

S. Huang, W. Wang, M. Brambley, S. Goyal, W. Zuo 2018. "An Agent-based Hardware-in-the-loop Simulation Framework for Building Controls." *Energy and Buildings*, 181, pp. 26-37.

An agent-based hardware-in-the-loop simulation framework for building controls

Sen Huang¹, Weimin Wang^{2,*}, Michael R. Brambley¹, Siddhartha Goyal^{3,ξ}, Wangda Zuo⁴

¹Pacific Northwest National Laboratory, Richland WA, USA

²University of North Carolina at Charlotte, Charlotte NC, USA

³Johnson Controls Inc., Milwaukee WI, USA

⁴University of Colorado at Boulder, Boulder CO, USA

Abstract: In general, a hardware-in-the-loop (HIL) building simulation has lower cost and fewer practical limitations (e.g., scheduling issues) than field tests in occupied buildings, while also overcoming limitations of simulations alone by capturing the full behavior of some physical systems, equipment, and components. However, the implementation of an HIL can be difficult due to the scarcity of appropriate tools. This paper presents an agent-based framework for HIL simulation. It can be used for investigation of controller performance via controller-in-the-loop simulations and also HIL for system synthesis. In the latter case, both controllers and major equipment participate in tests to ensure that dynamics of equipment operation are correctly captured in addition to controller performance. The HIL simulation framework presented allows such actual physical parts to be included in the framework while representing others for which behaviors are better known and modeled in simulation models. The mechanism implemented in the framework to synchronize simulations in software with real-time operation of physical equipment is described. As an example, use of the HIL simulation framework is illustrated through a brief study of speed control of the supply fan in the air handling unit of a variable-air-volume building heating, ventilating and air-conditioning system.

Keywords: *Hardware-in-the-loop simulation; VOLTRON; Modelica; building controls; multi-agent systems*

* Corresponding author. Email: Weimin.Wang@uncc.edu; Phone: 1-704-687-5066

^ξ Completed the work in this paper while working formerly at Pacific Northwest National Laboratory.

1. Introduction

In 2016, buildings were responsible for nearly 40% of the primary energy use and 36% of CO₂ emissions in the U.S. (EIA 2017). Similar situations can be found in other countries (EIA 2016, Li et al. 2011). As a result, it is widely accepted that increasing energy efficiency in buildings is one of the foremost important approaches to address global climate change and other energy-related environmental issues facing the world today (McNeil et al. 2016, Shi et al. 2016). Advanced controls across individual components (e.g., chillers, boilers, and fans), different systems (e.g., lighting and heating, ventilation, and air conditioning [HVAC] systems), and whole buildings, are commonly regarded critical to improve and maintain the operational efficiency of buildings (Shaikh et al. 2014).

Any innovative design of equipment and associated controls, once developed, must go through extensive testing and validation to ensure its quality and reliable operation before being deployed in the field at scale. The test and validation of building equipment and controls can be realized through field or full-scale laboratory tests, pure simulation, or hardware-in-the-loop (HIL) simulation.

A field test evaluates equipment and controllers in real operating conditions in the field. For example, Shim and Sone (2014) implemented three different fan control strategies on a variable-air-volume (VAV) system to measure the impacts on energy performance and thermal comfort satisfaction. Wang et al. (2015) reported a field test and evaluation of advanced control strategies for a total of 66 packaged rooftop units on 8 different buildings. Although not ordinarily occupied, research facilities (Bhandari and Buckberry 2012, McNeil et al. 2014, PNNL 2017) built to ease pilot testing of a variety of building technologies can be used to perform full-scale laboratory tests under realistic conditions. Because the equipment and controllers are tested and evaluated in their realistic operating conditions, the results from field testing are usually more convincing indicators of the performance than results from simulations.

Although field tests are the most reliable approach for performance evaluation, there are limitations associated with them. First, field tests in real buildings are costly and subject to many constraints. The range of parameter settings permissible and the number of tests that can be run over a specific time period are generally quite limited. There are more challenges to field testing if the control scenarios require subjecting equipment to destructive events. For example, evaluating fault detection methods may require imposing operational faults on equipment and systems, potentially decreasing equipment life or causing discomfort for occupants. Secondly, it is difficult to conduct

repetitive evaluations under identical operating conditions (e.g., weather and occupant presence), thereby making comparisons of different control methods or different products that provide the same function (e.g., pumps from three different manufacturers) challenging.

Computer simulation is another method widely used by researchers to develop and evaluate innovative control methods for many different applications in buildings. In simulation, numerical models are used to capture the dynamics of equipment and building operation in response to weather, internal loads, occupant behavior, and other stimuli. Once developed and validated, the numerical models can be used repetitively to evaluate building equipment and controls at a significantly lower cost compared to field tests, providing a key advantage especially for studies on large numbers of buildings of different designs in many different climate zones. This distinct advantage has led to numerous studies reported in the literature that use simulation models to devise, compare, and evaluate control methods and system design alternatives (e.g., Huang et al. 2016, 2017).

Model fidelity is critical to successful use of simulation in place of field tests. The models must have sufficient fidelity to mimic the real operation of equipment and buildings. Many simulation tools (Crawley et al. 2008) are now available that support validated models in a variety of domains (e.g., HVAC, building physics, and lighting). The advance of co-simulation capabilities (Wetter 2011) and computational technologies has expanded the modeling scope further to multi-domain systems at higher resolution.

Despite the plethora of simulation programs, there are some situations with which a simulation cannot easily deal. Numerical models may not capture all the necessary details of equipment or system characteristics and dynamics. For example, a controller may contain proprietary algorithms that are not available for use in simulation models. The behavior of such controllers cannot be modeled unless the algorithm owners reveal the algorithms adequately in patents or publications after protecting the algorithms. In addition, it is usually difficult to model the impact of controller hardware features (e.g., memory and processor speed) on the overall system performance (Bushby et al. 2010).

The methods of field testing and simulation can complement each other by synergistically combining hardware and software together to form an HIL simulation. Isermann et al. (1999) defined an HIL simulation as the operation of real physical components in connection with real-time simulated components. Which control-loop components are

selected for real hardware depends entirely on the experiment needs, but the parts (e.g., actuators) that may not guarantee acceptable accuracy through modeling are normally replaced with actual hardware in HIL simulations.

In comparison with field tests, HIL simulations offer the following major advantages (Fathy et al. 2006, Isermann et al. 1999): 1) the cost and time required to design and build prototype controllers are lower because less hardware is used; 2) hardware, including controllers and the equipment and systems controlled, can be tested beyond normal operating conditions (e.g., under sensor faults, actuator failures, and extended operation ranges) without destructive consequences; 3) experiments can be readily replicated with the same boundary conditions. In comparison with pure simulations, experiments with HIL can significantly increase the fidelity of results by incorporating hardware that is difficult to model. Moreover, with HIL, physical controllers with proprietary algorithms can be tested and evaluated without knowledge of the specifics of the algorithms because only their interfaces with the connected simulators are required.

HIL simulation has been extensively used in the automotive industry (Fathy et al. 2006, Isermann et al. 1999) and power systems design and analysis (IEEE PES Task Force 2015). In contrast, HIL simulation is used much less but has gained increasing attention in the field of building controls. In this paper, we describe an agent-based HIL simulation framework for building controls. The framework has three advantages compared to simulation or field testing alone. First, the communication between hardware and software is built on existing prevalent communication protocols. Therefore, no analog-digital converters are needed and thereby reduces the cost of HIL implementation. Second, it is flexible with respect to data exchange between hardware and simulations. The communication tunnel is independent of the data that it transfers. A large amount of data can be easily exchanged between the hardware and the simulations. Third, it supports several common simulation tools including EnergyPlus (DOE 2018), Dymola Modelica (Dassault Systems 2017) and MATLAB (MathWorks 2017a).

The rest of this paper is organized as follows: Section 2 provides an overview of previous work on HIL simulation for building controls. The agent-based HIL simulation framework is described in Section 3 with a focus on its components and the time synchronization approach. Section 4 demonstrates the use of the HIL simulation framework for the evaluation of supply fan controls in a VAV system. The paper ends with some conclusions and identification of future work in Section 5.

2. Review of HIL Simulation for Building Controls

In a review of HIL simulation in the automotive area, Fathy et al. (2006) noted the trend of HIL applications changing from controller-in-the-loop (CIL) simulations simply for controller validation to a tool for system synthesis. In CIL simulation, a controller, which is an electronic control unit including both the control hardware and the implemented control algorithms, is integrated with simulations of the devices and systems controlled. CIL simulations have been used primarily for controller design and testing ever since the inception of HIL. In contrast in system synthesis, major pieces of equipment beyond controllers are integrated with high-fidelity models for simulation to provide realistic operating conditions for the tested physical equipment. The transition from CIL simulations to HIL for system synthesis can also be observed in HIL use for building controls, as discussed in Sections 2.1 and 2.2, which follow. For the convenience of discussion, we hereinafter distinguish the term “physical equipment” from controller hardware as follows: physical equipment refers to the hardware that performs specific physical tasks (e.g., hot water generation by a boiler and water movement by a pump) while a controller refers to an electronic control unit that includes both the control hardware and the implemented control algorithms.

2.1 Controller-in-the-loop simulation

In the context of building controls, a CIL simulation consists of a real-time simulation of the building shell, internal loads, and HVAC and/or other systems together with a hardware interface that couples the simulation to a controller. A number of CIL studies (e.g., Dexter and Haves 1993, Vaezi-Nejad et al. 1991) have reported evaluations of control devices in building energy management systems. These CIL studies vary in design details, but a common characteristic of them is closed-loop or bidirectional interactions between the simulation and the controller (Figure 1). In this closed loop, analog signals (e.g., the commands for damper and fan speed modulation) produced by the controller are converted to digital signals via analog-to-digital (A/D) converters and the digital signals are then fed into the simulation. Similarly, sensor values (e.g., temperatures, pressures, and flow rates) from the simulation are converted to analog signals that are sent via wires to the sensor input terminals of the control hardware.

The architecture shown in Figure 1 was used in many earlier studies that successfully demonstrated the significance of CIL simulation in control system product testing (Lahrech et al. 2002, Riederer et al. 2005), education and training of engineers and technicians (Xu and Haves 2004), control loop parameter tuning (Dexter and Haves 1993), and communications bridge development (Bushby et al. 2010). In particular, the Virtual Cybernetic Building

Testbed (VCBT) provides multiple tools to simulate HVAC, building shells, air flows, and fire, which can be distributed across multiple computers for parallel processing to speed up the computation for integration with building controllers (Bushby et al. 2010). As an impactful example, the VCBT has proven the importance of CIL simulation to support the research and development of cybernetic building technologies.

