Cloud-based Hardware-in-the-Loop testing of building automation controllers

M Maghnie\textsuperscript{1}, F Stegemerten\textsuperscript{1}, A Niewöhner\textsuperscript{2}, M Baranski\textsuperscript{2}, A Kümpel\textsuperscript{1} and D Müller\textsuperscript{1}

\textsuperscript{1} RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustraße 10, 52072 Aachen, Germany
\textsuperscript{2} aedifion GmbH, Hohenzollernring 72, 50672 Köln, Germany

E-mail: marwa.maghnie@eonerc.rwth-aachen.de

Abstract. This paper aims to fill the gap in knowledge about building control testing, thus encouraging control evaluation, to ultimately improve energy efficiency in building operation. A hardware controller and its digital twin are tested in real-time in a separate platform in the cloud, and the results are analyzed to determine what insights can tests on virtual controllers give about the behaviors of their real-life hardware counter-parts. Key performance indicators (KPIs) from literature are used to evaluate the control. It is shown that although the controllers differ in their control output time-series, their overall performance profile along the entire simulation run follows a similar pattern (KPI percent error varies between 3\% and 8\%). This implies that virtual controller testing can be a good indicator of hardware controller behavior.

1. Introduction
To reduce the energy consumption in building energy systems, efficient and well-tuned controllers have to be implemented. Yet when controllers are not tested enough in practice before deployment, this leads to low energy efficiency. One reason is the high effort of comprehensive testing. Automating this process may help in alleviating the effort. Furthermore, conclusions in literature are mainly based on virtual controllers with unverified predictions for hardware controllers [1]. This paper aims to close these gaps.

We present a prototype for a cloud-based framework for controller testing and benchmarking. The controllers are tested with two different approaches: open-loop behavior for different signal types and closed-loop behavior using a virtual building. For the hardware tests, a programmable logic controller (PLC) is tested in Controller-Hardware-in-the-Loop (CHiL) experiments. The open-source automation library AixOCAT\textsuperscript{1} provides the code that runs on the hardware controller, whereas a functional mock-up unit (FMU\textsuperscript{2}) based on the modeling language Modelica represents the virtual controller. Both hardware and virtual controllers are based on proportional-integral (PI) logic, which is the most widely used type of control in industry. A thermal zone model for testing the controllers is adapted from the open-source Modelica library AixLib, developed at the E.ON ERC EBC [2]. These resources are used to benchmark the controllers against a set of key performance indicators (KPIs) from literature.

\textsuperscript{1} github.com/RWTH-EBC/AixOCAT
\textsuperscript{2} fmi-standard.org
2. Related work
Although recently the number of testing tools for building control has grown considerably, here we highlight the main influences on this work. Blum et al. [3] introduced the Python-based framework BOPTEST for testing building control strategies. It includes test cases with different thermal profiles, and relies on KPIs to evaluate control strategies. The KPIs used are mainly practical, which can give building commissioners real-life performance impressions. They also use FMUs to model buildings. The BOPTEST framework requires users to run simulations locally on their own machines. However, BOPTEST is being merged with Alfalfa\(^3\), an open source cloud-based building simulation framework. Their work does not focus on testing hardware controllers. O’Neill et al. [1] and Li et al. [4] presented a series of works to evaluate HVAC KPIs. They reviewed a wide selection of KPIs [1] and evaluated field test results under some of the KPIs [4]. Their proposed system seems versatile yet not open-source.

3. Experimental setup
In the following we briefly describe the individual experiment components: cloud-platform scheme and current state, virtual and hardware controllers, the virtual controlled system, and the KPIs to evaluate the controllers’ performances.

3.1. Cloud platform
In this section, we present the cloud-based framework idea for controller testing and benchmarking, and describe the prototype. In the current version of the framework, the controllers can communicate with virtual controlled systems via a cloud platform, which stores all data and standardizes data exchange by means of a well-defined application programming interface (API). The simulation of virtual controlled systems can be executed in the cloud, which could be beneficial for at least two reasons. Firstly, high computational power can be allocated on demand. Secondly, tests can be executed anywhere in the world and especially in technical facilities. The prototype could be further developed into a scalable web-service for model-based testing. Figure 1 shows the proposed backend concept of the prototype.

