature to protect equipment and specimens by preemptively imposing displacement (or force) limits. The limit check implemented in NICON provides an extra layer of safety in addition to the limits set in an actuator controller.

<u>Data logging:</u> NICON continuously logs data at pre-defined logging intervals. The logged data includes actual voltage outputs, voltage inputs, and network communication with the main hybrid simulation platform.

3. VALIDATION OF COORDINATETRANSFORMATION

The coordinate transformation of NICON is validated against experimental measurements using a three-DOF testing system as shown in Fig. 6. The system consists of three small-scale actuators with 15 kN of force capacity and ±38 mm stroke. The coordinates of the actuator pins and control points were based on the bottom-right corner of the specimen-mounting block. The coordinates are summarized in Table 1.

The objective of the validation test was to confirm that the target displacement commands in the Cartesian coordinate system at the control point could be accurately achieved using NICON. The target displacements were defined in Matlab and sent to NICON through a TCP/IP network. NICON transformed the commands to the actuators' strokes based on the coordinate conversion procedure in Section 2.3, and generated a ramp for the actuator controller. The actual movement of the specimen-mounting block was measured using a three-dimensional coordinate scanning system, a K-Series Optical CMM from NIKON, as shown in Fig. 6. Each actuator was calibrated before the validation tests.

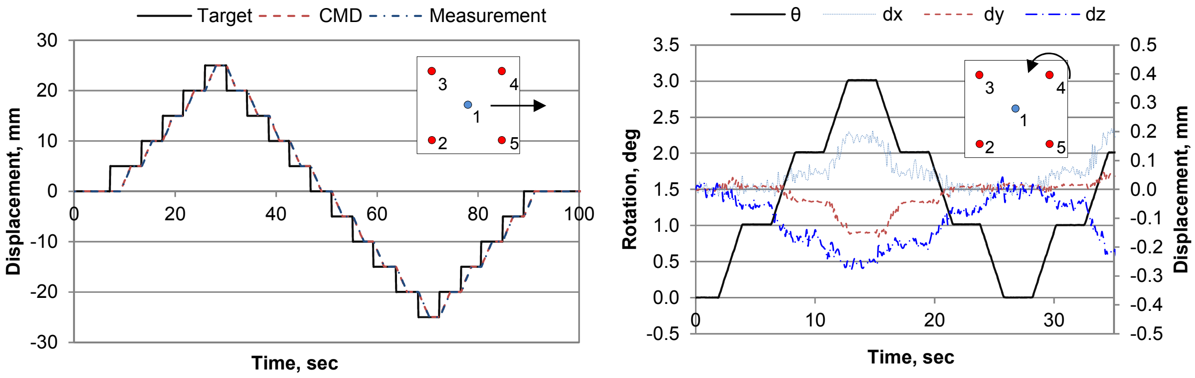


(a) Validation test configuration

(b) Picture of the testing setup

Fig. 6 Validation of the coordinate conversion process

The test results are compared in Fig. 7. In Fig. 7 (a), target displacement is compared against the displacement command generated with NICON, and the displacement measured with the displacement transducers in the actuators. The command was to move the control point in x'' direction. The figure clearly shows that all three values agree well with each other. Fig. 7 (b) compares the experimental results when rotation is imposed to control point #4. Three rotation values (1, 2, and 3 degrees) were commanded. The movement of control point #4 was tracked using the 3D coordinate measurement system. The results in Fig. 7 (b) show that the measured rotation is very close to the target rotation. The control point supposed not to move in x'' or y'' direction, but based on the 3D measurement system, it shows up to 0.3 mm of translational movement. The authors speculate that the error came from inaccurate positioning of the target LEDs that were used to track the coordinate using the 3D measurement system.



(a) Translational DOF: Comparison of target displacement, command to controller, and measurement

(b) Rotational DOF: Error in translational DOF when rotation is only imposed at control point #4

Fig. 7 Validation test results

4. CONCLUSION

Pseudo-dynamic hybrid simulation has been extensively investigated in the past decades. Yet, employment of the testing method in a structural testing facility without prior experience is still very challenging. To facilitate the adoption of the testing method, a network interface program for actuator controllers, NICON, is being developed. The program can receive displacement commands from a hybrid simulation framework, convert coordinate systems, and send voltage commands to an actuator controller. The program has been extensively verified. Validation tests confirmed that the program can accurately impose target displacements to the multi-axial testing apparatus. The program is being finalized and will be available for download. The authors expect that the program will greatly facilitate the adoption of hybrid simulation in many structural testing facilities.

ACKNOWLEDGEMENT

This research project was partially funded by the Natural Sciences and Engineering Council of Canada under the Discovery Grant Program.