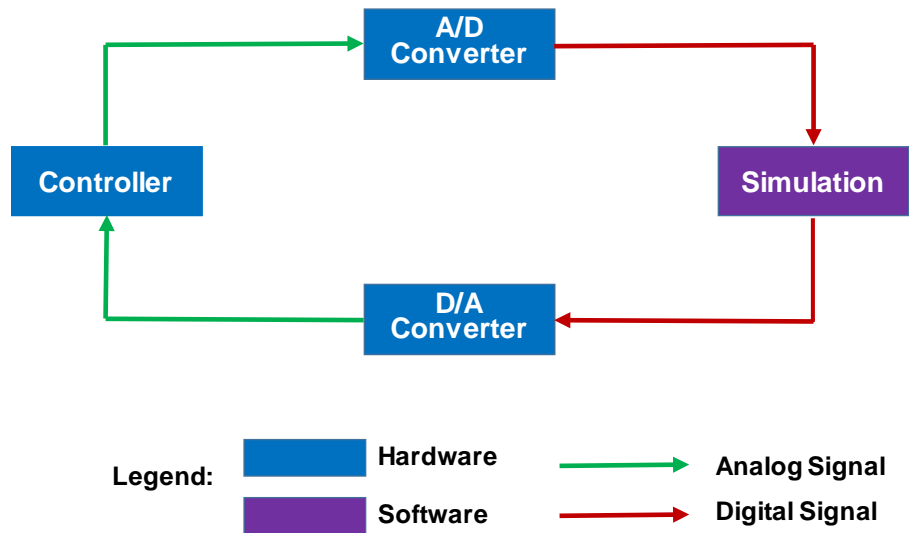


Figure 1: Controller-in-the-loop simulation using hardware interface

A major drawback of the CIL architecture shown in Figure 1 lies in its use of hardware interfaces (A/D and D/A converters). With the hardware interface, the rate at which data can be exchanged is restricted by the number of channels on the data acquisition system to which sensor inputs and control outputs can connect. In addition, changing the control hardware for a new test may involve rewiring and reconfiguring the hardware interface (e.g., for analog inputs resetting the type of input, voltage or current, and the range of the input).

Recognizing this problem with the scalability of the hardware interface, some researchers (Pan et al. 2011, Pang et al. 2012, and Pang et al. 2016) proposed a CIL architecture using a software interface (Figure 2). The software interface is middleware having the following functionalities: 1) communicate with both the control hardware and the simulation for the desired inputs and outputs and 2) synchronize the sampling time step used by the controller with the computing time step used by the simulation.

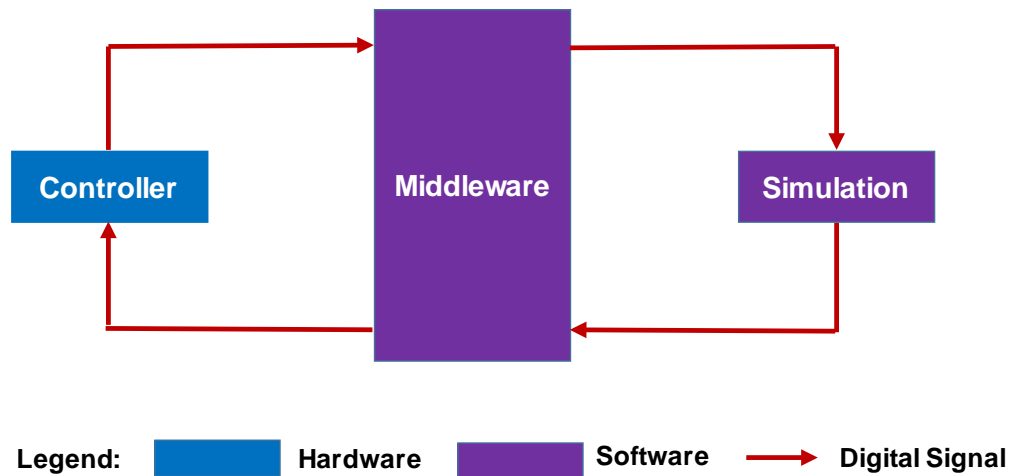


Figure 2: Controller-in-the-loop simulation using a software interface

In the work by Pan et al. (2011), an Open Platform Communications (OPC)-based middleware was developed to couple TRNSYS simulation software (SEL 2017) with the Siemens APOGEE building automation system (Siemens 2017). Specifically, a customized TRNSYS Type was developed to call OPC clients that can read/write BAS point values from/to the OPC server installed on a BAS workstation. The OPC server writes and reads data in controllers connected to the BAS workstation. Using the resulting CIL simulation test bed, Pan et al. (2011) evaluated the impact of different control strategies on HVAC energy consumption for a 3-story office building. Pang et al. (2012) proposed a real-time simulation framework using the Building Controls Virtual Test Bed (BCVTB) (Wetter 2011) as the middleware. The BCVTB has BACnet modules that allow direct reading from and writing to BACnet-compatible controllers. Furthermore, the BCVTB provides built-in linkages to connect simulation software such as EnergyPlus, TRNSYS, and MATLAB. The co-simulation capability of the BCVTB makes it straightforward to realize time synchronization between control hardware and simulation programs. Pang et al.'s BCVTB-based real-time simulation framework has been applied for HVAC system commissioning (Moon et al. 2015), automated fault detection (O'Neill et al. 2013), and model predictive control development (Kwak et al. 2015). All these applications have unidirectional interactions between the control hardware and the simulation: inputs from the controller drive the simulation, but simulation outputs do not affect the controller. In a subsequent study, Pang et al. (2016) enhanced the BCVTB-based real-time simulation framework in two respects: 1) use of the Functional Mockup Interface (FMI) (Dassault Systemes 2017) to couple any simulation programs that support the FMI protocol and 2)

leverage the Simple Measurement and Actuation Profile software (Dawson-Haggerty et al. 2010) for communication and data exchange among different physical instruments, to integrate building controllers.

2.2 HIL simulation for system synthesis

In HIL simulations for system synthesis, major physical equipment (e.g., chillers, heat pumps, boilers, and pumps) and their associated controllers are integrated with simulated devices or building spaces to investigate system behavior under realistic dynamic conditions. This is particularly important when the behavior of hardware cannot be captured adequately with a pure simulation in which case the actual equipment or component is included as hardware in the loop. Such experiments require more than CIL simulations, which communicate electrical signals between the simulation and the controller. Physical parameters such as water temperature and flow rate must be physically created based on the simulation outputs. For example, if a boiler is to be tested in an HIL simulation having models to simulate the heating load and the water distribution network in a building, a mechanism must be established to create water at the proper boiler input temperature and flow rate based on the simulation output. Such a mechanism might be achieved through the use of an electric water heater, a pump, and mixing valves. Rhee et al. (2014) called physical components and equipment that assist in realizing the boundary conditions of the real hardware test “quasi-real” hardware. The number of studies that use HIL for HVAC system synthesis is increasing. These studies, some of which are reviewed next, differ in the type and scope of the hardware, simulation tools, and the strategies to integrate and synchronize hardware and simulations.

Several studies (e.g. Lerch et al. 2014, MacDonald et al. 2015, de Silva and Knabe 2003) developed HIL simulation facilities based on TRNSYS and LabVIEW (National Instruments 2017). In the work by de Silva and Knabe (2003), a boiler and its controller were used as the hardware while TRNSYS was used to simulate boiler operating conditions in a house to evaluate the boiler control method. LabVIEW was used for data acquisition and control of the test rig to reproduce the desired return water temperature and hydraulic resistance of the hot water circuit as simulated by TRNSYS. A new TRNSYS type was developed to slow the simulation to real time and to communicate with LabVIEW. Lerch et al. (2014) evaluated heat pump performance under simulated operating conditions that the heat pump may be subjected to in the field. A new TRNSYS type was developed to interface with LabVIEW. Transmission Control Protocol (TCP) sockets were used for reliable communication and synchronization between the two software programs. MacDonald et al. (2015) described a detailed approach for real time simulation

using TRNSYS and LabVIEW to test the performance of water-source heat pumps in emulated operating conditions. The time taken to complete a TRNSYS time-step calculation was monitored. If the calculation time exceeds the LabVIEW sampling period, the measured data sent to TRNSYS is no longer synchronized with the TRNSYS timestep and the user is alerted about this situation. Although TRNSYS and LabVIEW are commonly used in HIL simulations for system synthesis, neither provides a readily available solution for their coupling. Customized TRNSYS types for real time simulation have been created from scratch for specific project needs, which is not trivial work. To avoid dealing with the issues of communication and synchronization between different software, Rhee et al. (2014) converted their MATLAB-based simulation models to LabVIEW programs when developing an HIL simulation facility for hydronic radiant floor heating systems.

In addition to TRNSYS, Dymola Modelica and MATLAB Simulink are the other two programs frequently used for simulation in HIL for system synthesis. Schneider et al. (2015) presented an HIL test bench for pumps, where models are used for a family dwelling and its HVAC system while the hardware includes a pump and the necessary “quasi-real” hardware to create the desired pump inlet conditions (i.e., temperature and pressure). They implemented a socket-based data interface using the TCP/IP protocol for the communication between Dymola and LabVIEW. Also using Dymola and LabVIEW, De La Cruz et al. (2017) described a test bench platform built and tested to perform HIL simulation for air-to-water heat pumps. An OPC server was employed to bridge the communication between Dymola and LabVIEW. Their HIL test platform includes a heat pump, a hot water tank, pumps, and valves, representing an implementation for whole system testing (Haller et al. 2013).

3. Agent-Based HIL Simulation Framework

The review of HIL simulation studies in Section 2 showed that previous work concentrated on either the controller only or system synthesis in laboratory experiments. LabVIEW was used in laboratory experiments for monitoring and controls, and it was coupled with a specific simulation program using various communication and time synchronization techniques. The new agent-based HIL simulation framework presented in this paper aims to provide a generic, scalable, yet easily customizable solution for HIL simulation that can 1) reduce the cost of HIL implementation; 2) support different hardware, including controllers, physical equipment, and physical systems in operating buildings, 3) support high volumes of data exchange between hardware and simulations; and 4) accommodate the use of several different simulation programs.