An API designed to interact with FMUs executed in the cloud (FMU-API) can start asynchronous simulations, which are executed in individual threads. The API is implemented with the help of the FastAPI\(^4\) framework. The presented FMU database is used for the storage of meta-data of individual FMUs as well as configuration and result storage of performed or running simulations. Results of the simulations are sent to an MQTT broker that is part of the cloud platform. The simulated data can be displayed and analyzed in a web frontend.

\(^3\) github.com/NREL/alfalfa
\(^4\) fastapi.tiangolo.com
The FMU-API allows various HTTP requests. An important API endpoint is a so-called POST endpoint, which is used to execute an existing FMU. Besides a unique identifier, the simulation type is a parameter that allows tailoring the simulation using the following options.

- **default** - Runs a simulation with the default experiment stored in the FMU
- **interval** - Runs a simulation over a defined interval
- **live** - Starts a simulation in real time
- **increased-real-time** - Runs a simulation accelerated by a fixed factor compared to real time

The current state of the prototype is illustrated in Figure 2.

![Figure 2](image)

**Figure 2:** The user terminal communicates with the simulation manager in the cloud via MQTT, where the virtual controller and virtual controlled system are stored and simulated. The Python library FMPy\(^5\) is used for the communication between the virtual controlled system and virtual controller, while the hardware controller and the simulation manager communicate via the python package pyads\(^6\). The simulation data exchanged includes setpoints, measured values, and control outputs.

### 3.2. Controllers

For the hardware controller, we use the PLC model CX9020 by Beckhoff\(^7\). PLCs are utilized frequently in building control \[5\], and their programming is straightforward to learn, which is done within Beckhoff’s TwinCAT3 development environment.

As for the virtual controller, we customize a PI-controller from AixLib to have the same parameters as those of the hardware controller. The virtual controller as well as the virtual controlled system are stored as FMUs, which ensures that the models can be handled in different modeling and simulation tools.

### 3.3. Virtual controlled system

The virtual controlled system is a thermal zone model from AixLib. This particular model uses the RC (Resistance-Capacitance) equivalent method to model internal and external walls. Solar radiation, ambient temperature, and constant internal gains are also taken into account. Within these boundaries, the corresponding heat demand from the thermal zone can be calculated. To regulate the temperature in the thermal zone, the controller adjusts the heat production of an ideal heat source to cover the heat demand. With the use of an ideal heat source, there is no need to examine the influences of elaborate heat producers. Different types of setpoint signals for the temperature of the thermal zone are provided, to operate the controller at its limits.

---

\(^5\) [github.com/CATIA-Systems/FMPy](https://github.com/CATIA-Systems/FMPy)

\(^6\) [github.com/stlehmann/pyads](https://github.com/stlehmann/pyads)

\(^7\) [beckhoff.com/en-en/products/ipc/embedded-pcs/cx9020-arm-cortex-a8/cx9020.html](https://beckhoff.com/en-en/products/ipc/embedded-pcs/cx9020-arm-cortex-a8/cx9020.html)
3.4. Key Performance Indicators

In order to evaluate the control quality, we calculate key performance indicators (KPIs) based on the error between the setpoint and measured value \( e(t) \) at time \( t \) (Table 1).

| Name     | Description                        | Function                  |
|----------|------------------------------------|---------------------------|
| IAE      | Integral Absolute Error            | \( \int |e(t)| \, dt \)     |
| ITAE     | Integral Time-weighted Absolute Error | \( \int t \cdot |e(t)| \, dt \) |
| ISE      | Integral Squared Error             | \( \int e(t)^2 \, dt \)   |
| MSE      | Mean Squared Error                 | \( \int t \cdot e(t)^2 \, dt \) |

4. Experimental Runs

We present two different experiments. First, open-loop control, where we only compare the control output of the two controllers. Second, closed-loop control, where we integrate the controllers into simulations with the virtual controlled system. For brevity, in the following \( PLC \) is used to indicate the hardware controller, whereas \( FMU \) signifies the virtual controller.