REFERENCES

Carrion, J. E., and Spencer, Jr., B. F. (2007). *Model-based Strategies for Real-time Hybrid Testing*. Urbana.

Chang, S.-Y., Yang, Y.-S., and Hsu, C.-W. (2011). “A family of explicit algorithms for general pseudodynamic testing.” *Earthquake Engineering and Engineering Vibration*, 10(1), 51–64.

Chen, C., and Ricles, J. M. (2008). “Development of Direct Integration Algorithms for Structural Dynamics Using Discrete Control Theory.” *Journal of Engineering Mechanics*, 134(8), 676.

Darby, a. P., Williams, M. S., and Blakeborough, A. (2002). “Stability and Delay Compensation for Real-Time Substructure Testing.” *Journal of Engineering Mechanics*, 128(12), 1276.

Elnashai, A. S., Spencer, B. F., Kim, S. J., Holub, C. J., and Kwon, O. S. (2008). “Hybrid distributed simulation of a bridge-foundation-soil interacting system.” *The 4th International Conference on Bridge Maintenance, Safety, and Management*, Seoul, Korea.

Horiuchi, T., Inoue, M., Konno, T., and Namita, Y. (1999). “Real-time hybrid experimental system with actuator delay compensation and its application to a piping system with energy absorber.” *Earthquake Engineering & Structural Dynamics*, 28(10), 1121–1141.

Hung, C.-C., and El-Tawil, S. (2009). “Full operator algorithm for hybrid simulation.” *Earthquake Engineering & Structural Dynamics*, 38(13), 1545–1561.

Kammula, V., Erochko, J., Kwon, O. S., and Christopoulos, C. (2014). “Application of hybrid-simulation to fragility assessment of the telescoping self-centering energy dissipative bracing system.” *Earthquake Engineering and Structural Dynamics*.

Karavasilis, T. L., Seo, C.-Y., and Ricles, J. (2008). *HybridFEM: A Program for Dynamic Time History Analysis of 2D Inelastic Framed Structures and Real-Time Hybrid Simulation*. Bethlehem, PA.

Kwon, O., Elnashai, A. S., and Spencer, B. F. (2008). “A framework for distributed analytical and hybrid simulations.” *Struct. Engr. and Mechanics*, 30(3), 331–350.

Kwon, O., and Kammula, V. (2013). “Model updating method for substructure pseudo-dynamic hybrid simulation.” *Earthq. Engr. & Str. Dynamics*, 42(13), 1971–1984.

Mahmoud, H. N., Elnashai, A. S., Spencer, B. F., Kwon, O., and Bennier, D. J. (2013). “Hybrid Simulation for Earthquake Response of Semirigid Partial-Strength Steel Frames.” *Journal of Structural Engineering*, 139(7), 1134–1148.

Mercan, O., and Ricles, J. M. (2009). “Experimental Studies on Real-Time Testing of Structures with Elastomeric Dampers.” *J. of Structural Engineering*, 135(9), 1124.

Mojiri, S., Huang, X., Kwon, O.-S., and Christopoulos, C. (2015). “Design and Development of Ten-Element Hybrid Simulator and Generalized Substructure Element for Coupled Problems.” *VI International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Venice, Italy, 12.

Nakata, N., Spencer, B. F., and Elnashai, A. S. (2010). “Sensitivity-Based External Calibration of Multiaxial Loading System.” *J. of Engr. Mechanics*, 136(2), 189–198.

Nakata, N., Spencer, Jr., B. F., and Elnashai, A. S. (2007). *Multi-dimensional Mixed-mode Hybrid Simulation Control and Applications*. Urbana, IL.

Phillips, B., and Spencer, B. (2012). “Model-Based Feedforward-Feedback Actuator Control for Real-Time Hybrid Simulation.” *Journal of Structural Engineering*.

Saouma, V., Kang, D.-H., and Haussmann, G. (2012). “A computational finite-element program for hybrid simulation.” *Earthq. Engr. & Struct. Dynamics*, 41(3), 375–389.

Schellenberg, A. H., Mahin, S. A., and Fenves, G. L. (2009). *Advanced Implementation of Hybrid Simulation*. 286.

Sextos, A. G., Bousias, S., Taskari, O., Evangeliou, N., Kwon, O., Elnashai, A., Di Sarno, L., and Palios, X. (2014). “An intercontinental hybrid simulation experiemnt for the purposes of seismic assessment of a three-span R/C bridge.” *10th National Conference on Earthquake Engineering2*, Anchorage, Alaska.

Shao, X., and Griffith, C. (2013). “An overview of hybrid simulation implementations in NEES projects.” *Engineering Structures*, Elsevier Ltd, 56, 1439–1451.