The HIL simulation framework uses middleware to interface between the hardware and the simulations. The middleware is built on VOLTTRON™ (Katipamula et al. 2016), an agent implementation and execution platform providing many services to easily communicate with physical devices (i.e., control hardware) and other resources (e.g., weather stations and utility incentive signals). In this section, a brief introduction of VOLTTRON is given first. Then, the framework structure and components are presented, followed by a description of the mechanism for time synchronization between hardware and simulation.

3.1 VOLTTRON

Implemented with the Python programming language and run in the Linux environment, VOLTTRON was initially developed in 2012 as an agent-based distributed control and sensing software platform for the power grid. Since then, it has been significantly enhanced with target applications for buildings, the power grid, and the integration of buildings with the grid. Several projects demonstrated the use of VOLLTRON to ease deployment of advanced control strategies for energy efficiency and grid services (Luo et al. 2017, Katipamula et al. 2016).

The VOLTTRON platform comprises a number of components and agents that provide services to other agents. Several key components and service agents related to this work are described below while more detailed descriptions of the VOLTTRON platform can be found in its online documentation (VOLTTRON 2018).

- Information exchange bus. The VOLTTRON message bus enables agents and services to exchange data using the publish/subscribe (pub/sub) messaging pattern. Any agent or service can publish a message formulated to consist of a topic, message body, and message content without specific recipients identified. Once the message is published on the information exchange bus, the agents that subscribed to the message topic are notified and receive the published message.
- Remote procedure calls (RPCs). VOLTTRON agents can interact directly via remote procedure calls, which are especially useful for implementing event-driven agent activities.
- Drivers. VOLTTRON has built-in drivers to support communication with physical devices that implement the BACnet or Modbus protocol. Because these two protocols are widely used in building automation systems, VOLTTON makes communication with a broad set of heterogeneous devices in buildings straightforward.
- Actuator agent. The VOLTTRON actuator agent accepts commands from other agents and issues the commands to a specified device. In other words, agents intended to control devices must publish a request to the actuator

agent. The actuator agent then forwards the command to the driver interfacing with that device if the requested access to the device is allowed.

The VOLTRON platform itself has low CPU, memory, and storage requirements, so it can be installed on a variety of small-form-factor computers such as the Raspberry Pi (Raspberry Pi Foundation 2017). These small-form-factor computers provide cost-effective processing platforms for HIL simulation.

3.2 Framework structure and components

Figure 3 shows the structure of the agent-based HIL simulation framework, which consists of hardware, simulations, and the VOLTRON platform.

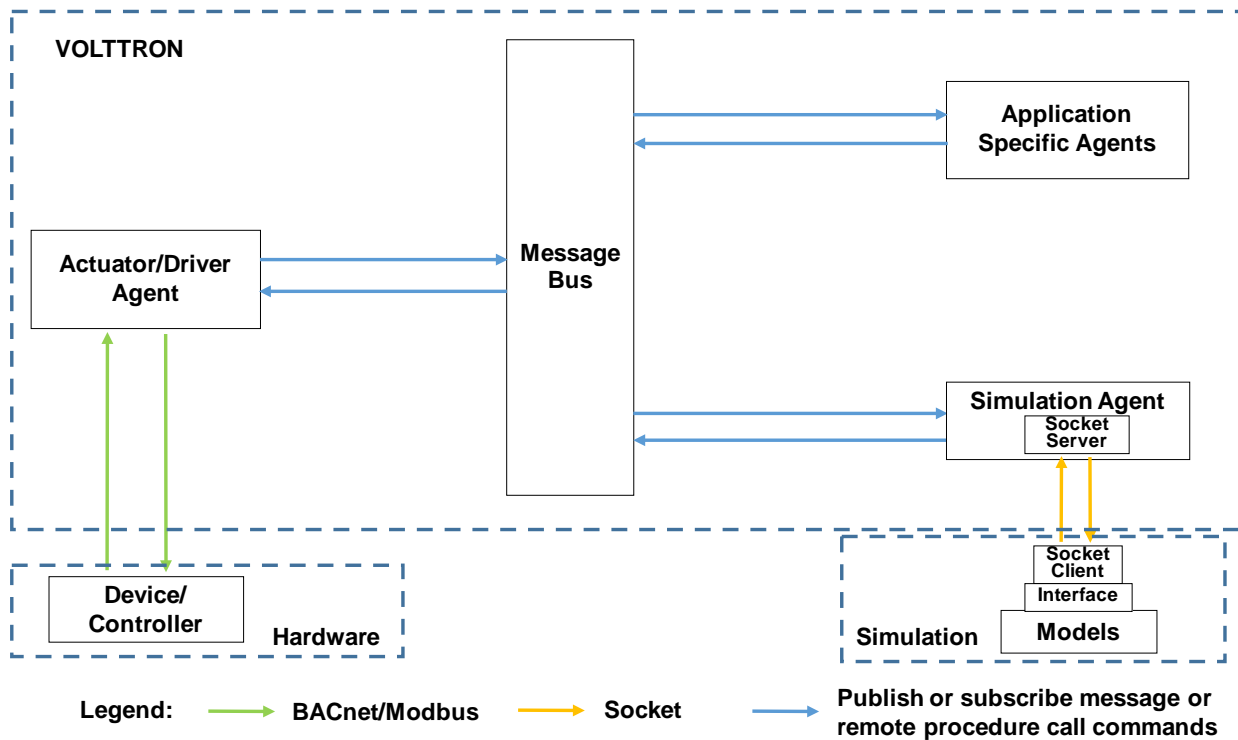


Figure 3: Agent-based framework for HIL simulations

Hardware receives control commands from the simulation and responds to the control commands with new sensed values of variables. In this framework, the hardware can be 1) a controller, 2) a single piece of physical equipment, or 3) the combination of controllers and physical equipment. For the first case of using a controller as the hardware, the controller can be an application-specific controller that associates with an individual piece of equipment or a

supervisory controller that associates with a whole system or even an entire building. The associated physical equipment and system need to be modelled in the simulation, the results of which are used for the controller's inputs. For the second case of using a single piece of physical equipment as the hardware, the equipment's control is modelled as software agents on VOLTTRON. The outputs from the software agents are used to determine equipment operation. For the third case of using the combination of controllers and physical equipment as the hardware, a common application scenario has an entire HVAC system as the hardware and a building modelled in the simulation.

The VOLTTRON platform acts as middleware that couples the hardware and simulation. The actuator/driver agent, an agent readily available in VOLTTRON, is used to communicate with hardware. In addition to the actuator/driver agent, a simulation agent and application-specific agents are added for HIL simulation. The simulation agent serves different roles that depend on the simulation program used. The simulation agent may refer to the building simulation program itself if it is implemented as a VOLTTRON agent. When an external stand-alone simulation program is used, the simulation agent acts as a communication bridge between VOLTTRON and the external simulation program. In this case, the simulation agent may also preprocess data from other agents for use as simulation inputs and post-process simulation outputs for use by other agents. In Figure 3, application-specific agents meet project-specific needs (e.g., implementation of control methods and generation of HIL simulation log messages).

Simulation refers to the external software for modeling building physics and equipment and system operation. The Python Socket package (Python Software Foundation 2017) was leveraged in this work to enable the simulation agent to read outputs of and write inputs to the connected external simulation software. The Python Socket package requires a server-client structure to establish the communication between two Python programs based on the TCP/IP protocol. The server program (i.e., the Simulation Agent in Figure 3) continuously listens to incoming messages while the client program (i.e., Simulation in Figure 3) sends messages to the targeted server according to the specified IP address/port. Currently, the framework supports three kinds of client programs. One is designed for EnergyPlus, which is developed based on the BCVTB. The second is designed for Dymola, which employs the Python interface provided by the Modelica Buildings Library (Wetter et al. 2014), and the third is for MATLAB and uses the MATLAB Engine API for Python (MathWorks 2017b, Sharma 2016).

3.3 Implementation of time synchronization

The hardware sampling frequency and the simulation computation time need to be carefully considered for the purpose of time synchronization. Let Δt denote the time interval at which the data sent from the hardware to the simulation (called “sensed variables”) need to be updated. Time synchronization requires that the data sent from the simulation to the hardware (called “controlled variables”) be updated in the same interval Δt . Because the actual time needed by the simulation to complete one computation time step ($\Delta t_{cpu,i}$) may differ from Δt , a mechanism is needed to speed up or slow down the simulation depending on whether $\Delta t_{cpu,i}$ is greater or less than Δt . The subscript i is used to identify each specific time step because the computation time likely varies across time steps. The case of having $\Delta t_{cpu,i} < \Delta t$ usually holds for building applications, hence it is the focus of this work. Figure 4 illustrates the time synchronization mechanism implemented in the framework.