4.1. Open-loop control

This test is a quick and low computational-effort way to run a preliminary check on the control logic. Both controllers are fed the same artificial signals for the setpoints and fixed process variables, to observe the calculated control output. Figure 3 shows these input signals and the corresponding control outputs from the controllers. Particularly for the Chirp and Sawtooth signals, there is a significant difference between the hardware and virtual control output. Closed-loop control tests may shed light on the underlying reasons for this diverging behavior.

![Figure 3: Results of open-loop control tests. Signal left to right: Chirp, Sawtooth, and Square.](image-url)
4.2. Closed-loop control
In this test, we use all components introduced in Section 3 to observe the controllers’ feedback-powered performance. The virtual controlled system is simulated from 6:30 am till 12:30 pm, and the ambient temperature is read from a sensor attached to the PLC at our test bench. The solar radiation in addition to internal gains are taken from a validated test case in AixLib. We perform this simulation twice, once with the FMU controller and then with the PLC.

Figure 4: Closed-loop control results

Figure 4 details the behaviors of the controllers. A short period of mismatch is recorded at the beginning of the simulation ($\leq 0.3\, \text{h}$), which is expected because of how differently the FMU and PLC are initialized. This could also be affected by the instability of the PLC control output, in contrast to the smooth curve of the FMU control output. However, homogeneous behavior follows. The differences in control output from the open-loop control tests are also evident here, but with a simulation of a longer run-time and a more stable setpoint signal (i.e. with a wave period of 1 hour instead of 10 minutes), the differences do not make a considerable impact overall. Only natural ventilation is considered in the thermal zone model as a cooling factor, therefore the controllers emit 0 as control output during periods where cooling is required.
5. Discussion
As the results show, differences in control behavior surfaced after subjecting the controllers to varying setpoint signal patterns and real-life boundary conditions. Although there were variations between the control results, the KPI evaluation in Figure 5 suggests that the difference as a whole is not that significant. Therefore, it seems that virtual control testing can indeed save the effort of hardware control testing.

![Figure 5: KPI results from closed-loop control testing. The relative percent error between the hardware and virtual controller KPIs are in gray.](image-url)

6. Conclusion and outlook
We showed that model-based testing can give insight into the hardware-based testing, which can save implementation effort, if the tests consider the required boundary conditions. This may particularly benefit the pre-building commissioning phase, where many hardware functional tests are performed. Furthermore, the proposed infrastructure can also be used during commissioning.

To further investigate the differences between virtual and hardware controllers, future work could use more complex control strategies and incorporate mechanical cooling. The expectation here is that the performance of the controllers would grow even closer, allowing virtual controller tests to give a more comprehensive impression of the hardware controller behavior.

Yet another aspect to evaluate is the performance of the controllers with respect to data communications: Future tests should focus on the role of data transfer and its challenges between the hardware controller and the simulation. In many instances in practice, this is the bottleneck.

A further practice-focused aspect is the target-audience of the framework. The presented and planned technical features should be encapsulated within an intuitively usable workflow for controller testing. Therefore, future research will investigate the usability in practical applications.

Acknowledgment
We gratefully acknowledge the financial support provided by the BMWi (Federal Ministry for Economic Affairs and Energy), promotional reference 03EN1014A. A thank you as well to our business partners, eurogard GmbH.

References
[1] O’Neill Z, Li Y and Williams K 2017 Science and Technology for the Built Environment 23 619–636
[2] Müller D, Lauster M, Constantin A, Fuchs M and Remmen P 2016 BauSIM 2016 2016 3–9
[3] Blum D, Jorissen F, Huang S, Arroyo J, Benne K, Li Y, Gavan V, Rivalin L, Helsen L, Vrabie D et al. 2019
[4] Li Y, O’Neill Z D and Zhou X 2019 Science and Technology for the Built Environment 25 656–673
[5] Goldschmidt T, Murugiah M K, Sonntag C, Schlich B, Biafas S and Weber P 2015 Cloud-based control: A multi-tenant, horizontally scalable soft-plc 2015 ieee 8th international conference on cloud computing (IEEE) pp 909–916