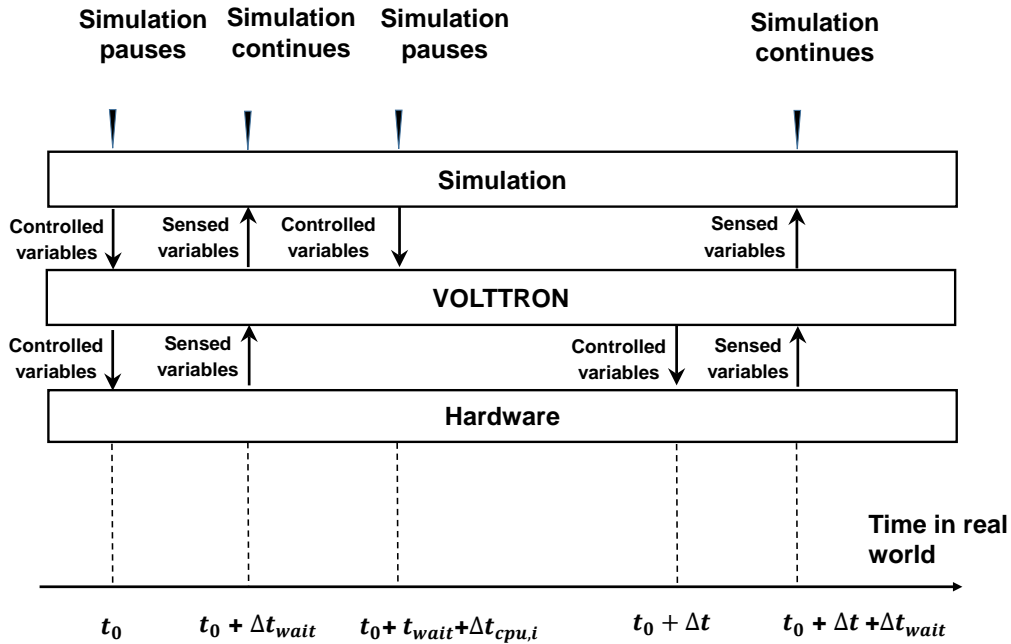


Figure 4: Schematic of time synchronization mechanism

The mechanism is implemented with the following sequence of steps:

- 1) At time t_0 when the HIL simulation begins, the initial values for the controlled variables are sent to the VOLTTRON simulation agent, which then forwards the values to the hardware.
- 2) The hardware responds to the received controlled variables. The driver agent scrapes the values of the sensed variables from the controller and then publishes the scraped variables to the VOLTTRON message bus after waiting Δt_{wait} until time $t_0 + \Delta t_{wait}$. The purpose of Δt_{wait} is to enable sampling of one representative value of each sensed variable in each time step for use by the simulation. Thus, the value of Δt_{wait} needs to be selected large enough to ensure that the hardware response to the controlled variables is settled.
- 3) The simulation agent reads the values of the sensed variables from the VOLTTRON message bus. The simulation continues and completes its computation for the new time step using time $\Delta t_{cpu,i}$ ($i = 1$ for the case shown in Figure 4). The updated values of the controlled variable are sent from the simulation to the simulation agent.
- 4) The simulation becomes idle after completion of its computation in Step 3 (at $t_0 + \Delta t_{wait} + \Delta t_{cpu,1}$) until $t_0 + i\Delta t$, at which time the simulation agent sends the control variable values to the hardware using the remote procedure call.
- 5) Steps 2-4 are repeated for each subsequent hardware time step Δt to move the simulation forward in time until a specified end time.

This synchronization procedure does not consider the communication time between the hardware and the VOLTTRON message bus or the communication between VOLTTRON agents because those communications are fast (taking milliseconds or less). In Step 2, the time of receiving controlled variables by hardware was marked using VOLTTRON's event-driven triggering mechanism (which triggers an action when a certain event occurs). The fixed time step Δt in Step 4 was established for the physical equipment using VOLTTRON's time-driven triggering mechanism (which triggers an action when the system time equals a predefined value).

4. Application of the HIL Framework for Supply Fan Controls

In this section, we apply the HIL simulation framework described in Section 3 to control of the supply fan in variable-air-volume (VAV) systems. The primary purpose of this application is to illustrate the use of the HIL simulation framework.

4.1 VAV fan control

VAV systems are widely used in commercial buildings. Terminal boxes are an integral part of the VAV system. For pressure independent terminal boxes in the cooling mode, the terminal controller normally uses two cascaded proportional-integral (PI) control loops (Figure 5): a zone temperature controller and a damper controller. The zone temperature controller samples the zone temperature measurement by the zone temperature sensor and calculates the required air flow rate (i.e., the supply air flow rate set point) based on the deviation of the space temperature from its cooling set point. The air flow rate set point is then used by the damper control loop, which samples the air flow rate via a differential pressure transmitter in the terminal box inlet and modulates the terminal damper to control the air flow rate. It should be noted that a typical VAV terminal box also has a reheat coil for heating up the supply air from an air-handling unit if the zone served by that terminal box is in heating mode. In the heating mode, there is a different mechanism for generating air flow rate set point (ASHRAE 2015a). To simplify the problem, here we only consider the cooling mode, which matches the experimental setup used in this application.

In a VAV system, conditioned air is supplied by an air-handling unit to zone terminal units at a relatively constant temperature. As cooling loads change, the terminals modulate the air flow rate supplied to their respective zones, which leads to pressure changes inside the supply ductwork. Because the zone thermal loads are not easily measured, a pressure sensor, an example of which can be seen in Figure 8, is usually installed downstream of the air handling unit (e.g., two-thirds of the distance down the supply duct) to measure the air static pressure, which can be used as an indicator of the change in cooling loads (more specifically the collective change in demand for supply-air by the terminal boxes served by the air handling unit). The most commonly used supply-fan control method for VAV systems is to regulate the fan speed to maintain the measured duct static pressure at its set point. When the measured duct static pressure decreases below its set point, a greater air flow rate is needed by the thermal zones served by the VAV system, and the supply fan speed increases. In contrast, when the measured static pressure increases above its set point, the supply fan speed decreases, lowering the static pressure to its set point.

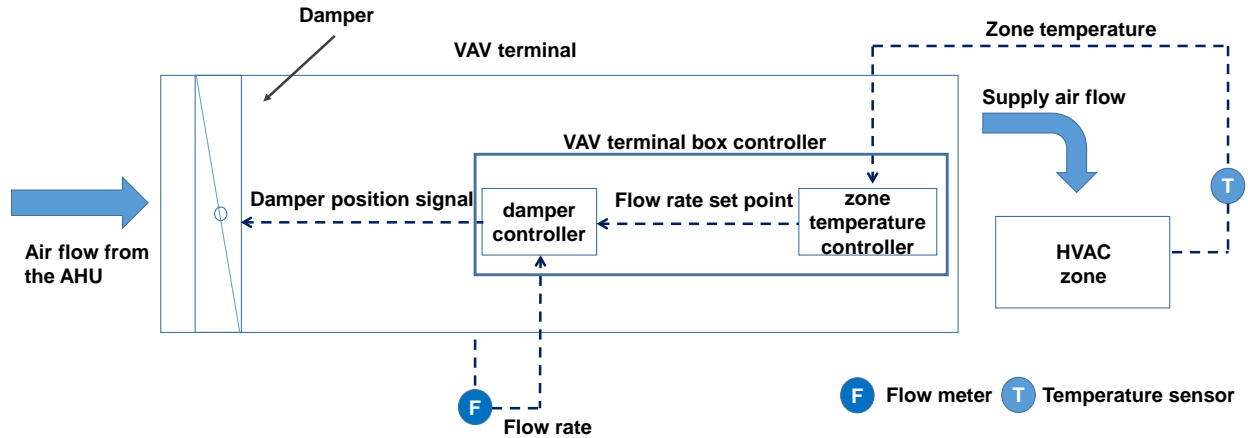


Figure 5: Schematic diagram of VAV terminal configuration and its controls in cooling mode

The duct static pressure set point can be maintained at a constant value corresponding to the design condition, but supply-fan control based on a fixed static pressure set point is not the most energy efficient. At a high fraction of all operating times, the VAV system will require supply-air flow rates less than the designed flow rate and thus run at part-load conditions. To address this, many investigators have explored control methods for static pressure reset to save fan energy (e.g., ASHRAE 2015b, Zhang et al. 2015). The method described in ASHRAE (2015b) is used in this work.

The underlying concept of the ASHRAE strategy for static pressure set point reset is to ensure that at least one of the VAV terminal boxes has its damper close to fully open at all times (corresponding to the least resistance to air flow). As shown in Figure 6, the control method samples the damper position signals of all terminal boxes at a fixed time step (e.g., 5 minutes). At each time step i , if the opening command for the damper with the largest value (Max_Damper in Figure 6) is greater than 98%, the static pressure set point is increased by ΔSP_+ (i.e., $SP_i = SP_{i-1} + \Delta SP_+$); otherwise, if $Max_Damper < 90\%$, SP_i is decreased by ΔSP_- (i.e., $SP_i = SP_{i-1} - \Delta SP_-$). For either case, SP_i should be within its predefined permissible range based on the supply fan operation limits. The recommended values for ΔSP_+ and ΔSP_- are both 5% of the design range for the static pressure.

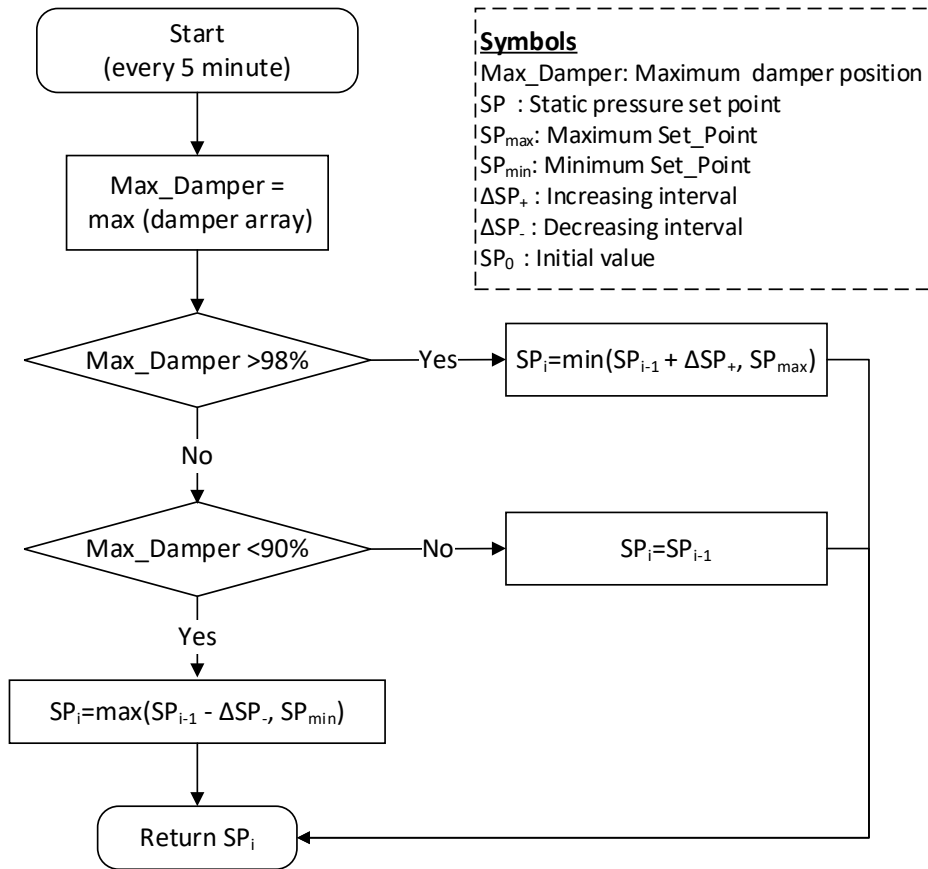


Figure 6: Control logic for the method recommended in ASHRAE (2015) to reset duct static pressure set point in VAV systems

4.2 HIL simulation setup

The HIL simulation was implemented in the Building Diagnostics Laboratory, which is an experimental facility for VAV system control and fault detection and diagnostics located on the campus of Pacific Northwest National Laboratory (PNNL) in Richland, WA. The experimental facility has a total floor area of 53.2 m² (at the time of testing) divided into four thermal zones (Figure 7). Physical partition walls separate Zones 1 and 2 and Zones 3 and 4, but no physical partition exists between Zone 1 and Zone 4 and between Zone 2 and Zone 3. The laboratory has a single-duct VAV system, which consists of an air handling unit, an air-cooled chiller that provides chilled water to the air handler, a chilled water pump, four VAV terminal boxes, and a commercial building automation system (BAS). The instantiation of the HIL simulation framework for this application is described next.

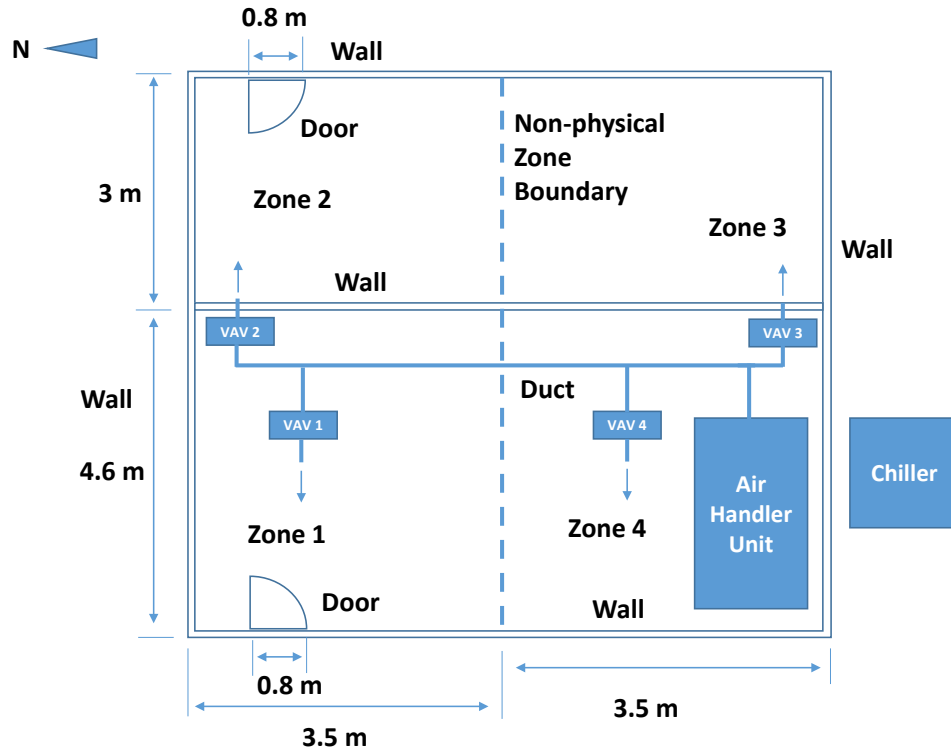


Figure 7: Floor plan of the PNNL Building Diagnostics Laboratory where the HIL simulation experiment was performed

4.2.1 Hardware

This example of HIL simulation was designed to test and evaluate VAV fan controls. Simulations represent the thermal zones, and actual hardware is used for the air-handling unit, the fan controller, and the supply air ducts. To illustrate the flexibility provided by the HIL simulation framework, we purposely split each terminal controller into two parts each corresponding to one of the two cascaded control loops as shown in Figure 5. The zone temperature control loop (i.e., zone temperature controller in Figure 8) is modeled in the simulation, and the damper control loop (i.e., damper controller in Figure 8) is realized through hardware. The flow rate set point determined by the simulation of the zone temperature control loop overrides the corresponding point in the physical controller. Thus, with this setup, the measured air flow rates of all terminals are the sensed variables, whereas the terminal air flow rate set points are the controlled variables. Figure 8 shows the HIL simulation setup, where the hardware is highlighted green.

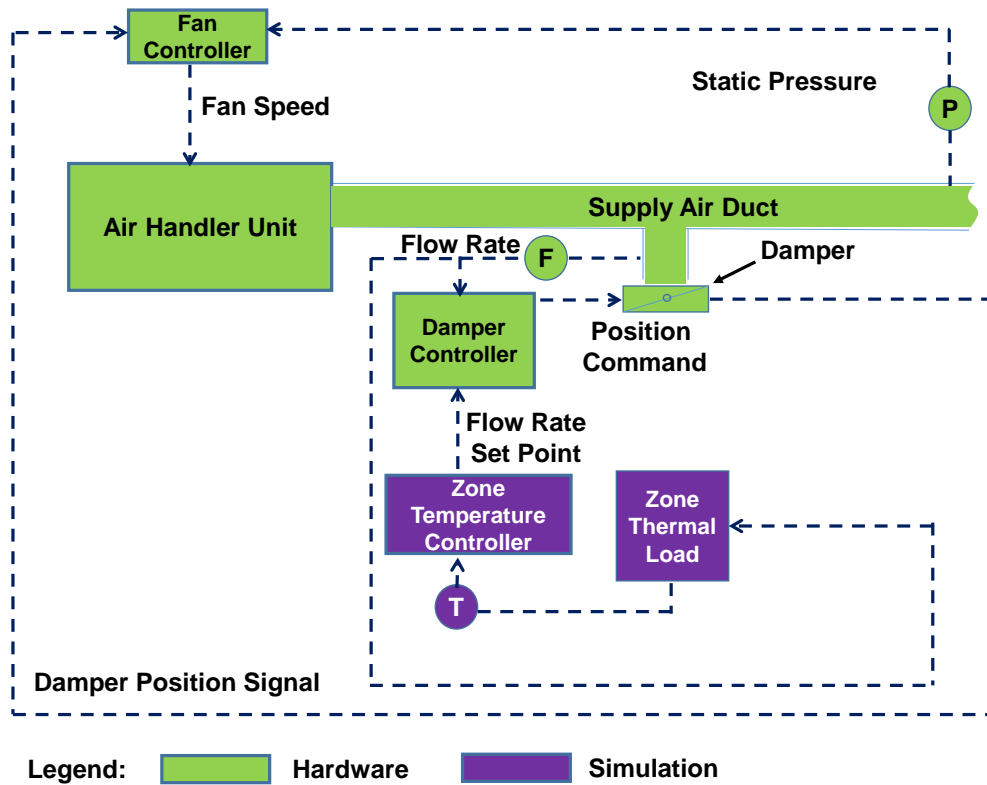


Figure 8: HIL simulation setup for the VAV fan control evaluation

4.2.2 Simulation

The zone temperature control loop is modelled in Dymola/Modelica using component models from the Modelica Buildings Library Version 3.0.0 (Wetter et al. 2014). As Figure 9 shows, the Dymola/Modelica model has two inputs (the supply air flow rate and the HVAC load) and two outputs (the supply air flow rate set point and the zone temperature). As mentioned earlier in Section 4.2.1, the supply air flow rate is obtained from the hardware (i.e., the flow rate sensor in the damper control loop) and the supply air flow rate set point is sent from the simulation to the hardware. The supply air temperature is held constant at 12.8 °C.

An EnergyPlus model for the simulation of zone HVAC loads is coupled with the Modelica model for the simulation of zone temperature control loops via a functional mockup interface (Wetter et al. 2014). The EnergyPlus model was built from the test facility's geometry (Figure 7) and its construction properties. Internal load profiles from the DOE reference model for warehouses (Deru et al. 2011) were assumed. However, no effort was made to

calibrate the EnergyPlus model to the actual building because the accuracy of thermal load simulation is not critical for this work.

The co-simulation between EnergyPlus and Dymola/Modelica intends to consider the dynamic interactions between the zone thermal loads and the VAV system operation. The Modelica model calculates the zone temperature based on the thermal loads received from the EnergyPlus model. Similarly, the EnergyPlus simulation calculates the thermal loads based on the zone temperatures received from the Modelica simulation. Because the EnergyPlus model and the Modelica model are loosely coupled, we used one minute (which is the minimum time step accepted by EnergyPlus) as the data exchange interval to mitigate the negative impacts of loose coupling on simulation accuracy.

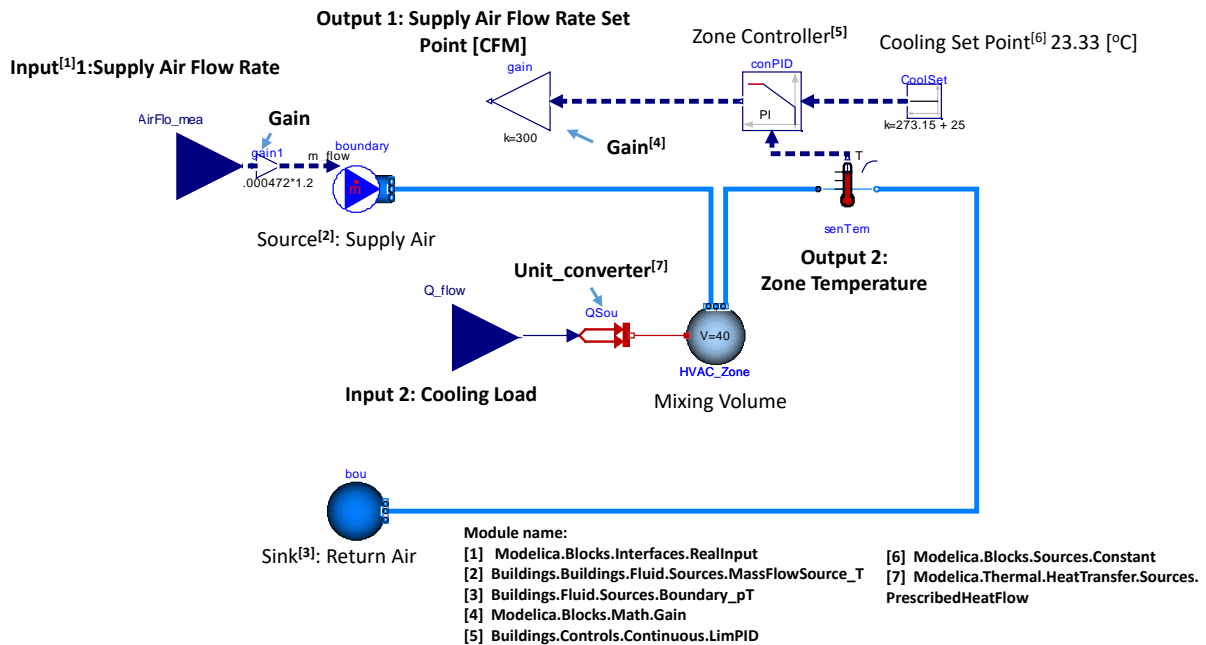


Figure 9: Zone temperature control loop model

4.2.3 VOLTTRON platform instantiation

Figure 10 shows the instantiated HIL simulation framework for the application of VAV fan controls. The following three agents are customized for this application:

- Reset agent. This agent is an example of an “application-specific agent” shown in Figure 3. The fan control agent contains the logic used to reset the duct static pressure set point as explained in Section 4.1 (Figure

- 6). The agent subscribes to the VOLLTRON message bus for the damper position signals of all four terminal boxes and publishes the static pressure set point values to the message bus.
- Simulation agent. The agent communicates with the Dymola/Modelica software via the Python Socket package, as discussed in Section 3.2. The agent subscribes to the message bus for the measured supply air flow rates for all four terminal boxes and publishes the corresponding terminal air flow rate set points to the message bus.
 - Actuator/Driver agent. This agent communicates with the supervisory controller of the facility building automation system via BACnet. The agent publishes the values of the BAS points for the terminal damper position commands and the terminal air flow rates, which are used in this application. At times $t_0 + i \times \Delta t$ (Figure 4), the agent also overrides the air flow rate set point of the damper control loop in each terminal box and the static pressure set point of the supply-fan controller.

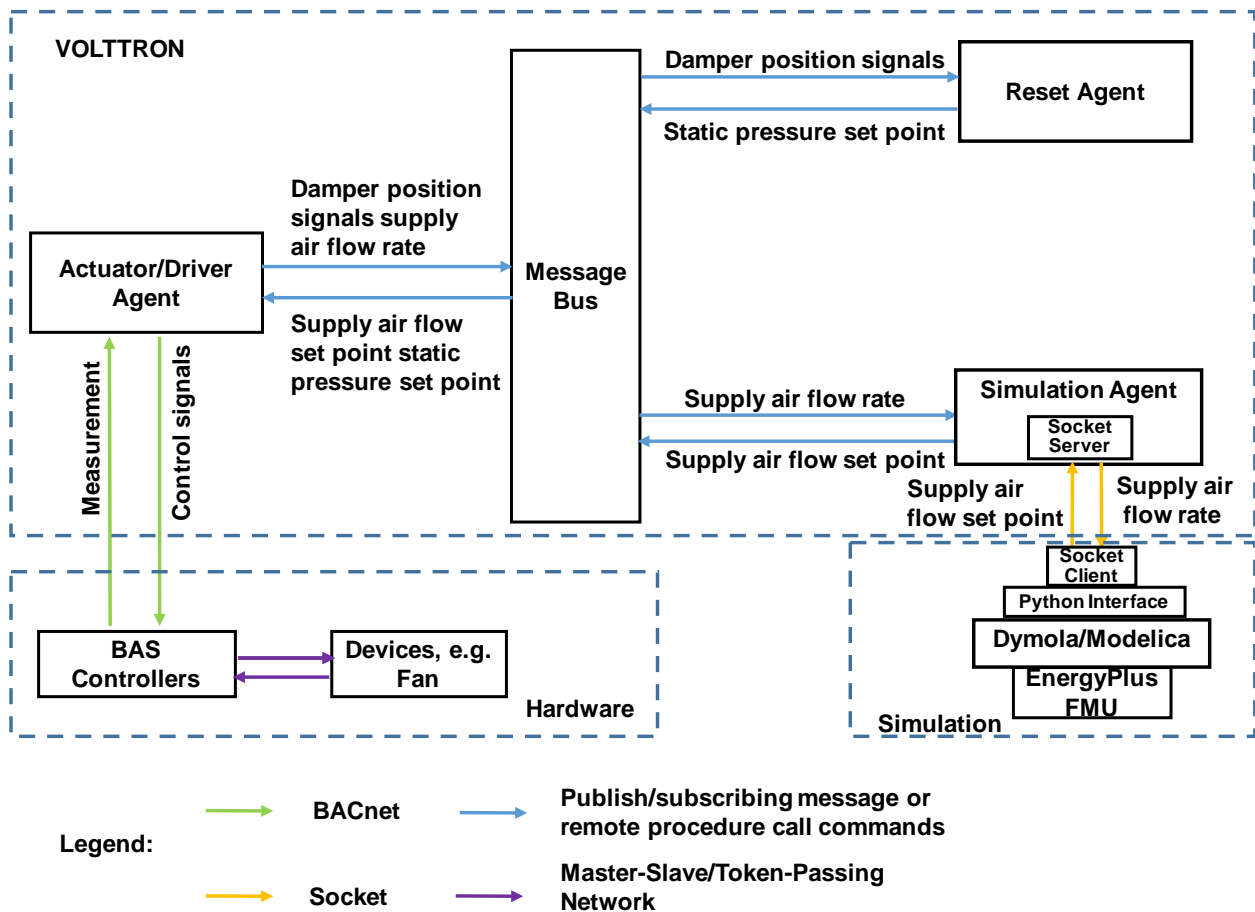


Figure 10: Instantiation of the HIL simulation framework for VAV fan controls

4.3 Verification of data exchange between hardware and simulation

Before running the HIL simulation as proposed in Section 4.2, we verified the correct data exchange between hardware and simulation software. In this verification effort, a single VAV terminal box (i.e., VAV 2 in Figure 7) was used as the hardware in the loop. A simple simulation model was created to emulate a hypothetical zone served by VAV 2. Since the purpose is to verify data exchange, the simulation model has no intent to capture the physical conditions of the actual lab space. Instead, a sinusoid load profile with the peak load of 1.6 kW and the cycle time of one day was assumed in the simulation model (see the upper part of Figure 11). We ran two simulation cases. The first case is pure simulation, which assumes ideal air supply to satisfy the load. The second case is HIL simulation, which calculates the air flow rate set point based on the load and then sends result to the VAV terminal box's damper controller (see Figure 8). In the second case, the damper controller adjusts the damper position based on the air flow rate set point received from the simulation and then sends the measured air flow rate to the simulation. The lower part of Figure 11 compares the air flow rate and its set point between the two simulation cases. This figure shows that 1) for the case of pure simulation, the air flow rate perfectly matches its set point because of the assumption of ideal air supply; 2) for the case of HIL simulation, the air flow rate closely but not perfectly follows its set point because of the deviation of air flow measurement and the latency between damper control signals and movements. Moreover, the air flow rate set points fluctuate around the set points used by the pure simulation, which demonstrates the correct functionality of the HIL simulation. In addition, we manually checked and verified that at each time step, the air flow rate set point calculated by the simulation control was received by the VAV 2 controller, and the air flow rate measured by the VAV 2 controller was received by the simulation.

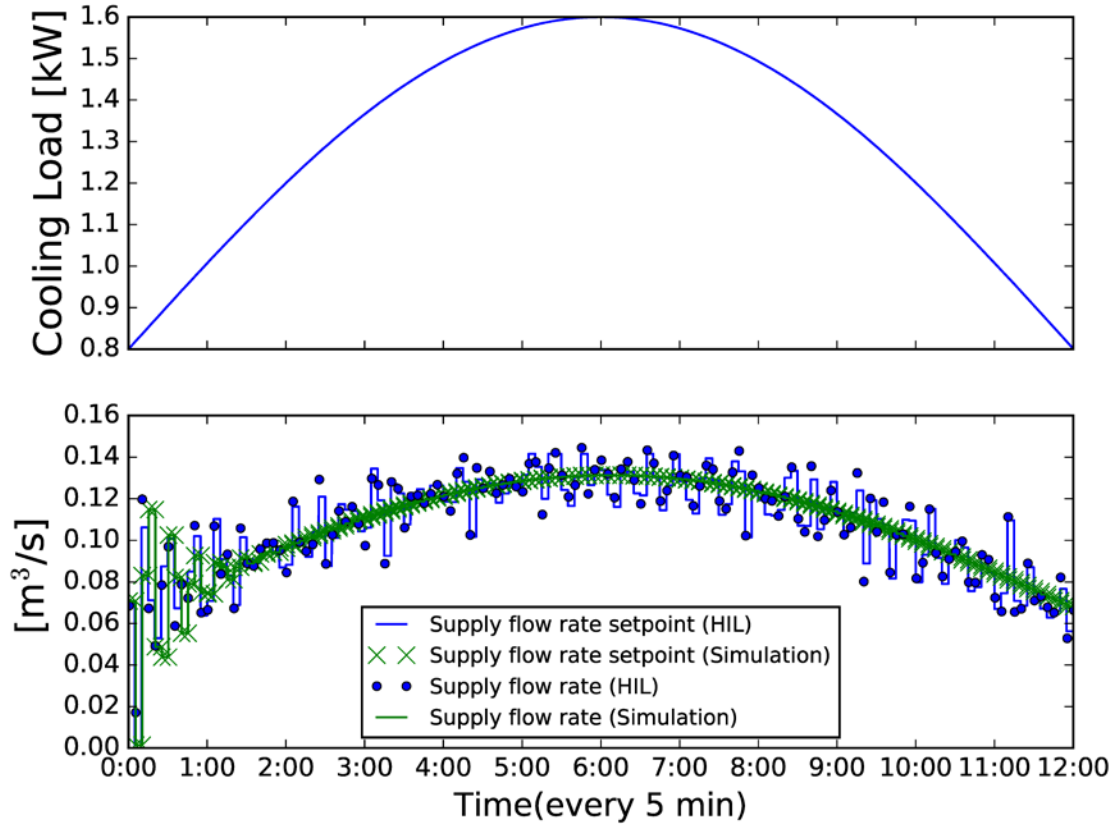


Figure 11: Results from the verification study

5. HIL Simulation Results and Discussion

The simulation was run for one day (Aug. 15) in the cooling season. Instead of using the weather data measured by the BAS, we used the typical meteorological year (TMY3) weather data (DOE 2017) to investigate the impact of supply-fan control in five locations: Richland, WA, Miami, FL, Phoenix, AZ, San Francisco, CA, and Chicago, IL. These five locations have different climate conditions and thereby different weather dependent loads, but identical internal load profiles were used. Parameter settings used for resetting the supply-fan static pressure set point are taken from ASHRAE (2015b) and are listed in Table 1. The sampling time step (Δt) for this HIL simulation is 30 seconds while the constant waiting time (Δt_{wait}) is 5 seconds (see Figure 4).

Table 1: Settings for the HIL Simulation

Parameter	Value	Unit
Step of pressure increase (ΔP_+)	12.4	Pa

Step of pressure decrease (ΔP)	12.4	Pa
Minimum static pressure (P_{\min})	24.8	Pa
Maximum static pressure (P_{\max})	248.8	Pa
HIL simulation time step (Δt)	30	second
Waiting time (Δt_{wait})	5	second

Table 2 summarizes the results on fan energy consumption and fan peak power, which were derived from the fan's variable-speed drive outputs for voltages and currents during all time steps of the simulated period. The results show that for all five locations, resetting the static pressure set point saves more than 50% of the fan energy use with a fixed static pressure set point (baseline), which agrees with the results from previous studies (e.g., Shim et al. 2014 and Zhang et al. 2015). The percentage fan energy savings are greater in cold and mild weather conditions (i.e., Chicago and San Francisco) than in hot weather conditions (i.e., Phoenix, Miami, and Richland), which is expected because the system runs more time at part load conditions in cold or mild climates. Table 2 also shows that static pressure reset can reduce the fan peak power in hot weather conditions.

Table 2: Fan energy consumption and fan peak power for the baseline control (constant static pressure set point at 100 Pa (0.4 inch of water column)) and the ASHRAE control method (static pressure reset based on terminal damper positions) for one day corresponding to August 15. Locations are listed in decreasing order of their annual cooling load.

Location	Fan energy consumption [kWh]		Fan energy consumption change [%]	Fan peak power [W]		Fan peak power change [%]
	Baseline*	ASHRAE reset		Baseline	ASHRAE reset	
Phoenix	2.01	1.00	-50	110	90	-18
Miami	2.01	0.85	-58	90	80	-11
Richland	1.98	0.88	-56	100	90	-10
Chicago	1.90	0.62	-67	80	80	0
San Francisco	1.90	0.67	-65	80	80	0

*In the baseline, we disabled the reset agent and kept the static pressure set point at 100.0 Pa

Results for the Richland location are used in the discussion that follows as an example to show some details of the static pressure reset control strategy. As Figure 12 shows, the supply-fan static pressure was maintained around 100 Pa for the baseline. For the reset control, the supply-fan pressure fluctuated between 24.8 Pa and 90 Pa from 10:00 to 20:00 while stayed at its minimum setting (i.e., 24.8 Pa) for other times. This frequent and abrupt fluctuation of static pressure indicates that the parameters (reset interval, ΔSP_+ , or ΔSP_-) for the supply-fan static pressure set point reset need to be further tuned. Following the same pattern as that for the static pressure set point, the supply-fan power fluctuates between 25 W and 90 W for the reset control, while varying between 70 and 90 W for the baseline.

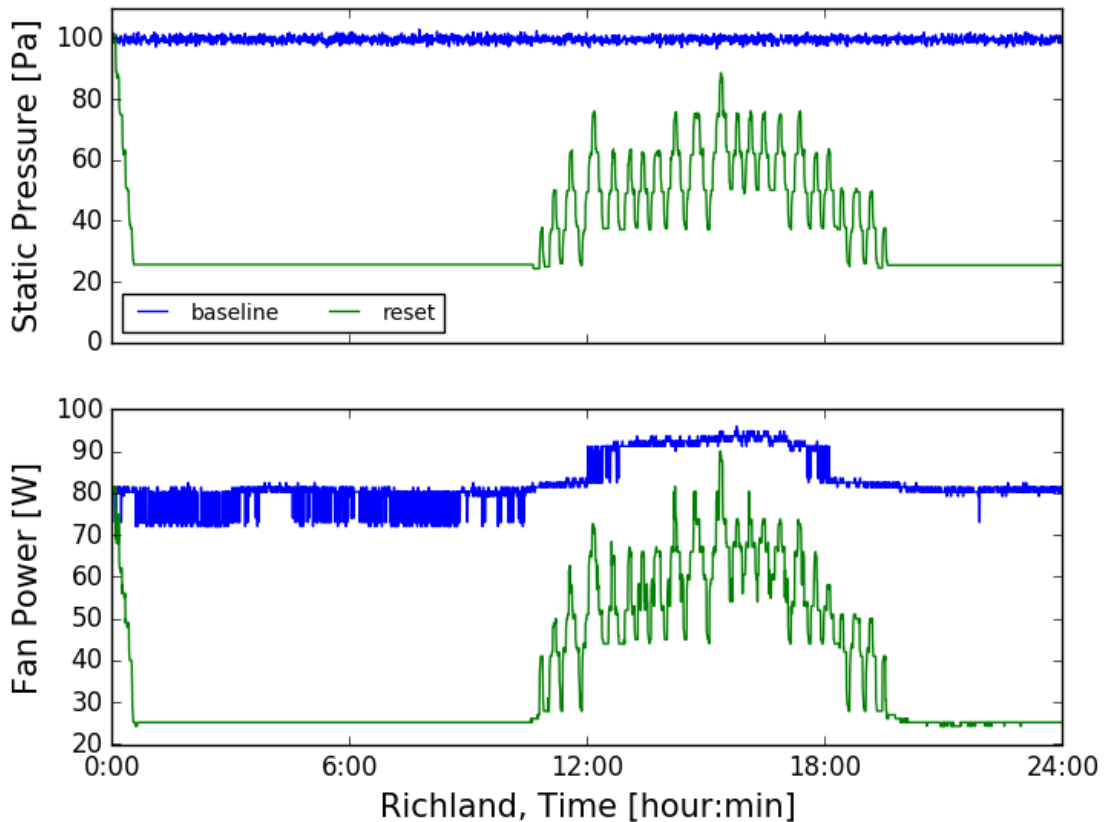


Figure 12: Comparison of fan static pressure and fan power for the baseline control and the ASHRAE control method for the HIL simulation experiment in Richland, WA. Data are collected at a 0.5 minute time step, and time is measured in hours:minutes after midnight.

6. Conclusions and Outlook

HIL simulations are increasingly used for investigations of building controls to overcome the limitations associated with simulation or field tests alone. There exist significant variations of the HIL simulation setups that depend on the hardware used, the specific simulation software, and the mechanism of coupling between hardware and software. A flexible and scalable approach to realize HIL simulation is therefore important to accommodate the wide variety of experimentation and testing configurations needed during development and performance characterization and validation of innovative building systems and controls.

The HIL simulation framework described in this paper is based on the open source software platform VOLTTRON (written in Python). Important features of the HIL simulation framework can be summarized as follows: 1) the supported hardware can be a controller, physical equipment of the controlled system, or the combination of both; 2) the supported simulation can be an agent in the platform or an instantiation of any external simulation tool that accommodates socket clients; 3) the capability of interfacing controllers that use BACnet and/or Modbus standard communication enables the framework to easily support HIL simulations for building controls, even including equipment and systems in real operating buildings; and 4) implementing HIL simulation becomes less cost prohibitive than traditional approaches because of the underlying open source VOLTTRON platform and the avoidance of using analog-digital converters.

We applied the HIL simulation framework to evaluating two different VAV fan controls as an example of its use. In this application, the air-handling unit, the fan controller, the supply air ducts, and the damper controllers of VAV terminals were represented by hardware in the loop, while the zones and the zone temperature controllers of VAV terminals were simulated. The results from the HIL example showed that resetting the static pressure set point according to the ASHREA control sequence saved more than 50% of fan energy consumption than the baseline control using a fixed static pressure set point for all five different locations. The example has demonstrated the use of the HIL simulation framework to evaluate building controls. Other supply-fan control strategies and HIL control for other applications can be evaluated in a similar manner.

Currently, a major assumption of the HIL simulation framework presented in this paper is that the simulation time runs faster than the real elapsed time. Although this assumption is valid for most applications in the field of building controls, precautions need to be taken to ensure that the simulation completes its time-step processing before the end of the time interval used by the platform to sample data from the hardware and send them to the simulation. This can

be done by carefully selecting the time step and implementing an automatic check at the end of every time step to identify whether the assumption is violated. If the assumption is violated, the HIL simulation stops and alerts the user of the need to adjust the time interval for sampling data from the hardware. In addition, considering the rich collection of building simulation programs that could potentially be used, flexible mechanisms should be found for the communication between the simulation agent and the external simulation software other than the socket approach used in the work reported in this paper.

References

- ASHRAE. 2015a. ASHRAE Handbook—HVAC Applications, Chapter 47: Design and Application of Controls. ASHRAE, Atlanta, GA.
- ASHRAE. 2015b. ASHRAE Handbook—HVAC Applications, Chapter 42: Supervisory Control Strategies and Optimization. ASHRAE, Atlanta, GA.
- Bushby, ST, MA Galler, NM Ferretti, C Park. 2010. The virtual cybernetic building testbed—A building emulator. ASHRAE Transactions 116(1):37-44.
- Bhandari, M, H Buckberry. 2012. ORNL MAXLAB Flexible Research Platforms. In Proceedings of the 2012 ACEEE Summer Study on Energy Efficiency in Buildings, pp. 13-24 – 13-35. ACEEE, Washington, DC.
- Crawley, DB, JW Hand, M Kummert, BT Griffith. 2008. Contrasting the capabilities of building energy performance simulation programs. Building and Environment 4:661-673.
- Dassault Systèmes. 2017. DYMOLA Systems Engineering. <https://www.3ds.com/products-services/catia/products/dymola/> accessed on October 10, 2017.
- Dawson-Haggerty, S, X Jiang, G Tolle, J Ortiz, D Culler. 2010. sMAP – a Simple Measurement and Actuation Profile for Physical Information. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, pp. 197-210. ACM, New York, NY.
- De La Cruz, AT, P Riviere, D Marchio, O Cauret, A Milu. 2017. Hardware in the Loop Test Bench using Modelica: A Platform to Test and Improve the Control of Heating Systems. Applied Energy 188:107-120.

- Deru, M, K Field, D Studer, B Liu, M Halverson, D Winiarski, et al. 2011. U.S. Department of Energy Commercial Reference Building Models of the National Building Stock. National Renewable Energy Laboratory, Golden CO.
- Dexter, AL, P Haves. 1993. Building control systems: Evaluation of performance using an emulator. Building Service Engineering Research and Technology 15:131-140.
- DOE. 2017. EnergyPlus Weather Data, U.S. Department of Energy. <https://energyplus.net/weather> accessed on May 15 2017
- DOE. 2018. EnergyPlus, U.S. Department of Energy. <https://energyplus.net/> accessed on April 8, 2018.
- EIA. 2016. International Energy Outlook 2016. U.S. Energy Information Administration. [https://www.eia.gov/outlooks/ieo/pdf/0484\(2016\).pdf](https://www.eia.gov/outlooks/ieo/pdf/0484(2016).pdf) accessed on October 15, 2017.
- EIA. 2017. Annual Energy Outlook 2017, Table 2 and Table 18. U.S. Energy Information Administration. https://www.eia.gov/outlooks/aeo/tables_ref.php accessed on July 20, 2017.
- Fathy, HK, ZS Filipi, J Hagen, JL Stein. 2006. Review of Hardware-in-the-loop Simulation and Its Prospects in the Automotive Area. In Proc. SPIE 6228, Modeling and Simulation for Military Applications, 62280E. doi:10.1117/12.667794
- Haller, MY, R Haberl, T Persson, C Bales, P Kovacs, D Cheze, P Papillon. 2013. Dynamic Whole System Testing of Combined Renewable Heating Systems – The Current State of the Art. Energy and Buildings 66:667-677.
- Huang, S, W Zuo, MD Sohn. 2016. Amelioration of the cooling load based chiller sequencing control. Applied Energy 168:204-215.
- Huang, S, W Zuo, MD Sohn. 2017. Improved Cooling Tower Control of Legacy Chiller Plants by Optimizing the Condenser Water Set Point. Building and Environment 111:33-46.
- IEEE PES Task Force on Real-Time Simulation of Power and Energy Systems. 2015. Real-Time Simulation Technologies for Power Systems Design, Testing, and Analysis. IEEE Power and Energy Technology Systems 2(2):63-73.

- Isermann, R, J Schaffnit, S Sinsel. 1999. Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice* 7:643-653.
- Katipamula, S, J Haack, B Akyol, G Hernandez, J Hagerman. 2016. VOLTTRON: An open-sourcesoftware platform of the future. *IEEE Electrification Magazine* 4(12):15-22. DOI: 10.1109/MELE.2016.2614178.
- Kwak, Y, JH Huh, C Jang. 2015. Development of a model predictive control framework through real-time building energy management system data. *Applied Energy* 155:1-13.
- Lahrech, R, P Gruber, P Riederer, P Tessier, JC Visier. 2002. Development of a testing method for control HVAC systems by emulation. *Energy and Buildings* 34:909-916.
- Lerch, W, A Heinz, R Heimrath. 2014. Evaluation of Combined Solar Thermal Heat Pump Systems using Dynamic System Simulations. *Energy Procedia* 48:598-607.
- Li, F, S Dong, X Li, Q Liang, W Yang. 2011. Energy consumption-economic growth relationship and carbon dioxide emissions in China. *Energy Policy* 39:568-574.
- Luo, J, H Pourbabak, W Su. 2017. The application of distributed control algorithms using VOLTTRON-based software platform. In *Proceedings of the 8th International Renewable Energy Congress*. Amman, Jordan. IEEE, New York, NY. DOI: [10.1109/IREC.2017.7926056](https://doi.org/10.1109/IREC.2017.7926056)
- Macdonald, F, K D'Avignon, M Kummert, A Daoud. 2015. A TRNSYS-LabVIEW bi-directional Connection for HVAC Equipment Testing using Hardware-in-the-loop Simulation. In *Proceedings of the 9th International Conference on System Simulation in Buildings*. Atelier des Presses, Liège, Belgium.
- MathWorks. 2017a. MATLAB® website. <https://www.mathworks.com/products/matlab.html> accessed on October 20, 2017.
- MathWorks. 2017b. MATLAB Engine API for Python. <https://www.mathworks.com/help/matlab/matlab-engine-for-python.html> accessed on October 20, 2017.
- McNeil, A, C Kohler, ES Lee, S Selkowitz. 2014. High performance building mockup in FLEXLAB. LBNL-1005151, Lawrence Berkeley National Laboratory.

- McNeil, MA, W Feng, S de La Rue du Can, NZ Khanna, J Ke, N Zhou. 2016. Energy Efficiency Outlook in China's Urban Buildings Sector Through 2030. *Energy Policy* 97:532-539.
- Moon, HJ, MS Choi, SH Ryu, JC Lee, DG Kim. 2015. Case Studies on HVAC System Performance Using A Whole Building Simulation Based Real-Time Energy Evaluation Approach With BEMS. In *Proceedings of Building Simulation 2015*. Hyderabad, India, pp. 2735-2740.
- National Instruments. 2017. LabVIEW. <http://www.ni.com/en-us/shop/labview.html> accessed on October 20, 2017.
- O'Neill, Z, X Pang, M Shashanka, P Haves, T Bailey. 2013. Model-based real-time whole building energy performance monitoring and diagnostics. *Journal of Building Performance Simulation* 7(2):83-99.
- Pan, Y, X Lin, Z Huang, J Sun, O Ahmed. 2011. A Verification Test Bed for Building Control Strategy Coupling TRNSYS with a Real Controller. In *Proceedings of Building Simulation 2011*. Sydney, Australia, pp. 215-222.
- Pang, X, M Wetter, P Bhattacharya, P Haves. 2012. A Framework for Simulation-based Real-time Whole Building Performance Assessment. *Building and Environment* 54:100-108.
- Pang, X, TS Nouidui, M Wetter, D Fuller, A Liao, P Haves. 2016. Building Energy Simulation in Real Time through an Open Standard Interface. *Energy and Buildings* 117:282-289.
- PNNL. 2017. Lab Homes. Pacific Northwest National Laboratory. <http://labhomes.pnnl.gov/> accessed on October 20, 2017.
- Python Software Foundation. 2017. The Python Standard Library 17.2: Socket—Low-level networking interface. <https://docs.python.org/2/library/socket.html> accessed on October 20, 2017.
- Raspberry Pi Foundation. 2017. Raspberry Pi. <https://www.raspberrypi.org/> accessed on October 20, 2017.
- Rhee, K, M Yeo, K Kim. 2014. Development of an Emulation Method for the Performance Evaluation of Radiant Floor Heating Systems. *Building Service Engineering and Research* 35:488-505.
- Riederer, P, F Bruyat, B Cléménçon, S Bernasconi, G Desjardins. 2005. SIMTEST-ELEC: A Virtual Laboratory for the Test of Terminal Controllers for Electric Heating Systems Following a Draft European Standard. In *Proceedings of Building Simulation 2005*. Montréal, Canada, pp. 1027-1034.

- Schneider, GF, J Oppermann, A Constantin, R Streblow, D Müller. 2015. Hardware-in-the-Loop-Simulation of a Building Energy and Control System to Investigate Circulating Pump Control Using Modelica. In Proceedings of the 11th International Modelica Conference. Versailles, France, pp. 225-233.
- SEL. 2017. A Transient Systems Simulation Program. Solar Engineering Laboratory (SEL), University of Wisconsin-Madison.
- Shaikh, PH, NBM Nor, P Nallagownden, I Elamvazuthi, T Ibrahim. 2014. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews* 34:409–429.
- Sharma, P. VOLTTRON integration with MATLAB, FNCS, Energy+ and GridLAB-D. 2016. PNNL-SA-120037, Pacific Northwest National Laboratory, Richland, WA, USA.
<https://energy.gov/sites/prod/files/2016/08/f33/21.%20VOLTTRON%20Integration%20with%20MATLAB%20C%20FNCS%20Energy%20and%20GridLAB-D%20%28Sharma%20PNNL%29.pdf> accessed on October 20, 2017.
- Shi, X, Z Tian, W Chen, B Si, X Jin. 2016. A Review on Building Energy Efficient Design Optimization from the Perspective of Architects. *Renewable and Sustainable Energy Reviews* 65:872-884.
- Shim, G, L Song, G Wang. 2014. Comparison of Different Fan Control Strategies on a Variable Air Volume Systems through Simulations and Experiments. *Building and Environment* 72:212-222.
- SIEMENS. 2017. System Architecture for the APOGEE Building Automation System with BACnet.
<http://w3.usa.siemens.com/buildingtechnologies/us/en/building-automation-and-energy-management/apogee/architecture/Pages/architecture.aspx> accessed on October 20, 2017.
- da Silva, P, G Knabe. 2003. LabHouse: System simulation and emulation within boiler development. *Building Services Engineering Research and Technology* 24(4):281-7.
- Vaezi-Nejad, H, E Hutter, P Haves, AL Dexter, GE Kelly, et al. 1991. Use of Building Emulators to Evaluate the Performance of Building Energy Management Systems. In *Proceedings of Building Simulation 1991*, Nice, France, pp. 209-213.

VOLTTRON Developer Team. 2018. VOLTTRON Documentation Release 5.0-rc2.

<https://media.readthedocs.org/pdf/volttron/develop/volttron.pdf> accessed on April 8, 2018.

Wang, W, S Katipamula, H Ngo, R Underhill, D Taasevigen, R Lutes. 2015. Field evaluation of advanced controls for the retrofit of packaged air conditioners and heat pumps. *Applied Energy* 154:344-351.

Wetter, M. 2011. Co-simulation of Building Energy and Control Systems with the Building Controls Virtual Test Bed. *Journal of Building Performance Simulation* 4(3):185-203.

Wetter, M, W Zuo, T Nouidui, X Pang. 2014. Modelica Buildings library. *Journal of Building Performance Simulation* 7(4):253-270.

Xu, P, P Haves, and J. Deringer. 2004. A Simulation-Based Testing and Training Environment for Building Controls. In *Proceedings of SimBuild 2004*. Boulder, CO.

Zhang, J, X Li, T Zhao, H Yu, T Chen, C Liu, X Yang. 2015. A review of Static Pressure Reset Control in Variable Air Volume Air Condition System. *Procedia Engineering* 121:1844-1850.